

# **Diseño e implementación de una base de datos relacional para la gestión sanitaria**

**PFC - BASES DE DATOS RELACIONALES**

## **MEMORIA**

Estudiante: **Daniel Jesús Rönmark Cordero**

Titulación: **Ingeniería Informática**

Consultor: ***Juan Martínez Bolaños***

Semestre: **2012-13/2**

Fecha de entrega: **12/06/2013**

## Dedicatoria y Agradecimientos

Dedico este proyecto fin de carrera a mi mujer, Julia, y a mi hijo de tres meses, nacido en este semestre, Daniel, que han tenido la paciencia necesaria para verme confinado frente al ordenador y, no solo no se han quejado, sino que me han animado. A ellos doy las gracias por eso y por ser el eje principal de mis motivaciones cuando toca estudiar y trabajar duramente.

Agradecimientos a mi familia y amigos por la comprensión mostrada durante todo el proceso de la carrera así como en este último proyecto.

También doy las gracias a mi profesor del Conservatorio de Música, Aníbal Soriano, que ha entendido el esfuerzo que requería la Ingeniería Informática y me ha permitido relajar el “ritmo” en las clases.

Gracias también a todos los consultores de la UOC que a través de sus conocimientos compartidos me han ayudado a la consecución de este proyecto fin de carrera.

## Resumen

El presente Proyecto Fin de Carrera se desenvuelve en el área de Bases de Datos, concretamente de tipo relacional, y tiene como objetivo demostrar las capacidades académicas y profesionales del alumno autor del proyecto, tanto en esta área como en la de gestión y redacción de proyectos de calidad, empleando para ello los conocimientos adquiridos a través de las asignaturas de la titulación.

El problema presentado por la universidad consiste en analizar los requerimientos del nuevo sistema informático del Ministerio de Sanidad, para así poder diseñar e implementar una base de datos y un almacén de datos que dé solución al almacenamiento y explotación de información referente a médicos, pacientes, centros, medicamentos, etc.

Además, es requisito que el acceso a los datos se haga, en todo caso, mediante procedimientos almacenados, lo que pondrá de manifiesto, ampliamente, el uso de lenguajes procedimentales. Asimismo, se deben emplear mecanismos de control de la funcionalidad de la base y almacén de datos, permitiendo llegar a un nivel de desarrollo propio de cualquier organización que desee proteger y asegurar su información.

El proceso seguido en la consecución de este trabajo ha partido del análisis de requisitos y ha concluido en la fase pruebas, pasando por el diseño y la implementación, que, en definitiva, comportan el ciclo de vida clásico de desarrollo de *software*.

Para la correcta gestión del proyecto se ha seguido un plan de trabajo que se muestra más adelante y que se resume en su correspondiente diagrama de Gantt.

En síntesis, este proyecto hace uso de los conocimientos adquiridos durante el proceso estudiantil de su autor para dar solución a un problema de informatización del Ministerio de Sanidad y de todos los entes (centros, pacientes, médicos, visitas, etc.) que gestiona. En este proceso de informatización, la base de datos y el almacén de datos son los objetivos de la siguiente memoria.

## Índice de Contenido

Dedicatoria y Agradecimientos .....	2
Resumen .....	3
1. PREFACIO .....	7
1.1. Motivación .....	7
1.2. Alcance .....	8
1.3. Objetivos Generales .....	8
1.4. Evaluación Continua .....	9
1.5. Metodología .....	9
1.6. Planificación .....	10
1.6.1. Fechas Clave (Hitos).....	10
1.6.2. Tareas a Realizar .....	11
1.6.3. Cronograma .....	12
1.6.4. Diagrama de Gantt .....	13
1.7. Gestión de Riesgos .....	14
1.8. Productos Entregables.....	15
2. BASE DE DATOS .....	16
2.1. Análisis de Requisitos .....	16
2.1.1. Descripción General.....	16
2.1.2. Requisitos Funcionales .....	17
2.1.3. Diagramas de Casos de Uso .....	17
2.1.4. Especificación Textual.....	20
2.2. Diseño Técnico .....	37
2.2.1. Diseño Conceptual en UML .....	37
2.2.2. Modelo Entidad-Relación .....	39
2.2.3. Diseño Lógico.....	41
2.2.4. Diseño Físico .....	44
2.3. Implementación de la Base de Datos .....	45
2.3.1. Codificación de Scripts de Creación .....	45
2.3.2. Procedimientos de Acceso a Datos.....	46
2.4. Pruebas.....	52
3. ALMACÉN DE DATOS .....	53
3.1. Análisis de Requisitos .....	53

---

3.1.1.	Descripción General.....	53
3.1.2.	Diagramas de Casos de Uso .....	54
3.1.3.	Especificación Textual.....	54
3.1.4.	Alternativa: Soluciones OLAP.....	57
3.2.	Diseño Técnico .....	58
3.2.1.	Dimensiones y Hechos.....	58
3.2.2.	Diseño Conceptual en UML .....	59
3.2.3.	Modelo Entidad-Relación .....	59
3.2.4.	Diseño Lógico.....	60
3.2.5.	Diseño Físico.....	61
3.3.	Implementación del Almacén de Datos .....	62
3.3.1.	Codificación de Scripts de Creación .....	62
3.3.2.	Proceso de Carga de Datos .....	63
3.3.3.	Sentencias de Consulta de Estadísticas .....	65
4.	PRUEBAS Y TESTEO.....	66
4.1.	Juego de Pruebas.....	66
4.1.1.	Base de Datos .....	66
4.1.2.	Almacén de Datos.....	67
4.2.	Comprobación y Registro ( <i>log</i> ) .....	67
5.	DOCUMENTACIÓN Y MANUAL .....	69
6.	EPÍLOGO .....	72
6.1.	Valoración Económica .....	72
6.2.	Conclusiones.....	73
6.3.	Glosario de Términos y Siglas .....	73
6.4.	Bibliografía .....	74
6.5.	Anexos.....	75

## Índice de Ilustraciones

ILUSTRACIÓN 1: CICLO DE VIDA EN CASCADA .....	9
ILUSTRACIÓN 2: DIAGRAMA DE GANTT .....	13
ILUSTRACIÓN 3: CASOS DE USO GESTIÓN DE CENTROS .....	18
ILUSTRACIÓN 4: CASOS DE USO GESTIÓN DE PACIENTES .....	18
ILUSTRACIÓN 5: CASOS DE USO GESTIÓN DE MEDICAMENTOS .....	19
ILUSTRACIÓN 6: CASOS DE USO GESTIÓN DE ENFERMEDADES .....	19
ILUSTRACIÓN 7: CASOS DE USO GESTIÓN DE FARMACIAS .....	19
ILUSTRACIÓN 8: DIAGRAMA CONCEPTUAL EN UML (BD) .....	37
ILUSTRACIÓN 9: MODELO ENTIDAD-RELACIÓN A NIVEL ENTIDAD (BD).....	39
ILUSTRACIÓN 10: MODELO ENTIDAD-RELACIÓN A NIVEL DE ATRIBUTOS (BD) .....	43
ILUSTRACIÓN 11: MODELO ENTIDAD-RELACIÓN A NIVEL DE ESQUEMA FÍSICO (BD) .....	44
ILUSTRACIÓN 12: SIGNATURA DEL PROCEDIMIENTO DE AUDITORÍA (BD) .....	47
ILUSTRACIÓN 13: LISTADO DE LOS PROCEDIMIENTOS DE INSERCIÓN .....	49
ILUSTRACIÓN 14: SIGNATURA DEL PROCEDIMIENTO DE BORRADO .....	49
ILUSTRACIÓN 15: SIGNATURA DEL PROCEDIMIENTO DE MODIFICACIÓN .....	50
ILUSTRACIÓN 16: SIGNATURA DE LOS PROCEDIMIENTOS DE CONSULTA .....	51
ILUSTRACIÓN 17: SIGNATURA DE LOS PROCEDIMIENTOS DE LISTADOS .....	52
ILUSTRACIÓN 18: CASOS DE USO (DW).....	54
ILUSTRACIÓN 19: DIMENSIONES DEL ALMACÉN DE DATOS .....	58
ILUSTRACIÓN 20: HECHOS DEL ALMACÉN DE DATOS .....	58
ILUSTRACIÓN 21: DIAGRAMA CONCEPTUAL EN UML (DW) .....	59
ILUSTRACIÓN 22: MODELO ENTIDAD-RELACIÓN A NIVEL ENTIDAD (DW) .....	59
ILUSTRACIÓN 23: MODELO ENTIDAD-RELACIÓN A NIVEL DE ATRIBUTOS (DW).....	60
ILUSTRACIÓN 24: MODELO ENTIDAD-RELACIÓN A NIVEL DE ESQUEMA FÍSICO (DW) .....	61
ILUSTRACIÓN 25: SIGNATURA DEL PROCEDIMIENTO DE AUDITORÍA (DW).....	63
ILUSTRACIÓN 26: LISTADO DE LOS PROCEDIMIENTOS DE CARGA .....	64
ILUSTRACIÓN 27: SIGNATURA DE LOS PROCEDIMIENTOS ESTADÍSTICOS.....	65
ILUSTRACIÓN 28: CAPTURA DE PANTALLA - EJEMPLO LISTADO 1 .....	66
ILUSTRACIÓN 29: CAPTURA DE PANTALLA - EJEMPLO LISTADO 2 .....	66
ILUSTRACIÓN 30: CAPTURA DE PANTALLA - EJEMPLO ESTADÍSTICA .....	67
ILUSTRACIÓN 31: CAPTURA DE PANTALLA - AUDITORÍA (BD).....	68
ILUSTRACIÓN 32: CAPTURA DE PANTALLA - AUDITORÍA (DW) .....	68
ILUSTRACIÓN 33: VALORACIÓN ECONÓMICA .....	72

## 1. PREFACIO

### 1.1. Motivación

Este proyecto fin de carrera pretende hacer uso de los conocimientos adquiridos durante el recorrido académico de la Ingeniería en Informática, concretamente sobre aspectos relacionados con bases y almacenes de datos (BBDD y *data warehouse*).

El proyecto consistirá en un trabajo completo que abarque todas las fases de un proyecto de calidad profesional, pasando por cada una de las fases clásicas de cualquier proyecto serio de ingeniería informática.

En primer lugar nos centraremos en la base de datos necesaria para almacenar todos los datos del sistema propuesto, para después pasar, en segundo lugar, al estudio y análisis de explotación de la información mediante un almacén de datos, que permita extraer estadísticas y facilitar la toma de decisiones por parte de responsables y directivos.

Por último, se pretende alcanzar un grado de calidad y excelencia que queden implícitos por el uso de mecanismos de control más avanzados, como registro de acciones realizadas, tablas de auditorías, pruebas sobre el funcionamiento de la base de datos, etc.

En conclusión, el trabajo debe ser completo y preciso, de manera que cubra todas las necesidades de un sistema de información informático profesional y de calidad.

## 1.2. Alcance

El Ministerio de Sanidad va a informatizar todo su sistema de información sanitaria, tanto en hospitales como en farmacias, y ha contado con los alumnos de la UOC para realizar un proyecto completo, desde el análisis de requisitos hasta su implementación y testeo.

Esta es una gran oportunidad para demostrar que los conocimientos del escribiente se han consolidado y han sido madurados con seriedad y con la capacidad de gestión de proyectos que se espera de un ingeniero superior. Es por ello, que el sistema a desarrollar debe tener en cuenta:

- ✓ Un ciclo de vida (definido en el apartado “Metodología”) que contemple todas las etapas de construcción del software, de documentación sobre el análisis funcional, diseño técnico, resultado de las pruebas, proceso de puesta en marcha, manuales para el usuario, etc.
- ✓ La gestión de los datos se realizará mediante procedimientos de base de datos, y debe cubrirse todo tipo de gestión de centros hospitalarios, médicos, pacientes, medicamentos, urgencias, etc. así como nuevas necesidades que vayan surgiendo con el devenir del uso por parte de los usuarios de la base de datos.
- ✓ El sistema debe ser escalable, para poder ir incorporando progresivamente todas aquellas necesidades que surgen durante su vigencia.
- ✓ Se debe definir un almacén de datos para extraer estadísticas, estudiar y analizar el funcionamiento del servicio sanitario y ayudar a la toma de decisiones.
- ✓ Además, se deben proponer mecanismos para testear la funcionalidad de la BD así como la creación de registros (*logs*) sobre la actividad de la misma.
- ✓ No se contempla la creación de interfaces de usuario (GUI).

## 1.3. Objetivos Generales

Los objetivos generales de este trabajo son:

- ✓ Análisis de requisitos del sistema de información
- ✓ Diseño e implementación de la BD necesaria
- ✓ Implementación de los procedimientos de base de datos que encapsulan las funcionalidades de accesos a los datos
- ✓ Diseño y desarrollo de las características que doten a la BD de capacidad de Almacén de Datos (*data warehouse*)
- ✓ Pruebas sobre la base de datos y sobre el almacén de datos
- ✓ Implementación de mecanismos de control, registro, auditorías, testeo y otras mejoras



## 1.4. Evaluación Continua

Las entregas que satisfacen el modelo de evaluación continua de este trabajo serán:

- ✓ PEC 1: Plan de Trabajo inicial, que será usado de guía para el trabajo que se necesita realizar
- ✓ PEC 2: documentación (informes, manuales...) y producto realizado hasta el momento de la entrega
- ✓ PEC 3: documentación (informes, manuales...) y producto realizado hasta el momento de la entrega
- ✓ Entrega final que, recopilando todo lo anterior, incluye:
  - Memoria (máximo 90 páginas)
  - Presentación (máximo 20 diapositivas)
  - Trabajo práctico (producto entregable, incluyendo BD, código SQL...)

## 1.5. Metodología

En este proyecto se seguirá una metodología clásica, también conocida como ciclo de vida en cascada, que aunque no es un enfoque basado en técnicas ingenieriles actuales, como podrían ser las metodologías ágiles, sí ha demostrado que funciona muy bien en proyectos de gran envergadura con requisitos bien definidos.

Este ciclo de vida, tal y como lo hemos estudiado en asignaturas de Ingeniería del Software y de Gestión de Proyectos, es un ciclo secuencial compuesto por diferentes etapas, con carácter lineal, que serán correspondientemente reflejadas en la planificación de este proyecto, tal y como vemos a continuación:



Ilustración 1: Ciclo de Vida en Cascada

Fuente: apuntes UOC de la asignatura "Gestión de Organizaciones y Proyectos Informáticos"

1. El estudio de oportunidad podemos considerarlo realizado en el momento de la definición de este proyecto, en tanto en cuanto ya se ha tomado la decisión de promover el proyecto informático y se han analizado los requisitos generales.
2. Del análisis del sistema de información debe obtenerse un documento de análisis de requisitos bien definidos y lo más completo posible, ya que la constante modificación de los requisitos es una de las causas más comunes de fracaso de un proyecto. Este documento especifica las funciones y los objetivos del sistema informático que se quiere implementar. Además se realizan otras tareas
3. El diseño consiste en la definición de una solución técnica concreta que satisfaga las especificaciones establecidas en la fase de análisis.
4. La fase de programación, también implementación, que es como la llamaremos en este proyecto, consistirá en la creación de la base de datos y almacén de datos, con la codificación en SQL que ello implica
5. Las pruebas dotan de consistencia, fiabilidad y calidad al sistema, y son, no solo necesarias, sino que deben suponer una parte importante del desarrollo del proyecto, para poder poner en producción el sistema con las máximas garantías.
6. Aunque se sale de los objetivos del presente proyecto, no debe despreciarse en una metodología clásica la fase de mantenimiento de la aplicación durante su vigencia, máxime cuando el propio enunciado indica que la BD debe ser escalable para dar cobertura a todas las necesidades que vayan surgiendo. Además, en esta fase se corrigen errores a medida que se detectan, se mejoran funcionalidades y se desarrollan evolutivos del sistema.

## 1.6. Planificación

### 1.6.1. Fechas Clave (Hitos)

Las fechas clave para cubrir la evaluación continua son las siguientes:

<u>Inicio del semestre</u>	<u>27/02/2013</u>
<u>Enunciado del proyecto</u>	<u>27/02/2013</u>
<u>Entrega PEC1 (Plan de Trabajo)</u>	<u>17/03/2013</u>
<u>Entrega PEC2</u>	<u>21/04/2013</u>
<u>Entrega MEMORIA</u>	<u>12/06/2013</u>
<u>Presentación Memoria + Presentación + Producto</u>	<u>12/06/2013</u>
<u>Tribunal virtual</u>	<u>28/06/2013</u>

### 1.6.2. Tareas a Realizar

A continuación se indican las principales tareas en que se divide el proyecto, con una breve descripción de las mismas:

- ✓ **Inicio PFC:** consiste en la lectura del plan docente, materiales, documentación, enunciado del problema que se desea resolver, las recomendaciones de los consultores, etc. Además, en este inicio incluimos la planificación en el Plan de Trabajo, que es el entregable (PEC1) en el que culmina esta fase inicial.
- ✓ **Software de Desarrollo:** consiste en la instalación, estudio, comprensión y resolución de problemas del entorno de trabajo ORACLE que debe instalarse y configurarse.
- ✓ **Análisis de la Base de Datos:** una de las tareas críticas del proyecto, si no la que más, definir con claridad y exactitud los requisitos del sistema, así como los actores y casos de uso.
- ✓ **Diseño de la Base de Datos:** aquí se realizará, principalmente, el diseño conceptual de la base de datos en formato UML, identificando entidades, del que se obtendrá el modelo E/R, el diseño lógico y diseño físico de la BD.
- ✓ **Implementación de la Base de Datos:** mediante el SGBD de ORACLE se podrán crear la BD, sus tablas, relaciones, secuencias, disparadores... así como los procedimientos de acceso a datos, que son la única forma de acceder a los mismos.
- ✓ **Análisis del Almacén de Datos:** se definen los requisitos que debe cumplir el almacén de datos, y se estudiará las soluciones OLAP (*On-Line Analytical Processing*) para el procesamiento ágil de grandes cantidades de datos.
- ✓ **Diseño del Almacén de Datos:** se realizará el diseño conceptual del almacén de datos, en formato UML, así como su diseño lógico y diseño físico.
- ✓ **Implementación del Almacén de Datos:** en este momento se crearán los procesos automáticos de carga del almacén de datos, y todas las sentencias SQL derivadas de las necesidades detectadas.
- ✓ **Pruebas y Testeo:** las pruebas hacen referencia a juego de pruebas sobre el trabajo realizado, para corroborar que todo ha sido perfectamente definido e implementado. El testeo hace referencia a los mecanismos de testeo sobre los problemas de integración con el resto del sistema (*logs*, auditorías, etc.).
- ✓ **Fase Final:** en esta fase, en primer lugar se realizará una revisión y recapitulación del trabajo elaborado, en busca de posibles errores, y en consecuencia procediendo a su corrección. Además se deben terminar los entregables finales, que serán el producto que se facilita al cliente, en este caso el tribunal.

### 1.6.3. Cronograma

En la página siguiente se muestra una vista de la planificación del proyecto en Diagrama de Gantt, con las fechas de los hitos principales indicadas sobre el propio gráfico.

En dicho diagrama se ha establecido una jornada de trabajo de dos horas diarias, de lunes a domingo, sin indicar festivos, ya que en los días festivos se trabajará igualmente. Sin embargo, se han dejado algunas jornadas sin planificación de trabajo como contingencia ante contratiempos que puedan surgir.

Obsérvese que el proyecto comienza el 4 de marzo de 2013, en lugar del 27 de febrero. Esto es porque el 24 de febrero nació el primer hijo del que escribe, y tuvo que permanecer en el hospital acompañando a su mujer y al neonato.

### 1.6.4. Diagrama de Gantt

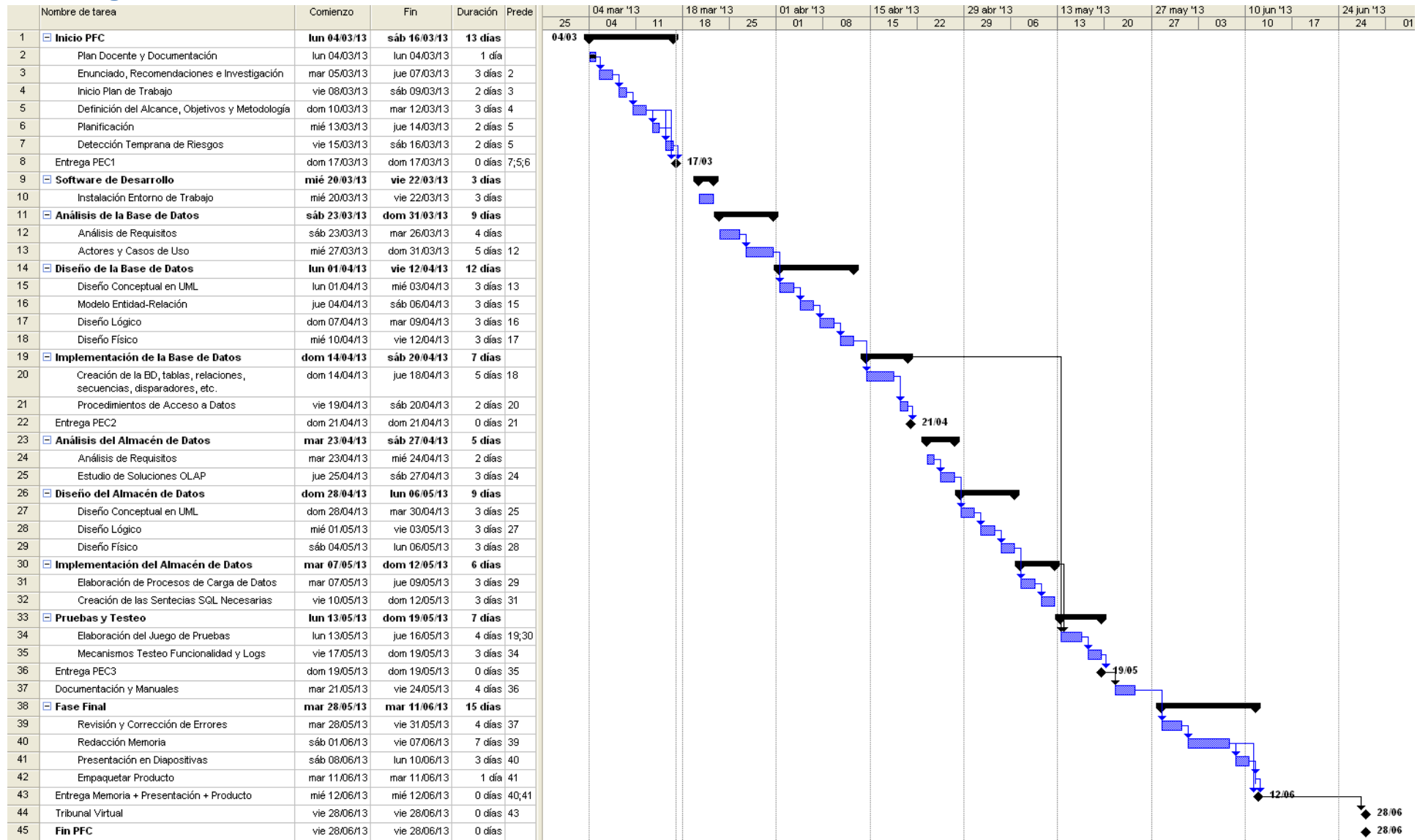


Ilustración 2: Diagrama de Gantt

## 1.7. Gestión de Riesgos

En todo proyecto pueden surgir situaciones que dificulten la correcta consecución de sus tareas, en tiempo o forma. Es por ello que se deben tener identificados cuantos riesgos sean posibles para poder anticipar un plan de contingencia que entrará en acción si procede.

En este proyecto se han tenido en cuenta los siguientes riesgos:

- ✓ Incompatibilidad con el trabajo: en los tiempos que estamos viviendo se están produciendo muchos cambios en la administración pública y sus agencias. Se da el caso de que este estudiante trabaja en una agencia pública, que constantemente debe acatar los decretos de la Junta de Andalucía, a veces con muy poco plazo de actuación. Esto podría provocar horas extra en el trabajo, como ya ha ocurrido, que imposibilitaran la jornada diaria de 3 horas que se van a dedicar al proyecto.
  - ✓ La acción mitigadora sería recuperar las horas en los fines de semana, trabajando en el proyecto las 3 horas que corresponden más las que se hayan dejado de trabajar por motivos laborales.
- ✓ Motivos personales: enfermedad/hospitalización de un familiar.
  - ✓ En el diagrama de Gantt se puede observar que tras cada entrega de PEC se ha dejado entre uno y dos días sin planificación, precisamente para mitigar este tipo de problemas, sobre todo con un hijo recién nacido como es el caso.
- ✓ Exámenes y audiciones: dado que este estudiante compagina los estudios de ingeniería con los de música, podría haber situaciones ineludibles en estos últimos que necesitaran un esfuerzo especial. Por suerte, no hay más asignaturas UOC pendientes de cursar.
  - ✓ Los profesores del conservatorio de música están avisados de que el proyecto es mi máxima prioridad, por lo que no está previsto dejar de dedicar las horas necesarias del proyecto para dedicarlas a los estudios de música. En el próximo curso ya habrá ocasión de recuperar el tiempo perdido.
- ✓ Problemas técnicos: muchas veces, algún problema técnico que no debería existir hace perder mucho tiempo. En este caso, con el software de ORACLE se podrían dar problemas de instalación, de servicios, de arranque, de espacio, etc.
  - ✓ Si esto ocurriera, habría que dedicar más horas al día de las 3 planificadas para mitigar la dificultad.
  - ✓ Si el problema viene dado por el PC de trabajo, se dispone de un portátil para continuar el proyecto hasta que el de sobremesa vuelva a estar disponible.
- ✓ Dificultades de conocimiento: es el riesgo más temido, y que puede provocar la pérdida de mucho tiempo, y es que algo se puede resistir por desconocimiento de la materia.

- ✓ En este caso se usarán horas adicionales, sobre todo en los fines de semana, para formarse. También es posible solicitar la ayuda de los consultores para que sirvan de guía y despejar una situación problemática.

## **1.8. Productos Entregables**

Una vez finalizado el proyecto fin de carrera, se tendrán los siguientes entregables, que serán correspondientemente cargados en el Repositorio Institucional de la UOC:

- ✓ Producto final o trabajo práctico que incluye la BD, el código SQL...
- ✓ Memoria, con un máximo 90 páginas, que recoge todo el trabajo hecho
- ✓ Presentación virtual, con un máximo 20 diapositivas, que resuma lo que se ha hecho

## 2. BASE DE DATOS

En este capítulo quedarán recogidas todas las fases (análisis, diseño e implementación) necesarias para la construcción de la base de datos.

### 2.1. Análisis de Requisitos

Para el análisis de requisitos será necesario basarse en el documento entregado por el cliente (enunciado), con la libertad de poder aplicar mejoras no recogidas en la especificación inicial, labor que también se considera parte de las funciones del consultor experto en el que el Ministerio de Sanidad ha confiado.

#### 2.1.1. Descripción General

Una aproximación inicial a los requisitos solicitados por el cliente, añadiendo algunas mejoras de información ofrecidas por el ente desarrollador (el estudiante escribiendo) determina los siguientes requisitos del sistema:

- ✓ Toda la gestión de los datos se realizará, únicamente, mediante procedimientos almacenados de base de datos.
- ✓ La cobertura de gestión de la base de datos se extiende desde centros hospitalarios, médicos, pacientes, enfermedades que ha tenido, visitas que ha hecho, días que ha permanecido en baja y su tipo (Incapacidad Temporal o Accidente de Trabajo, en adelante IT/AT), médicos que le atendieron, medicamentos recetados, pruebas diagnósticas que le han practicado, tipo de visita (urgencia u ordinaria, con cita previa), etc.
- ✓ El sistema debe ser escalable, para poder ir incorporando progresivamente todas aquellas necesidades que surgen durante su vigencia.
- ✓ Se debe definir un almacén de datos (*data warehouse*) para extraer estadísticas, estudiar y analizar el funcionamiento del servicio sanitario y ayudar a la toma de decisiones. Por ejemplo, en qué época hay más urgencias médicas, a qué edad se consumen más medicamentos, cuál es el tiempo medio que una persona está de baja, etc.
- ✓ Además, se deben proponer mecanismos para testear la funcionalidad de la BD así como la creación de registros (*logs*) sobre la actividad de la misma.



### 2.1.2. Requisitos Funcionales

Los requisitos funcionales de una base de datos no incluyen descripciones sobre interfaces de usuarios, sino que los actores que interactuarán con nuestro sistema serán *scripts* y otras aplicaciones que accederán a los datos mediante los procedimientos almacenados que serán definidos. Estos procedimientos almacenados deben permitir acciones sobre las tablas de manera que se pueda realizar el alta, baja, modificación y consulta de:

- Centros hospitalarios
- Farmacias
- Médicos
- Pacientes
- Catálogo de enfermedades
- Catálogo de medicamentos
- Visitas al médico (urgencias y citas previas)
- Médico que le atendió en cada visita
- Enfermedades que ha tenido el paciente
- Medicamentos recetados
- Pruebas diagnósticas practicadas
- Días de baja (IT/AT)

### 2.1.3. Diagramas de Casos de Uso

Mediante notación UML representamos, a continuación, las acciones que los actores externos pueden realizar en el sistema que se está desarrollando.

Respecto al actor de los casos de uso, suponemos que las distintas aplicaciones que acceden al sistema tienen un control de usuario, lo que les confiere de una adecuada seguridad por columnas y seguridad por filas, de manera que cuando llega la petición de acceso a datos a nuestro sistema de gestión de bases de datos, el usuario que accede a los mismos siempre es, por defecto, SYSTEM, lo que justifica el único actor que aparece en los casos de uso. Más adelante, en el proceso de implementación, definiremos el usuario administrador de nuestro sistema, como por ejemplo "SANIDAD".

Por otro lado, queda implícito, por lo que no se representará en los diagramas, que las acciones realizadas alimentarán la tabla de auditoría (*logs*), lo que permite resolver posibles problemas de integración, etc.

Separaremos en cuatro grupos temáticos de gestión, según se han detectado, de la siguiente manera:

- Gestión de Centros Hospitalarios
- Gestión de Pacientes
- Gestión de Medicamentos
- Gestión de Enfermedades
- Gestión de Farmacias

*Gestión de Centros*

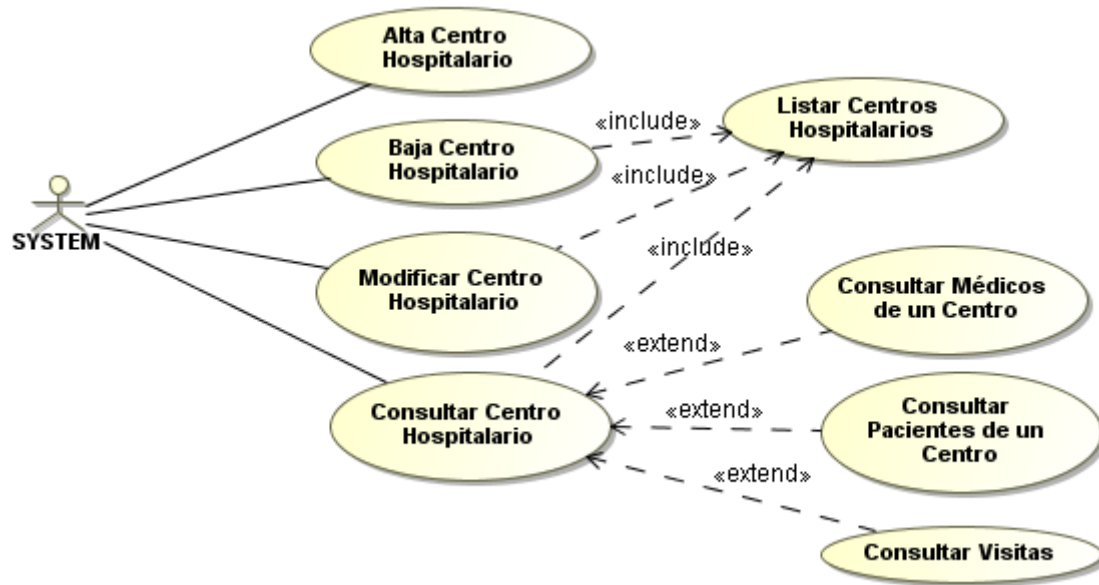


Ilustración 3: Casos de Uso Gestión de Centros

*Gestión de Pacientes*

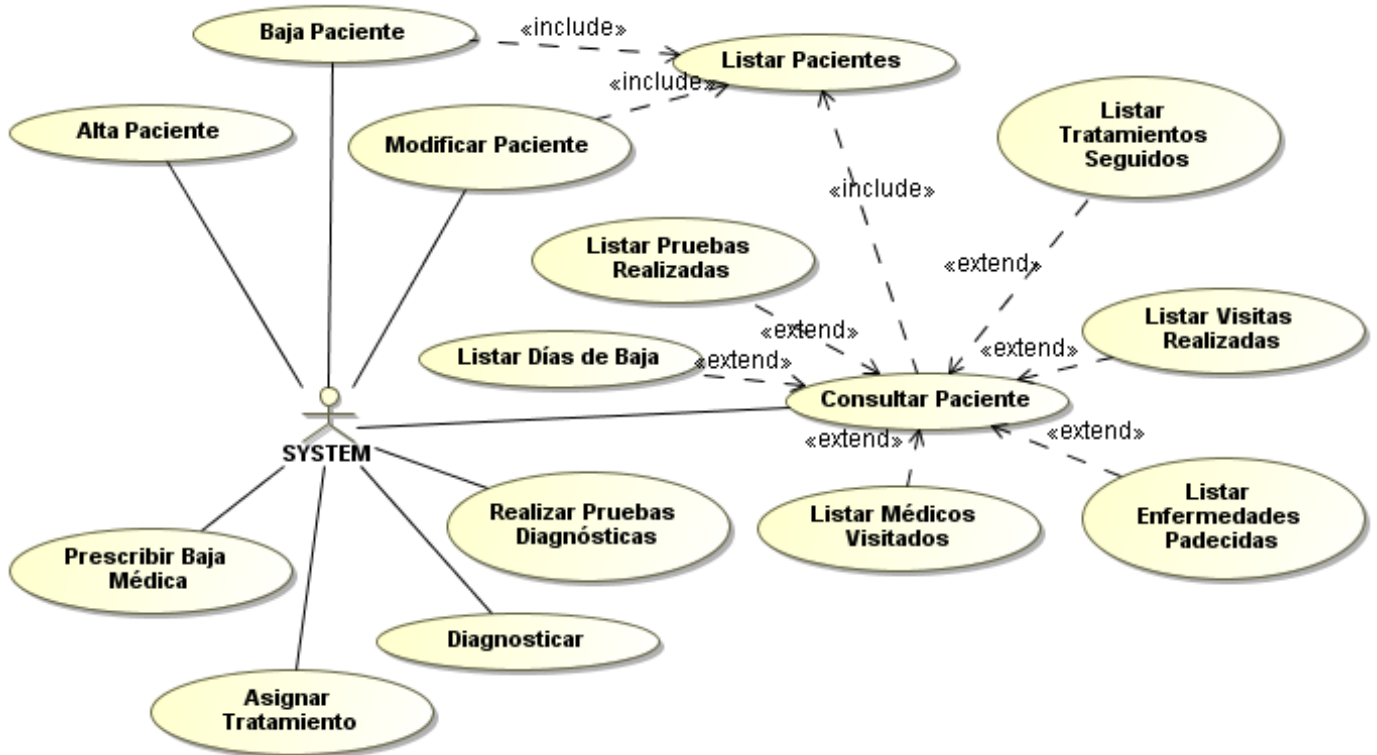


Ilustración 4: Casos de Uso Gestión de Pacientes

### Gestión de Medicamentos

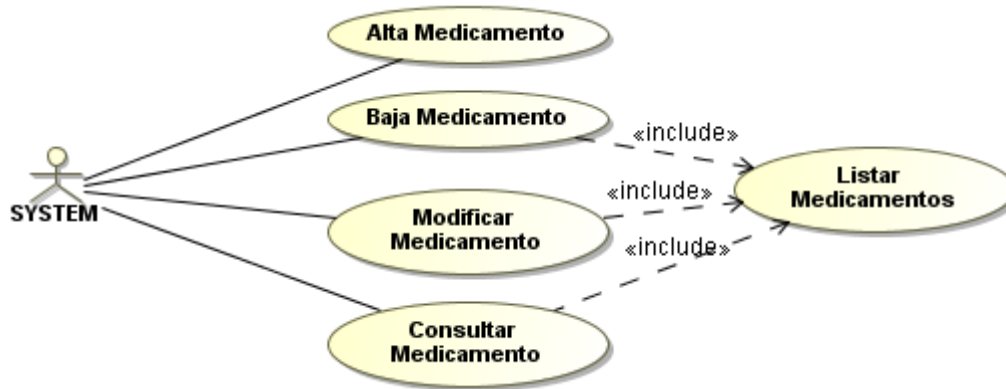


Ilustración 5: Casos de Uso Gestión de Medicamentos

### Gestión de Enfermedades

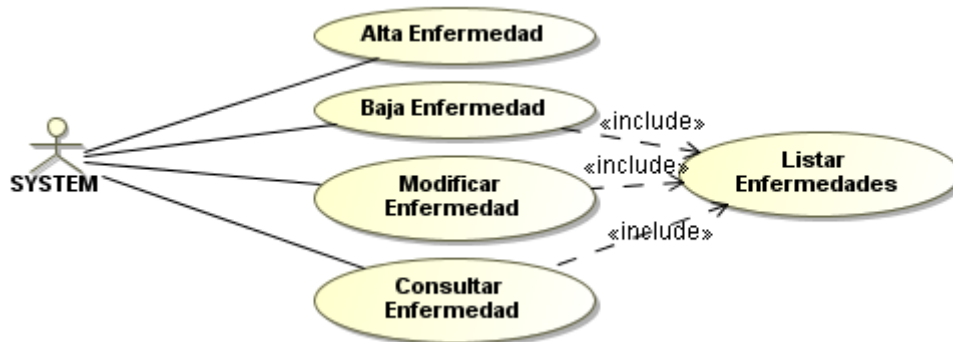


Ilustración 6: Casos de Uso Gestión de Enfermedades

### Gestión de Farmacias

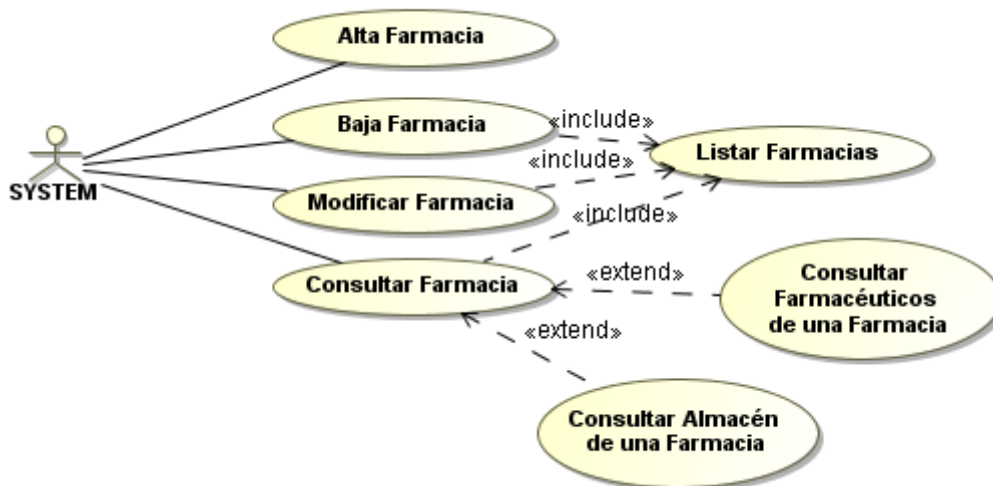


Ilustración 7: Casos de Uso Gestión de Farmacias

### 2.1.4. Especificación Textual

A continuación se describen textualmente los casos de uso presentados más arriba, con el objetivo de clarificar el alcance, entre otras características, de cada una de las funcionalidades de acceso a datos que ofrece nuestro sistema.

#### *Gestión de Centros Hospitalarios*

<b>Caso de uso</b>	<b>Alta Centro Hospitalario</b>
<b>Descripción</b>	Crea un nuevo registro en la tabla de centros hospitalarios
<b>Actor</b>	SYSTEM
<b>Precondición</b>	La tabla de centros hospitalarios existe y se encuentra en estado correcto
<b>Postcondición</b>	Se ha creado un nuevo centro hospitalario
<b>Parámetros</b>	Como mínimo, los campos obligatorios, o todos los campos de alta
<b>Salida</b>	Mensaje de operación exitosa o de error
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las altas de centros hospitalarios
<b>Flujo interacción</b>	Se comprueba la existencia del centro hospitalario en la base de datos Si existe:       se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:     se crea el registro se devuelve un mensaje de éxito en la operación se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Baja Centro Hospitalario</b>
<b>Descripción</b>	Elimina un registro de la tabla de centros hospitalarios
<b>Actor</b>	SYSTEM
<b>Precondición</b>	El registro que se desea eliminar existe en la tabla de centros hospitalarios
<b>Postcondición</b>	El registro ya no se encuentra en la tabla de centros hospitalarios
<b>Parámetros</b>	Identificador único del centro hospitalario
<b>Salida</b>	Mensaje de operación exitosa o de error
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las bajas de centros hospitalarios
<b>Flujo interacción</b>	Se comprueba la existencia del centro hospitalario en la base de datos Si existe:       se elimina el registro se devuelve un mensaje de éxito en la operación se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:     se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

Caso de uso	<b>Modificar Centro Hospitalario</b>
Descripción	Modifica determinados campos de un registro de la tabla de centros hospitalarios
Actor	SYSTEM
Precondición	El registro que se desea modificar existe en la tabla de centros hospitalarios
Postcondición	El registro ha sido actualizado
Parámetros	Identificador único del centro hospitalario y campos/valores que deben actualizarse
Salida	Mensaje de operación exitosa o de error
Disparador	Se ejecuta el procedimiento almacenado para las modificaciones de centros hospitalarios
Flujo interacción	Se comprueba la existencia del centro hospitalario en la base de datos Si existe:           se modifica el registro se devuelve un mensaje de éxito en la operación se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:       se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
Auditoría	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

Caso de uso	<b>Consultar Centro Hospitalario</b>
Descripción	Muestra un registro de la tabla de centros hospitalarios
Actor	SYSTEM
Precondición	El registro que se desea consultar existe en la tabla de centros hospitalarios
Postcondición	No
Parámetros	Identificador único del centro hospitalario y campos/valores que se desea consultar
Salida	Datos del centro hospitalario consultado o mensaje de error en caso de no existir
Disparador	Se ejecuta el procedimiento almacenado para las consultas de centros hospitalarios
Flujo interacción	Se comprueba la existencia del centro hospitalario en la base de datos Si existe:           se devuelven los datos del registro se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:       se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
Auditoría	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

Caso de uso	<b>Listar Centros Hospitalarios</b>
Descripción	Muestra un listado de centros hospitalarios según uno o varios criterios de tipo clausula WHERE
Actor	SYSTEM
Precondición	Existen registros en la tabla de centros hospitalarios que coinciden con los criterios indicados

<b>Postcondición</b>	No
<b>Parámetros</b>	Criterios identificadores de los registros que se desean obtener
<b>Salida</b>	Listado de centros hospitalarios o mensaje de error en caso de no haber encontrado coincidencias
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para listar centros hospitalarios
<b>Flujo interacción</b>	Se comprueba la existencia de los centros hospitalarios en la base de datos Si existe: se devuelven los datos del registro se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe: se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Consultar Médicos de un Centro</b>
<b>Descripción</b>	Devuelve una lista de registros de la tabla de médicos que están adscritos a un centro hospitalario concreto
<b>Actor</b>	SYSTEM
<b>Precondición</b>	La tabla de médicos existe y los registros que cumplen el criterio indicado existen
<b>Postcondición</b>	No
<b>Parámetros</b>	Identificador único del centro hospitalario cuyos médicos adscritos se desean consultar
<b>Salida</b>	Datos básicos de los médicos del centro hospitalario consultado o mensaje de error en caso de no existir
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las consultas de médicos de un centro hospitalario
<b>Flujo interacción</b>	Se comprueba la existencia del centro hospitalario en la base de datos Se comprueba la existencia de médicos adscritos a ese centro hospitalario Si existen: se devuelven los datos solicitados se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existen: se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Consultar Pacientes de un Centro</b>
<b>Descripción</b>	Devuelve una lista de registros de la tabla de pacientes que están adscritos a un centro hospitalario concreto
<b>Actor</b>	SYSTEM
<b>Precondición</b>	La tabla de pacientes existe y los registros que cumplen el criterio indicado existen
<b>Postcondición</b>	No
<b>Parámetros</b>	Identificador único del centro hospitalario cuyos pacientes adscritos se desean consultar
<b>Salida</b>	Datos básicos de los pacientes del centro hospitalario consultado o mensaje

	de error en caso de no existir
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las consultas de pacientes de un centro hospitalario
<b>Flujo interacción</b>	Se comprueba la existencia del centro hospitalario en la base de datos Se comprueba la existencia de pacientes adscritos a ese centro hospitalario Si existen:       se devuelven los datos solicitados se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existen:    se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Consultar Visitas</b>
<b>Descripción</b>	Devuelve una lista de registros de la tabla de visitas médicas que ha recibido cualquier médico de un centro hospitalario concreto
<b>Actor</b>	SYSTEM
<b>Precondición</b>	La tabla de visitas médicas existe y los registros que cumplen el criterio indicado existen
<b>Postcondición</b>	No
<b>Parámetros</b>	Identificador único del centro hospitalario cuyas visitas médicas recibidas se desean consultar
<b>Salida</b>	Datos básicos de las visitas médicas del centro hospitalario consultado o mensaje de error en caso de no existir
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las consultas de visitas médicas de un centro hospitalario
<b>Flujo interacción</b>	Se comprueba la existencia del centro hospitalario en la base de datos Se comprueba la existencia de visitas médicas recibidas en ese centro hospitalario Si existen:       se devuelven los datos solicitados se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existen:    se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

### *Gestión de Pacientes*

<b>Caso de uso</b>	<b>Alta Paciente</b>
<b>Descripción</b>	Crea un nuevo registro en la tabla de pacientes
<b>Actor</b>	SYSTEM
<b>Precondición</b>	La tabla de pacientes existe y se encuentra en estado correcto
<b>Postcondición</b>	Se ha creado un nuevo paciente
<b>Parámetros</b>	Como mínimo, los campos obligatorios, o todos los campos de alta

<b>Salida</b>	Mensaje de operación exitosa o de error
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las altas de pacientes
<b>Flujo interacción</b>	Se comprueba la existencia del paciente en la base de datos Si existe: se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe: se crea el registro se devuelve un mensaje de éxito en la operación se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Baja Paciente</b>
<b>Descripción</b>	Elimina un registro de la tabla de pacientes
<b>Actor</b>	SYSTEM
<b>Precondición</b>	El registro que se desea eliminar existe en la tabla de pacientes
<b>Postcondición</b>	El registro ya no se encuentra en la tabla de pacientes
<b>Parámetros</b>	Identificador único del paciente
<b>Salida</b>	Mensaje de operación exitosa o de error
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las bajas de pacientes
<b>Flujo interacción</b>	Se comprueba la existencia del paciente en la base de datos Si existe: se elimina el registro se devuelve un mensaje de éxito en la operación se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe: se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Modificar Paciente</b>
<b>Descripción</b>	Modifica determinados campos de un registro de la tabla de pacientes
<b>Actor</b>	SYSTEM
<b>Precondición</b>	El registro que se desea modificar existe en la tabla de pacientes
<b>Postcondición</b>	El registro ha sido actualizado
<b>Parámetros</b>	Identificador único del paciente y campos/valores que deben actualizarse
<b>Salida</b>	Mensaje de operación exitosa o de error
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las modificaciones de pacientes
<b>Flujo interacción</b>	Se comprueba la existencia del paciente en la base de datos Si existe: se modifica el registro se devuelve un mensaje de éxito en la operación se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe: se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido,



	tanto si se tiene éxito como si no
--	------------------------------------

Caso de uso	Consultar Paciente
Descripción	Muestra un registro de la tabla de pacientes
Actor	SYSTEM
Precondición	El registro que se desea consultar existe en la tabla de pacientes
Postcondición	No
Parámetros	Identificador único del paciente y campos/valores que se desea consultar
Salida	Datos del paciente consultado o mensaje de error en caso de no existir
Disparador	Se ejecuta el procedimiento almacenado para las consultas de pacientes
Flujo interacción	Se comprueba la existencia del paciente en la base de datos Si existe:           se devuelven los datos del registro se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:       se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
Auditoría	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

Caso de uso	Realizar Pruebas Diagnósticas
Descripción	Crea las entradas en las tablas necesarias para que una prueba diagnóstica quede solicitada y posteriormente, tras su realización, registrados sus resultados
Actor	SYSTEM
Precondición	El paciente tiene registrada una visita médica
Postcondición	La prueba ha quedado registrada
Parámetros	Identificador único del paciente y campos/valores y tablas que deben actualizarse para solicitar/realizar la prueba
Salida	Mensaje de operación exitosa o de error
Disparador	Se ejecuta el procedimiento almacenado para la realización de pruebas
Flujo interacción	Se comprueba la existencia del paciente en la base de datos Si existe:           se añade el registro en la tabla de prueba diagnóstica se devuelve un mensaje de éxito en la operación se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:       se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
Auditoría	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

Caso de uso	Diagnosticar
Descripción	Realiza el diagnóstico a un paciente tras realizar la visita al médico
Actor	SYSTEM
Precondición	El paciente tiene registrada una visita médica
Postcondición	El diagnóstico ha quedado registrada
Parámetros	Identificador único del paciente y campos/valores que deben actualizarse para el diagnóstico
Salida	Mensaje de operación exitosa o de error

<b>Disparador</b>	Se ejecuta el procedimiento almacenado para la realización de diagnósticos
<b>Flujo interacción</b>	Se comprueba la existencia del paciente en la base de datos Si existe:        se añade el registro en la tabla de diagnósticos se devuelve un mensaje de éxito en la operación se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:    se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Asignar Tratamiento</b>
<b>Descripción</b>	Asigna un tratamiento a un paciente tras realizar la visita al médico y ser diagnosticado. Es posible que conlleve pruebas diagnósticas
<b>Actor</b>	SYSTEM
<b>Precondición</b>	El paciente tiene registrada una visita médica
<b>Postcondición</b>	El paciente tiene un tratamiento asociado
<b>Parámetros</b>	Identificador único del paciente y campos/valores que deben actualizarse para el tratamiento
<b>Salida</b>	Mensaje de operación exitosa o de error
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para la realización de tratamientos
<b>Flujo interacción</b>	Se comprueba la existencia del paciente en la base de datos Si existe:        se añade el registro en las tabla de tratamientos, que tiene líneas para varios medicamentos se devuelve un mensaje de éxito en la operación se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:    se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Prescribir Baja Médica</b>
<b>Descripción</b>	Prescribe una baja médica a un paciente tras realizar la visita al médico y ser diagnosticado. Es posible que conlleve pruebas diagnósticas
<b>Actor</b>	SYSTEM
<b>Precondición</b>	El paciente tiene registrada una visita médica
<b>Postcondición</b>	El paciente se encuentra en situación de baja médica
<b>Parámetros</b>	Identificador único del paciente y campos/valores que deben actualizarse para la baja médica
<b>Salida</b>	Mensaje de operación exitosa o de error
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para la realización de bajas médicas
<b>Flujo interacción</b>	Se comprueba la existencia del paciente en la base de datos Si existe:        se añade el registro en la tabla de baja médica se devuelve un mensaje de éxito en la operación se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:    se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )

<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no
------------------	--

<b>Caso de uso</b>	<b>Listar Pacientes</b>
<b>Descripción</b>	Muestra un listado de pacientes según uno o varios criterios de tipo cláusula WHERE
<b>Actor</b>	SYSTEM
<b>Precondición</b>	Existen registros en la tabla de pacientes que coinciden con los criterios indicados
<b>Postcondición</b>	No
<b>Parámetros</b>	Criterios identificadores de los registros que se desean obtener
<b>Salida</b>	Listado de pacientes o mensaje de error en caso de no haber encontrado coincidencias
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para listar pacientes
<b>Flujo interacción</b>	Se comprueba la existencia del paciente en la base de datos Si existe: se devuelven los datos del registro se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe: se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Listar Tratamientos Seguidos</b>
<b>Descripción</b>	Devuelve una lista de registros de la tabla de tratamientos que ha seguido un paciente concreto. También incluye los tratamientos actuales
<b>Actor</b>	SYSTEM
<b>Precondición</b>	Las tablas de pacientes y tratamientos existen y el paciente y sus tratamientos consultados existen
<b>Postcondición</b>	No
<b>Parámetros</b>	Identificador único del paciente cuyos tratamientos se desean consultar
<b>Salida</b>	Datos básicos de los tratamientos del paciente consultado o mensaje de error en caso de no existir
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las consultas de tratamientos seguidos por un paciente
<b>Flujo interacción</b>	Se comprueba la existencia del paciente en la base de datos Se comprueba la existencia de tratamientos seguidos por ese paciente Si existen: se devuelven los datos solicitados se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existen: se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Listar Visitas Realizadas</b>
--------------------	----------------------------------

<b>Descripción</b>	Devuelve una lista de registros de la tabla de visitas médicas que ha realizado un paciente concreto
<b>Actor</b>	SYSTEM
<b>Precondición</b>	La tabla de visitas médicas existe y los registros que cumplen el criterio indicado existen
<b>Postcondición</b>	No
<b>Parámetros</b>	Identificador único del paciente cuyas visitas médicas realizadas se desean consultar
<b>Salida</b>	Datos básicos de las visitas médicas del paciente consultado o mensaje de error en caso de no existir
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las consultas de visitas médicas de un paciente
<b>Flujo interacción</b>	Se comprueba la existencia del paciente en la base de datos Se comprueba la existencia de visitas médicas realizadas por ese paciente Si existen:       se devuelven los datos solicitados se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existen:    se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Listar Enfermedades Padecidas</b>
<b>Descripción</b>	Devuelve una lista de registros de la tabla de enfermedades del paciente que ha padecido un paciente concreto. También incluye las enfermedades actuales
<b>Actor</b>	SYSTEM
<b>Precondición</b>	Las tablas de pacientes y enfermedades del paciente existen y el paciente y sus enfermedades existen
<b>Postcondición</b>	No
<b>Parámetros</b>	Identificador único del paciente cuyas enfermedades se desean consultar
<b>Salida</b>	Datos básicos de las enfermedades del paciente consultado o mensaje de error en caso de no existir
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las consultas de enfermedades padecidas por un paciente
<b>Flujo interacción</b>	Se comprueba la existencia del paciente en la base de datos Se comprueba la existencia de enfermedades padecidas por ese paciente Si existen:       se devuelven los datos solicitados se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existen:    se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Listar Médicos Visitados</b>
<b>Descripción</b>	Devuelve una lista de registros de la tabla de médicos que ha visitado un paciente concreto

<b>Actor</b>	SYSTEM
<b>Precondición</b>	La tabla de visitas médicas existe, así como el paciente consultado
<b>Postcondición</b>	No
<b>Parámetros</b>	Identificador único del paciente cuyas visitas médicas se desean consultar
<b>Salida</b>	Datos básicos de las visitas médicas del paciente consultado o mensaje de error en caso de no existir
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las consultas de visitas médicas realizadas por un paciente
<b>Flujo interacción</b>	Se comprueba la existencia del paciente en la base de datos Se comprueba la existencia de visitas médicas realizadas por ese paciente Si existen:       se devuelven los datos solicitados se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existen:    se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Listar Días de Baja</b>
<b>Descripción</b>	Devuelve una lista de registros de la tabla de procesos IT/AT que ha sufrido un paciente concreto
<b>Actor</b>	SYSTEM
<b>Precondición</b>	La tabla de procesos IT/AT existe, así como el paciente consultado
<b>Postcondición</b>	No
<b>Parámetros</b>	Identificador único del paciente cuyos procesos IT/AT se desean consultar
<b>Salida</b>	Datos básicos de los procesos IT/AT del paciente consultado o mensaje de error en caso de no existir
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las consultas de procesos IT/AT sufridos por un paciente
<b>Flujo interacción</b>	Se comprueba la existencia del paciente en la base de datos Se comprueba la existencia de procesos IT/AT sufridos por ese paciente Si existen:       se devuelven los datos solicitados se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existen:    se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Listar Pruebas Realizadas</b>
<b>Descripción</b>	Devuelve una lista de registros de la tabla de pruebas diagnósticas a las que se ha sometido un paciente concreto
<b>Actor</b>	SYSTEM
<b>Precondición</b>	La tabla de pruebas diagnósticas del paciente existe, así como el paciente consultado
<b>Postcondición</b>	No
<b>Parámetros</b>	Identificador único del paciente cuyas pruebas diagnósticas se desean

	consultar
<b>Salida</b>	Datos básicos de las pruebas diagnósticas del paciente consultado o mensaje de error en caso de no existir
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las consultas de pruebas diagnósticas a las que se ha sometido un paciente
<b>Flujo interacción</b>	Se comprueba la existencia del paciente en la base de datos Se comprueba la existencia de pruebas diagnósticas a las que se ha sometido ese paciente Si existen:        se devuelven los datos solicitados se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existen:    se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

### *Gestión de Medicamentos*

<b>Caso de uso</b>	<b>Alta Medicamento</b>
<b>Descripción</b>	Crea un nuevo registro en la tabla de medicamentos
<b>Actor</b>	SYSTEM
<b>Precondición</b>	La tabla de medicamentos existe y se encuentra en estado correcto
<b>Postcondición</b>	Se ha creado un nuevo medicamento
<b>Parámetros</b>	Como mínimo, los campos obligatorios, o todos los campos de alta
<b>Salida</b>	Mensaje de operación exitosa o de error
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las altas de medicamentos
<b>Flujo interacción</b>	Se comprueba la existencia del medicamento en la base de datos Si existe:        se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:    se crea el registro se devuelve un mensaje de éxito en la operación se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Baja Medicamento</b>
<b>Descripción</b>	Elimina un registro de la tabla de medicamentos
<b>Actor</b>	SYSTEM
<b>Precondición</b>	El registro que se desea eliminar existe en la tabla de medicamentos
<b>Postcondición</b>	El registro ya no se encuentra en la tabla de medicamentos
<b>Parámetros</b>	Identificador único del medicamento
<b>Salida</b>	Mensaje de operación exitosa o de error
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las bajas de medicamentos
<b>Flujo interacción</b>	Se comprueba la existencia del medicamento en la base de datos

	<p>Si existe:        se elimina el registro                       se devuelve un mensaje de éxito en la operación                       se registra la acción en la tabla de auditoría (<i>logs</i>)</p> <p>Si no existe:    se devuelve un mensaje de error SQL explicando lo ocurrido                       se registra la acción en la tabla de auditoría (<i>logs</i>)</p>
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Modificar Medicamento</b>
<b>Descripción</b>	Modifica determinados campos de un registro de la tabla de medicamentos
<b>Actor</b>	SYSTEM
<b>Precondición</b>	El registro que se desea modificar existe en la tabla de medicamentos
<b>Postcondición</b>	El registro ha sido actualizado
<b>Parámetros</b>	Identificador único del medicamento y campos/valores que deben actualizarse
<b>Salida</b>	Mensaje de operación exitosa o de error
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las modificaciones de medicamentos
<b>Flujo interacción</b>	<p>Se comprueba la existencia del medicamento en la base de datos</p> <p>Si existe:        se modifica el registro                       se devuelve un mensaje de éxito en la operación                       se registra la acción en la tabla de auditoría (<i>logs</i>)</p> <p>Si no existe:    se devuelve un mensaje de error SQL explicando lo ocurrido                       se registra la acción en la tabla de auditoría (<i>logs</i>)</p>
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Consultar Medicamento</b>
<b>Descripción</b>	Muestra un registro de la tabla de medicamentos
<b>Actor</b>	SYSTEM
<b>Precondición</b>	El registro que se desea consultar existe en la tabla de medicamentos
<b>Postcondición</b>	No
<b>Parámetros</b>	Identificador único del medicamento y campos/valores que se desea consultar
<b>Salida</b>	Datos del medicamento consultado o mensaje de error en caso de no existir
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las consultas de medicamentos
<b>Flujo interacción</b>	<p>Se comprueba la existencia del medicamento en la base de datos</p> <p>Si existe:        se devuelven los datos del registro                       se registra la acción en la tabla de auditoría (<i>logs</i>)</p> <p>Si no existe:    se devuelve un mensaje de error SQL explicando lo ocurrido                       se registra la acción en la tabla de auditoría (<i>logs</i>)</p>
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Listar Medicamentos</b>
<b>Descripción</b>	Muestra un listado de medicamentos según uno o varios criterios de tipo clausula WHERE
<b>Actor</b>	SYSTEM
<b>Precondición</b>	Existen registros en la tabla de medicamentos que coinciden con los criterios indicados
<b>Postcondición</b>	No
<b>Parámetros</b>	Criterios identificadores de los registros que se desean obtener
<b>Salida</b>	Listado de medicamentos o mensaje de error en caso de no haber encontrado coincidencias
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para listar medicamentos
<b>Flujo interacción</b>	Se comprueba la existencia del medicamento en la base de datos Si existe:        se devuelven los datos del registro se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:    se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

### *Gestión de Enfermedades*

<b>Caso de uso</b>	<b>Alta Enfermedad</b>
<b>Descripción</b>	Crea un nuevo registro en la tabla de enfermedades
<b>Actor</b>	SYSTEM
<b>Precondición</b>	La tabla de enfermedades existe y se encuentra en estado correcto
<b>Postcondición</b>	Se ha creado una nueva enfermedad
<b>Parámetros</b>	Como mínimo, los campos obligatorios, o todos los campos de alta
<b>Salida</b>	Mensaje de operación exitosa o de error
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las altas de enfermedades
<b>Flujo interacción</b>	Se comprueba la existencia de la enfermedad en la base de datos Si existe:        se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:    se crea el registro se devuelve un mensaje de éxito en la operación se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Baja Enfermedad</b>
<b>Descripción</b>	Elimina un registro de la tabla de enfermedades
<b>Actor</b>	SYSTEM
<b>Precondición</b>	El registro que se desea eliminar existe en la tabla de enfermedades



<b>Postcondición</b>	El registro ya no se encuentra en la tabla de enfermedades
<b>Parámetros</b>	Identificador único de la enfermedad
<b>Salida</b>	Mensaje de operación exitosa o de error
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las bajas de enfermedades
<b>Flujo interacción</b>	Se comprueba la existencia de la enfermedad en la base de datos Si existe:       se elimina el registro se devuelve un mensaje de éxito en la operación se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:     se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Modificar Enfermedad</b>
<b>Descripción</b>	Modifica determinados campos de un registro de la tabla de enfermedades
<b>Actor</b>	SYSTEM
<b>Precondición</b>	El registro que se desea modificar existe en la tabla de enfermedades
<b>Postcondición</b>	El registro ha sido actualizado
<b>Parámetros</b>	Identificador único de la enfermedad y campos/valores que deben actualizarse
<b>Salida</b>	Mensaje de operación exitosa o de error
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las modificaciones de enfermedades
<b>Flujo interacción</b>	Se comprueba la existencia de la enfermedad en la base de datos Si existe:       se modifica el registro se devuelve un mensaje de éxito en la operación se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:     se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Consultar Enfermedad</b>
<b>Descripción</b>	Muestra un registro de la tabla de enfermedades
<b>Actor</b>	SYSTEM
<b>Precondición</b>	El registro que se desea consultar existe en la tabla de enfermedades
<b>Postcondición</b>	No
<b>Parámetros</b>	Identificador único de la enfermedad y campos/valores que se desea consultar
<b>Salida</b>	Datos de la enfermedad consultada o mensaje de error en caso de no existir
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las consultas de enfermedades
<b>Flujo interacción</b>	Se comprueba la existencia de la enfermedad en la base de datos Si existe:       se devuelven los datos del registro se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:     se devuelve un mensaje de error SQL explicando lo ocurrido

	se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Listar Enfermedades</b>
<b>Descripción</b>	Muestra un listado de enfermedades según uno o varios criterios de tipo clausula WHERE
<b>Actor</b>	SYSTEM
<b>Precondición</b>	Existen registros en la tabla de enfermedades que coinciden con los criterios indicados
<b>Postcondición</b>	No
<b>Parámetros</b>	Criterios identificadores de los registros que se desean obtener
<b>Salida</b>	Listado de enfermedades o mensaje de error en caso de no haber encontrado coincidencias
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para listar enfermedades
<b>Flujo interacción</b>	Se comprueba la existencia de la enfermedad en la base de datos Si existe:           se devuelven los datos del registro se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:       se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

### *Gestión de Farmacias*

<b>Caso de uso</b>	<b>Alta Farmacia</b>
<b>Descripción</b>	Crea un nuevo registro en la tabla de farmacias
<b>Actor</b>	SYSTEM
<b>Precondición</b>	La tabla de farmacias existe y se encuentra en estado correcto
<b>Postcondición</b>	Se ha creado una nueva farmacia
<b>Parámetros</b>	Como mínimo, los campos obligatorios, o todos los campos de alta
<b>Salida</b>	Mensaje de operación exitosa o de error
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las altas de farmacias
<b>Flujo interacción</b>	Se comprueba la existencia de la farmacia en la base de datos Si existe:           se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:       se crea el registro se devuelve un mensaje de éxito en la operación se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Baja Farmacia</b>
<b>Descripción</b>	Elimina un registro de la tabla de farmacias
<b>Actor</b>	SYSTEM

<b>Precondición</b>	El registro que se desea eliminar existe en la tabla de farmacias
<b>Postcondición</b>	El registro ya no se encuentra en la tabla de farmacias
<b>Parámetros</b>	Identificador único de la farmacia
<b>Salida</b>	Mensaje de operación exitosa o de error
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las bajas de farmacias
<b>Flujo interacción</b>	Se comprueba la existencia de la farmacia en la base de datos Si existe:        se elimina el registro se devuelve un mensaje de éxito en la operación se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:    se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Modificar Farmacia</b>
<b>Descripción</b>	Modifica determinados campos de un registro de la tabla de farmacias
<b>Actor</b>	SYSTEM
<b>Precondición</b>	El registro que se desea modificar existe en la tabla de farmacias
<b>Postcondición</b>	El registro ha sido actualizado
<b>Parámetros</b>	Identificador único de la farmacia y campos/valores que deben actualizarse
<b>Salida</b>	Mensaje de operación exitosa o de error
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las modificaciones de farmacias
<b>Flujo interacción</b>	Se comprueba la existencia de la farmacia en la base de datos Si existe:        se modifica el registro se devuelve un mensaje de éxito en la operación se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:    se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Consultar Farmacia</b>
<b>Descripción</b>	Muestra un registro de la tabla de farmacias
<b>Actor</b>	SYSTEM
<b>Precondición</b>	El registro que se desea consultar existe en la tabla de farmacias
<b>Postcondición</b>	No
<b>Parámetros</b>	Identificador único de la farmacia y campos/valores que se desea consultar
<b>Salida</b>	Datos de la farmacia consultado o mensaje de error en caso de no existir
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las consultas de farmacias
<b>Flujo interacción</b>	Se comprueba la existencia de la farmacia en la base de datos Si existe:        se devuelven los datos del registro se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:    se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido,

	tanto si se tiene éxito como si no
--	------------------------------------

Caso de uso	Listar Farmacias
<b>Descripción</b>	Muestra un listado de farmacias según uno o varios criterios de tipo clausula WHERE
<b>Actor</b>	SYSTEM
<b>Precondición</b>	Existen registros en la tabla de farmacias que coinciden con los criterios indicados
<b>Postcondición</b>	No
<b>Parámetros</b>	Criterios identificadores de los registros que se desean obtener
<b>Salida</b>	Listado de farmacias o mensaje de error en caso de no haber encontrado coincidencias
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para listar farmacias
<b>Flujo interacción</b>	Se comprueba la existencia de las farmacias en la base de datos Si existe:           se devuelven los datos del registro se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existe:       se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

Caso de uso	Consultar Almacén de una Farmacia
<b>Descripción</b>	Devuelve una lista de registros de la tabla de almacén que tiene una farmacia concreto, de manera que se pueden ver todos los medicamentos disponibles y sus cantidades
<b>Actor</b>	SYSTEM
<b>Precondición</b>	La tabla de almacén de una farmacia existe y los registros que cumplen el criterio indicado existen
<b>Postcondición</b>	No
<b>Parámetros</b>	Identificador único de la farmacia cuyo almacén se desea consultar
<b>Salida</b>	Datos básicos de la composición del almacén de la farmacia consultada o mensaje de error en caso de no existir
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para las consultas de almacén de una farmacia
<b>Flujo interacción</b>	Se comprueba la existencia de la farmacia en la base de datos Se comprueba la existencia de almacén de esa farmacia Si existen:           se devuelven los datos solicitados se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existen:       se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

## 2.2. Diseño Técnico

Tras el análisis de requisitos, cabe diseñar técnicamente la solución propuesta. Esto se llevará a cabo mediante un diseño conceptual que se presentará en un diagrama UML, lo que permitirá obtener el modelo entidad-relación (E/R). Una vez encontrado este modelo, solo queda traducirlo a la tecnología Oracle mediante el diseño lógico y el diseño físico de la BD.

### 2.2.1. Diseño Conceptual en UML

El diseño conceptual presenta las entidades que se desprenden del análisis de requisitos, independientemente de la tecnología del sistema de gestión de bases de datos. Aquí veremos su representación en UML.

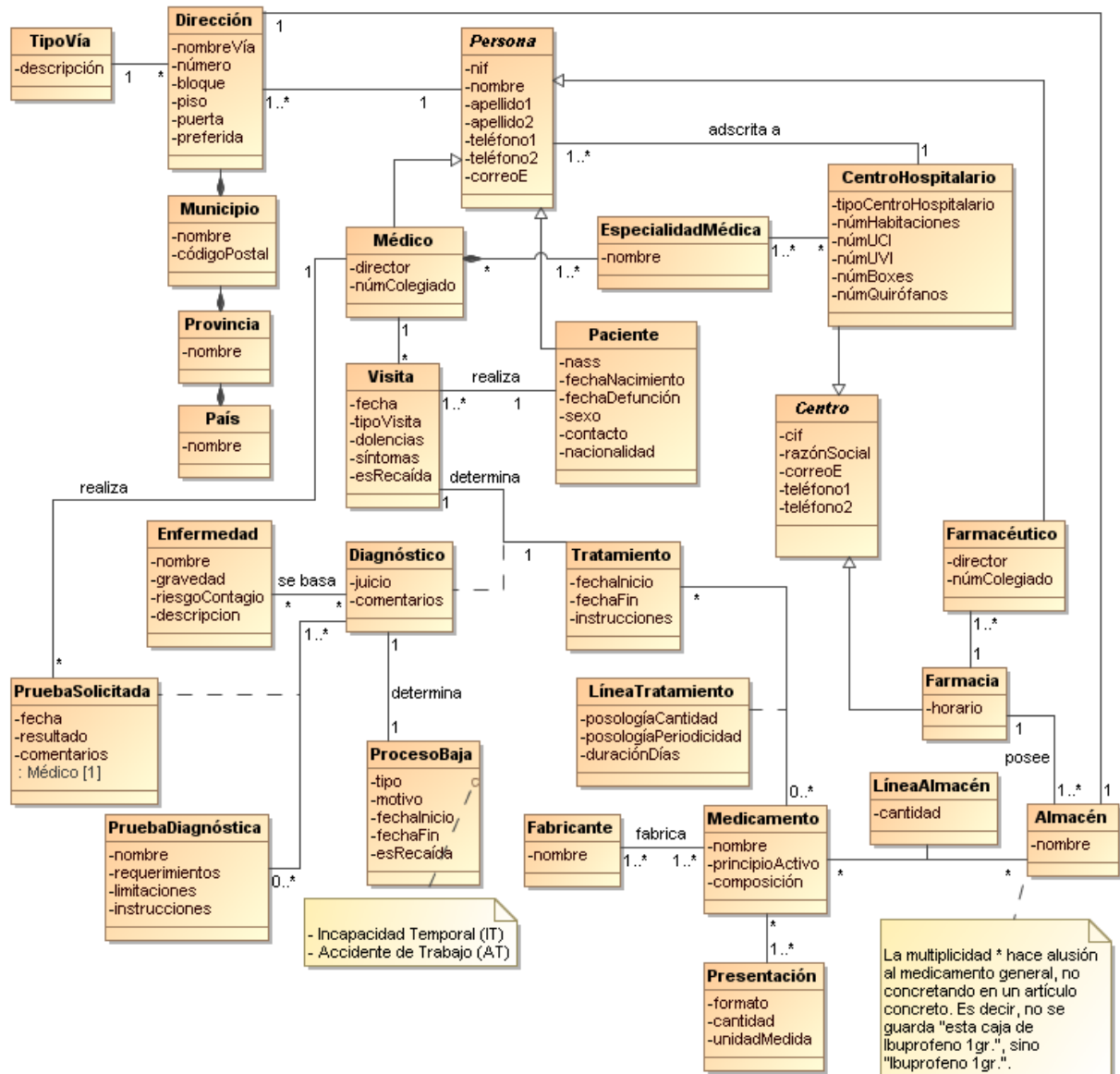


Ilustración 8: Diagrama Conceptual en UML (BD)

### *Aclaraciones*

- El diagnóstico, a pesar de haber sido representado como una entidad asociativa entre “Visita” y “Tratamiento” para que su comprensión sea más inmediata (“lo que hay entre una visita al médico y un tratamiento, es un diagnóstico”), es el centro sobre el que giran las enfermedades, los tratamientos, los procesos de baja y las pruebas diagnósticas. Esto quiere decir que siempre es necesario que un médico dictamine un diagnóstico para:
  - ✓ Indicar un tratamiento a un paciente.
  - ✓ Prescribir una baja médica
  - ✓ Solicitar la realización de una prueba diagnóstica.
  - ✓ Diagnosticar una enfermedad.
- ✓ Así, para llegar a estas desde el paciente, debemos acceder a la visita en la que se ha procedido a dicho dictamen.
- No se contempla la posibilidad de que un farmacéutico trabaje en varias farmacias, no es habitual, pero sí que en una farmacia trabajen varios farmacéuticos.
- El atributo “director” de la entidad “Farmacéutico” y “Médico” es booleano e indica si es o no el director del centro donde trabaja.
- El atributo “tipo” de “ProcesoBaja” indica si es IT o AT, por el tratamiento diferenciado que esta tiene en la Seguridad Social, mientras que “motivo” es más específico, y da información sobre lo que causa la baja, como “riesgo por embarazo”, “riesgo por lactancia”, “psicológico”, “inmovilidad”, etc.
- El atributo “esRecaída” de las entidades “Visita” y “ProcesoBaja” son booleanos que indican si la dolencia es la recaída parte de un proceso anterior.
- El atributo “tipoVisita” de “Visita” ofrece una visión general del tipo de visita, “urgente”, “ordinaria”, “recetas”, “revisión análisis”, “extracción sangre”, etc.
- El “tipoCentroHospitalario” de “CentroHospitalario” puede presentar valores como “Ambulatorio”, “Hospital de la Mujer”, “Infantil”, “Salud Mental”, “Urgencias”, etc.
- La multiplicidad \* desde “Medicamento” hacia “Almacén” hace alusión al medicamento general, no concretando en un artículo concreto. Es decir, no se guarda "esta caja de Ibuprofeno 1gr.", sino "Ibuprofeno 1gr."
- El atributo “contacto” de “Paciente” es de texto libre y permite cualquier tipo de información sobre la persona a la que debe contactarse en caso de que sea necesario.
- “PruebaSolicitada” es una clase asociativa entre “PruebaDiagnóstica” y “Diagnóstico”, que tendrán, necesariamente, en el modelo relacional una asociación “de muchos a muchos”. Esta “PruebaSolicitada” la realiza siempre un único médico, que es el responsable de la

misma, aunque a nivel práctico la delegue en otro personal sanitario, como puede ser un enfermero.

- El atributo “preferida” de “Dirección” es un booleano que indica la dirección principal o preferida para envío de notificaciones.
- “LíneaTratamiento” es una clase asociativa entre “Tratamiento” y “Medicamento”, que tendrán, necesariamente, en el modelo relacional una asociación “de muchos a muchos”. Esta “LíneaTratamiento” contiene cada uno de los medicamentos que debe tomar con su posología dividida en cantidad / periodicidad, por ejemplo “5 gotas cada 24 horas”, así como la duración en días que tiene, que no tiene que coincidir con el tratamiento completo.

### 2.2.2. Modelo Entidad-Relación

Mediante técnicas de ingeniería del *software* realizaremos una primera conversión desde el modelo estático de análisis hacia un modelo relacional.

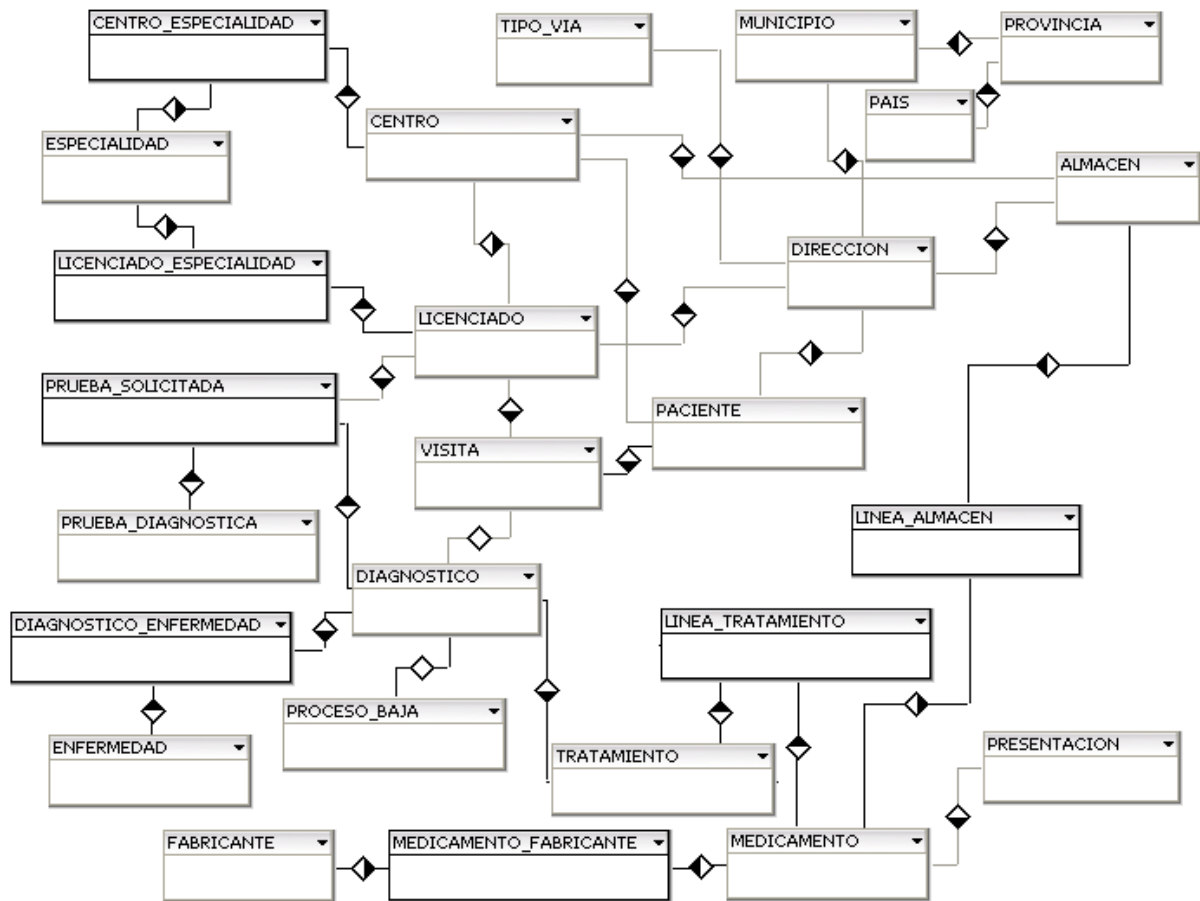


Ilustración 9: Modelo Entidad-Relación a Nivel Entidad (BD)

### Decisiones de diseño

- Las sub-entidades “CentroHospitalario” y “Farmacia” pasarán sus atributos a la clase abstracta “Centro”, que es la que queda en el modelo relacional definitivo, ya que las asociaciones que ambas tienen con otras entidades así lo permiten. Para indicar si el centro, ahora genérico, se trata de una farmacia, se incluirá un tipo más en los valores de “tipoCentroHospitalario”, que ahora será “tipoCentro”, que indique “Farmacia”.
- Las sub-entidades que son especializaciones de la entidad general abstracta “Persona” quedarán de la siguiente manera:
  - ✓ “Médico” y “Farmacéutico” comparten una misma entidad llamada “Licenciado”, que deberá incluir un campo para identificar el tipo de personal de que se trata, “Médico” o “Farmacéutico”. Además este campo podría servir para identificar otro tipo de personal si queremos ampliar más adelante. El identificador único de esta entidad puede ser “nif”.
  - ✓ “Paciente” pasa a tener su propia entidad “PACIENTE”. El identificador único de esta entidad no puede ser “nif”, sino que necesitará un identificador auto incremental, ya que los pacientes niños pueden no tener NIF en el momento de la asistencia.
  - ✓ De modo que los atributos de la entidad abstracta son copiados a ambos “Licenciado” y “Paciente”.
- Las entidades que se relacionan mediante una multiplicidad “\*” a ambos lados han necesitado una entidad relacional débil que incluya, al menos, los identificadores de su entidades fuertes.
- Valores XLAT: existen atributos cuyo contenido debe validarse contra una lista de valores válidos, como por ejemplo el “motivo” de “ProcesoBaja” (“riesgo por embarazo”, “riesgo por lactancia”, “psicológico”, “inmovilidad”, etc.).

La solución propuesta para estos casos se basa en la economía de diseño, de forma que en lugar de existir una tabla de valores válidos para cada uno de estos casos (como sí existe en “TipoVía”) será una tabla común de valores para todos ellos, en donde se indique la tabla, el campo, el valor que adopta y la descripción que le corresponde a ese valor, que será la que aparecerá en lugar del propio valor numérico, de forma similar al comando XLAT del lenguaje ensamblador. Por ello, llamaremos a esta solución VALORES\_XLAT.

- Para los registros de auditoría, que analizamos en los requerimientos, se propone una tabla AUDITORIA que registre datos de la acción lanzada, como fecha y hora, nombre del procedimiento lanzado, parámetros de entrada y salida lanzada por el sistema.



### 2.2.3. Diseño Lógico

Con una primera abstracción del modelo relacional a nivel de entidad, podemos desarrollar los campos que formarán parte de cada una de estas entidades.

Primero mostraremos las entidades detectadas con los atributos que deben crearse en la base de datos, indicando, en negrita, el identificador único del registro, y en cursiva los identificadores foráneos, que mostrará entre corchetes la tabla a la que hace referencia.

CENTRO ( **CIF** , RAZON\_SOCIAL , CORREO\_E , TELEFONO1 , TELEFONO2 , TIPO\_CENTRO , HORARIO , NUM\_HABITACIONES , NUM\_UCI , NUM\_UVI , NUM\_BOXES , NUM QUIROFANOS )

PAIS ( **ID\_PAIS** , NOMBRE )

PROVINCIA ( **ID\_PROVINCIA** , *ID\_PAIS [PAIS]* , NOMBRE )

MUNICIPIO ( **ID\_MUNICIPIO** , *ID\_PROVINCIA [PROVINCIA]* , NOMBRE , CODIGO\_POSTAL )

TIPO\_VIA ( **ID\_TIPO\_VIA** , DESCRIPCION )

LICENCIADO ( **NIF** , *CIF [CENTRO]* , NOMBRE , APELLIDO1 , APELLIDO2 , TELEFONO1 , TELEFONO2 , CORREO\_E , DIRECTOR , NUM\_COLEGIADO , TIPO\_PERSONAL )

ESPECIALIDAD ( **ID\_ESPECIALIDAD** , NOMBRE )

LICENCIADO\_ESPECIALIDAD ( *NIF [LICENCIADO]* , *ID\_ESPECIALIDAD [ESPECIALIDAD]* )

CENTRO\_ESPECIALIDAD ( *CIF [CENTRO]* , *ID\_ESPECIALIDAD [ESPECIALIDAD]* )

PACIENTE ( **ID\_PACIENTE** , NIF , NOMBRE , APELLIDO1 , APELLIDO2 , TELEFONO1 , TELEFONO2 , CORREO\_E , NASS , FECHA\_NACIMIENTO , FECHA\_DEFUNCION , SEXO , CONTACTO , NACIONALIDAD , *CIF [CENTRO]* )

VISITA ( **ID\_PACIENTE [PACIENTE]** , **FECHA** , TIPO\_VISITA , SINTOMAS , DOLENCIAS , ES\_RECAIDA , *NIF [LICENCIADO]* )

PRESENTACION ( **ID\_PRESENTACION** , FORMATO , CANTIDAD , UNIDAD\_MEDIDA )

MEDICAMENTO ( **ID\_MEDICAMENTO** , NOMBRE , PRINCIPIO\_ACTIVADO , COMPOSICION , *ID\_PRESENTACION [PRESENTACION]* )

FABRICANTE ( **ID\_FABRICANTE** , NOMBRE )

MEDICAMENTO\_FABRICANTE ( **ID\_MEDICAMENTO [MEDICAMENTO]** , **ID\_FABRICANTE [FABRICANTE]** )

ALMACEN ( **ID\_ALMACEN** , NOMBRE , CIF [CENTRO])

LINEA\_ALMACEN ( **ID\_ALMACEN** [ALMACEN] , **ID\_MEDICAMENTO** [MEDICAMENTO] , CANTIDAD )

DIAGNOSTICO ( **ID\_DIAGNOSTICO** , FECHA [VISITA] , **ID\_PACIENTE** [VISITA] , JUICIO , COMENTARIOS )

ENFERMEDAD ( **ID\_ENFERMEDAD** , NOMBRE , GRAVEDAD , RIESGO\_CONTAGIO , DESCRIPCION )

DIAGNOSTICO\_ENFERMEDAD ( **ID\_DIAGNOSTICO** [DIAGNOSTICO] , **ID\_ENFERMEDAD** [ENFERMEDAD] )

TRATAMIENTO ( **ID\_DIAGNOSTICO** [DIAGNOSTICO] , FECHA\_INICIO , FECHA\_FIN , INSTRUCCIONES )

LINEA\_TRATAMIENTO ( **ID\_DIAGNOSTICO** [TRATAMIENTO] , **ID\_MEDICAMENTO** [MEDICAMENTO] , POSOLOGIA\_CANTIDAD , POSOLOGIA\_PERIODICIDAD , DURACION\_DIAS )

PRUEBA\_DIAGNOSTICA ( **ID\_PRUEBA\_DIAGNOSTICA** , NOMBRE , REQUERIMIENTOS , LIMITACIONES , INSTRUCCIONES )

PRUEBA\_SOLICITADA ( **ID\_DIAGNOSTICO** [DIAGNOSTICO] , **ID\_PRUEBA\_DIAGNOSTICA** [PRUEBA\_DIAGNOSTICA] , FECHA , RESULTADO , COMENTARIOS , NIF [LICENCIADO] )

Nota: el campo “NIF” hace referencia al médico responsable de realizar la prueba, no al solicitante, que ya queda implícito en el valor “NIF” de la entidad “VISITA”.

DIRECCION ( **ID\_PACIENTE** [PACIENTE] , NIF [LICENCIADO] , **ID\_ALMACEN** [ALMACEN] , NOMBRE\_VIA , **ID\_MUNICIPIO** [MUNICIPIO] , **ID\_TIPO\_VIA** [TIPO\_VIA] , NUMERO , BLOQUE , PISO , PUERTA , PREFERIDA )

PROCESO\_BAJA ( **ID\_DIAGNOSTICO** [DIAGNOSTICO] , TIPO , MOTIVO , FECHA\_INICIO , FECHA\_FIN , ES\_RECAIDA )

VALORES\_XLAT ( **TABLA** , **CAMPO** , **VALOR** , DESCRIPCION )

AUDITORIA ( FECHA\_HORA , PROCEDIMIENTO , ENTRADA , SALIDA )

### *Modelo relacional*

Una vez que hemos definido todos los atributos de las entidades de nuestro sistema, podemos realizar una representación gráfica del modelo entidad-relación (E/R), que se muestra a continuación:

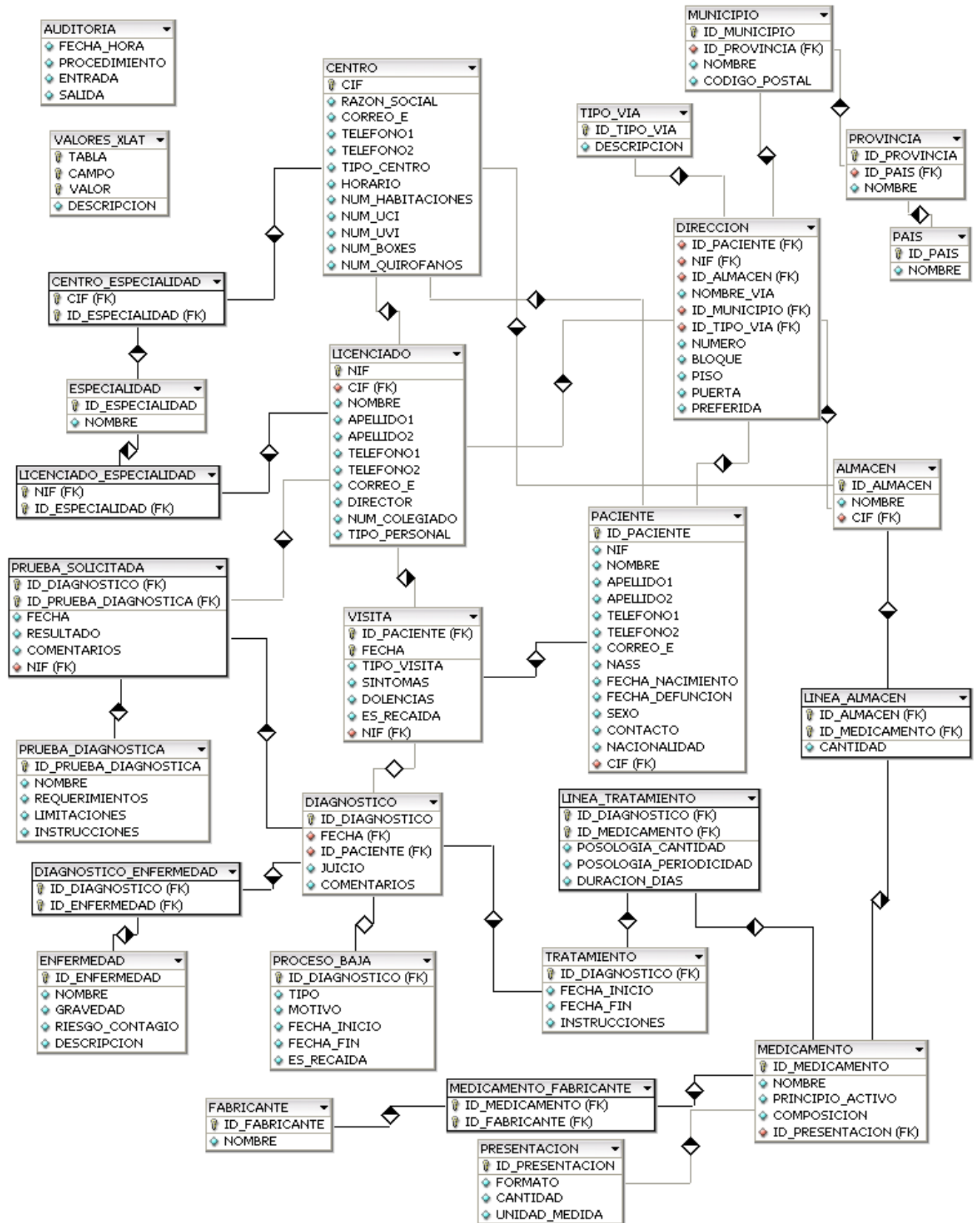


Ilustración 10: Modelo Entidad-Relación a Nivel de Atributos (BD)

### 2.2.4. Diseño Físico

Por último, podemos elaborar un diseño pensando en el esquema físico de base de datos, contemplando los tipos de datos que permite Oracle. Por consiguiente, este diagrama se corresponde con el anterior, pero incluye la definición del tipo de datos de cada atributo.

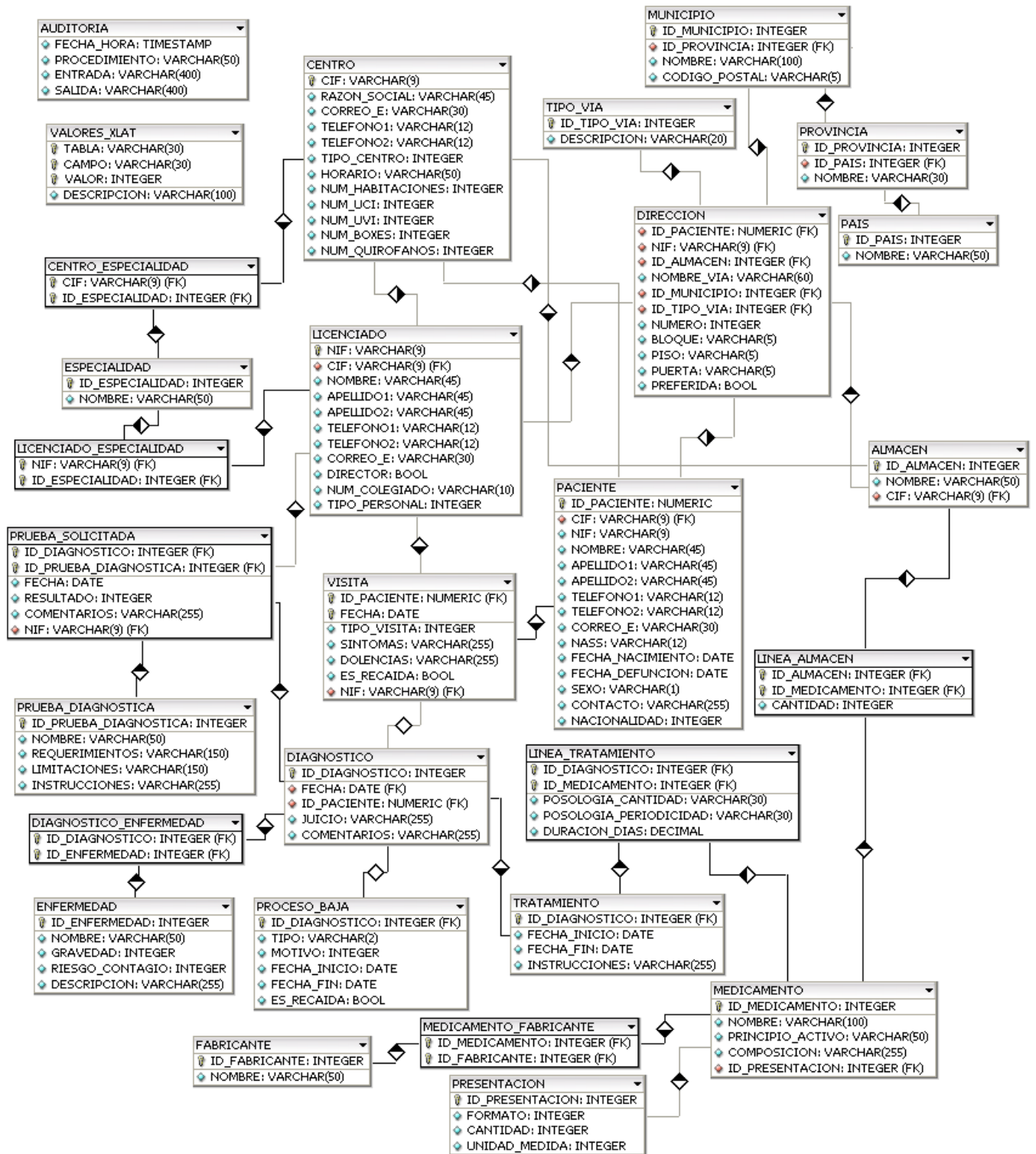


Ilustración 11: Modelo Entidad-Relación a Nivel de Esquema Físico (BD)

## 2.3. Implementación de la Base de Datos

Tras obtener el diseño físico detallado de la base de datos para el SGBD Oracle, concretamente en su versión 10g Express Edition, podemos codificar los *scripts* de creación de *tablespaces*, esquemas... hasta llegar al nivel de secuencias, disparadores... Veamos paso a paso este proceso de implementación.

### 2.3.1. Codificación de Scripts de Creación

Todos los *scripts* se encuentran en la carpeta “Producto” adjunta a esta memoria. Veamos cada uno de ellos.

#### *Tablespaces Para Tablas e Índices*

En el archivo 00\_TABLESPACES.sql tenemos las sentencias necesarias para crear nuestros propios espacios de tablas, para nuestro sistema, donde se almacenarán todos los elementos de la base de datos. Diferenciaremos un *tablespace* para las tablas y otro para los índices, cada uno con 60 MB. iniciales, que podrán incrementarse de forma automática gracias a la instrucción “EXTENT MANAGEMENT LOCAL AUTOALLOCATE”.

#### *Usuario Administrador del Sistema*

En el archivo 01\_USUARIO.sql se encuentra el *script* de creación del usuario “SANIDAD”, al que se le otorgan todos los permisos necesarios y se identifican los espacios de tabla por defecto. Su contraseña es “sanidad”.

#### *Creación de Tablas*

En el archivo 02\_CREACION\_TABLAS.sql se encuentra el *script* de creación de todas las tablas del sistema, con sus claves primarias, foráneas y restricciones. Además, debe tenerse en cuenta que las tablas e índices se crean en los espacios definidos a tal efecto, TS\_SANIDAD\_TABLAS y TS\_SANIDAD\_INDICES.

#### *Creación de Índices Adicionales*

A pesar de que la creación de tablas anterior ha creado los índices de las claves primarias de todas las tablas, tanto si son campos propios como si son foráneos, es interesante crear índices para todas las claves foráneas individuales que tenga la tabla, ya que esto puede mejorar considerablemente la velocidad del acceso a los datos. A tal efecto, en el archivo 03\_INDICES.sql encontramos la creación de todos estos índices adicionales.

### *Creación de Secuencias*

Para el caso de los identificadores que son auto incrementales, Oracle ofrece como solución la creación de secuencias, que vayan contando desde un valor inicial, típicamente 1, hacia adelante. En el archivo 04\_SECUENCIAS.sql encontramos las secuencias para cada una de las tablas que tienen una clave primaria auto incremental.

### *Creación de Disparadores*

Pero para que la inserción de un registro en una tabla con clave auto incremental tenga en cuenta el nuevo valor que debe asignar a dicha clave, debemos crear un disparador que compruebe en la secuencia cuál es el próximo valor, y lo asigne. Estos disparadores se encuentran en el archivo 05\_DISPARADORES.sql.

#### **2.3.2. Procedimientos de Acceso a Datos**

Tal y como se expone en la especificación de requisitos, toda la gestión y acceso a la información se hará mediante procedimientos de base de datos, siendo esta la única manera de acceder a ellos.

Para conseguir esto debemos emplear técnicas Oracle PL/SQL (Procedural Language/Structured Query Language), que encapsulan en procedimientos (*procedures*) códigos completos que combinan SQL con código de control, funciones, variables, control de flujo...

Una función interesante es %TYPE, que nos permitirá asignar el tipo correcto a los atributos de los procedimientos, para que coincidan con el tipo del atributo correspondiente en la tabla destino. Su funcionamiento es sencillo: %TYPE consulta el tipo en la tabla y lo asigna al atributo, así, no nos tenemos que preocupar de cambiar el procedimiento si, en algún momento posterior, modificamos el tipo en la tabla.

Por otro lado, haremos uso de las excepciones (*exception*), que controlarán el posible error SQL que se pueda producir, guardando un registro (*log*) del mismo en la tabla de auditoría y lanzará un mensaje (*RAISE\_APPLICATION\_ERROR*) que recibirá la aplicación o usuario que intentó la ejecución.

Estos procedimientos son fáciles de llamar desde aplicaciones externas, que recibirá una respuesta indicando si la acción que pretendía ha tenido éxito o no.

Dividiremos estos procedimientos en cuatro grupos:

1. Un procedimiento de inserción para cada una de las tablas.
2. Uno o varios procedimientos de consulta para cada una de las tablas.
3. Un procedimiento genérico para la modificación del valor de un campo de una tabla.
4. Un procedimiento genérico para la eliminación de un registro de una tabla.

### Procedimiento de Auditoría

En primer lugar crearemos un procedimiento almacenado que inserta un registro en la tabla de auditorías. Este procedimiento será llamado desde cualquier otro procedimiento, y será el encargado de dejar huella en la tabla AUDITORIA de lo que se ha intentado hacer, tanto si ha habido éxito como si no.

Podemos encontrar este procedimiento para las auditorías en el archivo 06\_PROCEDIMIENTO\_AUDITORIA.sql.

Su signatura es la siguiente:

Procedimiento	Entrada	Salida
<b>P_INSERTAR_AUDITORIA</b>	PROCEDIMIENTO ENTRADA SALIDA	Si correcto: ✓ COMMIT Si incorrecto por valor nulo: ✓ Lanza mensaje indicando que el campo no puede ser nulo Si incorrecto por otro motivo: ✓ Registro del error en tabla auditoría

Ilustración 12: Signatura del Procedimiento de Auditoría (BD)

Adicionalmente, el campo FECHA\_HORA se rellena con la fecha y hora del sistema (*SYSDATE*).

### Procedimientos de Inserción

Se ha creado un procedimiento de inserción para cada una de las tablas de la base de datos, incluso para el mantenimiento de la tabla VALORES\_XLAT y otras tablas maestras.

El mecanismo es el siguiente:

- ✓ Los parámetros de entrada son todos los campos que tiene la tabla destino, excepto los auto incrementales, que se dejan al disparador para que este lo rellene consultando la secuencia correspondiente. Como ya se ha mencionado, el tipo de dato es el exacto que podemos ver, en el modelo entidad-relación del apartado de “Diseño Físico”, ya que se obtiene mediante la función %TYPE.
- ✓ Se definen dos atributos internos, LOG\_ENTRADA y LOG\_SALIDA, que serán los valores a registrar en la tabla de auditoría.
- ✓ Se intenta la inserción en la tabla correspondiente:
  - ✓ Si tiene éxito, se confirma la inserción (*COMMIT*), registrando la acción en la tabla de auditoría.
  - ✓ Si no tiene éxito, no se realiza la inserción, pero se registra el error en la tabla de auditoría y se lanza una excepción.

En el archivo 07\_PROCEDIMIENTOS\_INSERTION.sql están todos los *scripts* de creación de estos procedimientos, declarados en el mismo orden que en la definición de atributos del apartado “Diseño Lógico”.

El listado completo de procedimientos de inserción y sus campos de entrada es el siguiente:

PROCEDIMIENTO	ENTRADA
P_INSERTAR_ALMACEN	NOMBRE CIF
P_INSERTAR_AUDITORIA	PROCEDIMIENTO ENTRADA SALIDA
P_INSERTAR_CENTRO	CIF RAZON_SOCIAL CORREO_E TELEFONO1 TELEFONO2 TIPO_CENTRO HORARIO NUM_HABITACIONES NUM_UCI NUM_UVI NUM_BOXES
P_INSERTAR_CENTRO_ESPECIALIDAD	CIF ID_ESPECIALIDAD
P_INSERTAR_DIAGNOSTICO	FECHA ID_PACIENTE JUICIO COMENTARIOS
P_INSERTAR_DIAGNOSTICO_ENFERM	ID_DIAGNOSTICO ID_ENFERMEDAD
P_INSERTAR_DIRECCION	ID_PACIENTE NIF ID_ALMACEN NOMBRE_VIA ID_MUNICIPIO ID_TIPO_VIA NUMERO BLOQUE PISO PUERTA PREFERIDA
P_INSERTAR_ENFERMEDAD	NOMBRE GRAVEDAD RIESGO_CONTAGIO DESCRIPCION
P_INSERTAR_ESPECIALIDAD	NOMBRE
P_INSERTAR_FABRICANTE	NOMBRE
P_INSERTAR_LICENCIADO	NIF CIF NOMBRE APELLIDO1 APELLIDO2

PROCEDIMIENTO	ENTRADA
P_INSERTAR_LINEA_TRATAMIENTO	ID_DIAGNOSTICO ID_MEDICAMENTO POSOLOGIA_CANTIDAD POSOLOGIA_PERIODICIDAD DURACION_DIAS
P_INSERTAR_MEDICAMENTO	NOMBRE PRINCIPIO_ACTIVO COMPOSICION ID_PRESENTACION
P_INSERTAR_MEDICAMENTO_FABRICA	ID_MEDICAMENTO ID_FABRICANTE
P_INSERTAR_MUNICIPIO	ID_PROVINCIA NOMBRE CODIGO_POSTAL
P_INSERTAR_PACIENTE	NIF CIF NOMBRE APELLIDO1 APELLIDO2 TELEFONO1 TELEFONO2 CORREO_E NASS FECHA_NACIMIENTO FECHA_DEFUNCION
P_INSERTAR_PAIS	NOMBRE
P_INSERTAR_PRESENTACION	FORMATO CANTIDAD UNIDAD_MEDIDA
P_INSERTAR_PROCESO_BAJA	ID_DIAGNOSTICO TIPO MOTIVO FECHA_INICIO FECHA_FIN ES_RECAIDA
P_INSERTAR_PROVINCIA	ID_PAIS NOMBRE
P_INSERTAR_PRUEBA_DIAGNOSTICA	NOMBRE REQUERIMIENTOS LIMITACIONES INSTRUCCIONES
P_INSERTAR_PRUEBA_SOLICITADA	ID_DIAGNOSTICO ID_PRUEBA_DIAGNOSTICA FECHA RESULTADO COMENTARIOS



	TELEFONO1		NIF
	TELEFONO2	<b>P_INSERTAR_TIPO_VIA</b>	DESCRIPCION
	CORREO_E	<b>P_INSERTAR_TRATAMIENTO</b>	ID_DIAGNOSTICO
	DIRECTOR		FECHA_INICIO
	NUM_COLEGIADO		FECHA_FIN
	TIPO_PERSONAL		INSTRUCCIONES
<b>P_INSERTAR_LICENCIADO_ESPECIAL</b>	NIF	<b>P_INSERTAR_VALORES_XLAT</b>	TABLA
	ID_ESPECIALIDAD		CAMPO
<b>P_INSERTAR_LINEA_ALMACEN</b>	ID_ALMACEN		VALOR
	ID_MEDICAMENTO		DESCRIPCION
	CANTIDAD	<b>P_INSERTAR_VISITA</b>	ID_PACIENTE
			FECHA
			TIPO_VISITA
			SINTOMAS
			DOLENCIAS
			ES_RECAIDA
			NIF

Ilustración 13: Listado de los Procedimientos de Inserción

### Procedimiento de Borrado

La acción de borrado de un registro se llevará a cabo mediante un procedimiento almacenado que recibe el nombre de la tabla de la que se quiere eliminar una fila y los campos y valores que lo identifican inequívocamente, su clave primaria.

Dado que en nuestra base de datos hay tablas con claves primarias compuestas por hasta tres campos, necesitamos todos ellos para identificar el registro a borrar. En caso de borrar una fila de una tabla con menos campos en la clave primaria, debemos establecer NULL en los campos que no usamos.

Veamos un ejemplo de borrado de una tabla con una clave primaria compuesta por un solo campo:

```
EXECUTE P_BORRAR_REGISTRO ('PAIS', 'ID_PAIS', '3', NULL, NULL, NULL, NULL)
```

Y ahora una con hasta tres campos en su clave primaria:

```
EXECUTE P_BORRAR_REGISTRO ('VALORES_XLAT', 'TABLA', 'PRESENTACION', 'CAMPO', 'UNIDAD_MEDIDA', 'VALOR', '2')
```

Podemos encontrar el procedimiento de borrado en el archivo 08\_PROCEDIMIENTO\_BORRADO.sql. Su signatura es la siguiente:

Procedimiento	Entrada	Salida
<b>P_BORRAR_REGISTRO</b>	TABLA VARCHAR, PK1 VARCHAR, ID1 VARCHAR, PK2 VARCHAR, ID2 VARCHAR, PK3 VARCHAR, ID3 VARCHAR	Si correcto: ✓ COMMIT ✓ Registro correcto en auditoría Si incorrecto: ✓ Registro del error en tabla auditoría ✓ Se lanza una excepción

Ilustración 14: Signatura del Procedimiento de Borrado

### Procedimiento de Modificación

Al igual que en la solución del borrado, la acción de modificación se llevará a cabo mediante un procedimiento almacenado que recibe el nombre de la tabla, el campo y el nuevo valor que se desea establecer. Del mismo modo, recibe los campos y valores que lo identifican inequívocamente, su clave primaria.

Otra vez, dado que en nuestra base de datos hay tablas con claves primarias compuestas por hasta tres campos, necesitamos todos ellos para identificar el registro a modificar. En caso de modificar una fila de una tabla con menos campos en la clave primaria, debemos establecer NULL en los campos que no usamos.

Veamos un ejemplo de modificación de una tabla con una clave primaria compuesta por un solo campo:

```
EXECUTE P_MODIFICAR_REGISTRO ('PAIS', 'NOMBRE', 'Noruega', 'ID_PAIS', '44', NULL, NULL, NULL, NULL)
```

Y ahora una con hasta tres campos en su clave primaria:

```
EXECUTE P_MODIFICAR_REGISTRO ('VALORES_XLAT', 'DESCRIPCION', 'ml', 'TABLA', 'PRESENTACION', 'CAMPO', 'UNIDAD_MEDIDA', 'VALOR', '2')
```

Para evitar errores con el tipo de dato DATE en este procedimiento, se ha creado la función F\_TIPO\_DATO, que nos indicará si el campo es de tipo fecha, para poder convertirlo y tratarlo adecuadamente.

Podemos encontrar el procedimiento de modificación y la función F\_TIPO\_DATO en el archivo 09\_PROCEDIMIENTO\_MODIFICACION.sql. Su signatura es la siguiente:

Procedimiento	Entrada	Salida
<b>P_MODIFICAR_REGISTRO</b>	TABLA VARCHAR, CAMPO NVARCHAR2, VALOR NVARCHAR2, PK1 VARCHAR, ID1 VARCHAR, PK2 VARCHAR, ID2 VARCHAR, PK3 VARCHAR, ID3 VARCHAR	Si correcto: ✓ COMMIT ✓ Registro correcto en auditoría Si incorrecto: ✓ No se realiza la modificación ✓ Registro del error en tabla auditoría ✓ Se lanza una excepción
<b>F_TIPO_DATO</b> (función)	TABLA IN NVARCHAR2, CAMPO IN NVARCHAR2	Devuelve en un VARCHAR2 el tipo de dato del campo consultado

Ilustración 15: Signatura del Procedimiento de Modificación

### Procedimientos de Consulta

Para los procedimientos de consulta necesitamos que la aplicación externa solicitante declare una variable del tipo de fila (*ROWTYPE*) de la tabla que va a consultar y la envíe a la función junto con el valor de la clave del registro que va a consultar, lo que nos devolvería el registro coincidente. Veamos un ejemplo:

```
DECLARE
    REGISTRO_CENTRO CENTRO%ROWTYPE;
BEGIN
    P_CONS_CENTRO_CIF ('A4128', REGISTRO_CENTRO);
    dbms_output.put_line (REGISTRO_CENTRO.CIF || ', ' || REGISTRO_CENTRO.RAZON_SOCIAL || ', ' ||
        REGISTRO_CENTRO.TELEFONO1);
END;
```

Esto devolvería todos los campos del centro cuyo CIF es el “A4128”.

La signatura de los procedimientos de consulta creados es la siguiente:

PROCEDIMIENTO	ENTRADA	SALIDA
P_CONS_ALMACEN_ID_ALMACEN	ID_ALMACEN	ALMACEN@ROWTYPE
P_CONS_CENTRO_CIF	CIF	CENTRO@ROWTYPE
P_CONS_ESPECIALIDAD_ID_ESPECI	ID_ESPECIALIDAD	ESPECIALIDAD@ROWTYPE
P_CONS_FABRICANTE_ID_FABR	ID_FABRICANTE	FABRICANTE@ROWTYPE
P_CONS_LICENCIADO_NIF	NIF	LICENCIADO@ROWTYPE
P_CONS_MEDICAMENTO_ID_MEDI	ID_MEDICAMENTO	MEDICAMENTO@ROWTYPE
P_CONS_PACIENTE_ID_PACIENTE	ID_PACIENTE	PACIENTE@ROWTYPE
P_CONS_PROCESO_BAJA_ID_DIAGNOS	ID_DIAGNOSTICO	PROCESO_BAJA@ROWTYPE
P_CONS_PRUEBA_DIAGNOSTICA_ID_P	ID_PRUEBA_DIAGNOSTICA	PRUEBA_DIAGNOSTICA@ROWTYPE
P_CONS_TRATAMIENTO_ID_DIAGNOST	ID_DIAGNOSTICO	TRATAMIENTO@ROWTYPE
P_CONS_VISITA_ID_PACIEN_FECHA	ID_PACIENTE FECHA	VISITA@ROWTYPE

Ilustración 16: Signatura de los Procedimientos de Consulta

Pero en los casos de uso dimos solución a los requerimientos de consulta sobre el paciente, como listar los tratamientos seguidos, días que ha estado de baja, etc. Para estos listados, se han realizado los siguientes procedimientos de consulta, que devuelven una salida DBMS, con valores separados por comas, que la aplicación externa podrá recibir y formatear.

Para ejecutar estos procedimientos, podemos lanzar esta instrucción:

```
BEGIN
    P_CONS_LIST_PACIENTES_CENTRO ('A4128'); -- Pacientes del centro con CIF 'A4128'
END;
```

La signatura de estos procedimientos es la siguiente:

PROCEDIMIENTO	OBJETIVO	ENTRADA	SALIDA
P_CONS_LIST_PACIENTE_DIAS_BAJA	Listar los días en que un paciente ha estado de baja.	ID_PACIENTE	ID_PACIENTE, TIPO, MOTIVO, FECHA_INICIO, FECHA_FIN, ES_RECAIDA
P_CONS_LIST_PACIENTE_ENFERMEDA	Listar las enfermedades que ha padecido un paciente.	ID_PACIENTE	ID_PACIENTE, FECHA, NOMBRE (de la enfermedad)
P_CONS_LIST_PACIENTE_MEDI_VISI	Listar los médicos que ha visitado un paciente.	ID_PACIENTE	ID_PACIENTE, FECHA, NIF, NOMBRE, APELLIDO1, APELLIDO2
P_CONS_LIST_PACIENTE_PRUEBAS	Listar las pruebas diagnósticas que se ha realizado un paciente.	ID_PACIENTE	ID_PACIENTE, NOMBRE, FECHA, RESULTADO, COMENTARIOS, NIF (del médico que hace la prueba)
P_CONS_LIST_PACIENTES_CENTRO	Listar todos los pacientes adscritos a un centro.	CIF	ID_PACIENTE , NIF , NOMBRE , APELLIDO1 , APELLIDO2 , TELEFONO1 , TELEFONO2 , CORREO_E , NASS , FECHA_NACIMIENTO , FECHA_DEFUNCION , SEXO , CONTACTO , NACIONALIDAD , CIF
P_CONS_LIST_PACIENTE_TTOS	Listar todos los tratamientos médicos que ha tenido un paciente.	ID_PACIENTE	ID_PACIENTE, FECHA_INICIO, FECHA_FIN, INSTRUCCIONES, NOMBRE (del medicamento), DURACION_DIAS
P_CONS_LIST_PACIENTE_VISITAS	Listar todas las visitas que ha realizado un paciente.	ID_PACIENTE	ID_PACIENTE, FECHA, TIPO_VISITA, SINTOMAS, DOLENCIAS, ES_RECAIDA, NIF (del médico)

Ilustración 17: Signatura de los Procedimientos de Listados

Podemos encontrar tanto los procedimientos de consultas por clave, como los de listados, en el archivo 10\_PROCEDIMIENTOS\_CONSULTAS.sql.

## 2.4. Pruebas

Tal y como quedó reflejado en la planificación del proyecto, a la finalización de la construcción del almacén de datos se ha reservado una semana para elaborar un juego de pruebas, tanto de la base de datos como del almacén. Por este motivo, las pruebas realizadas durante esta fase de implementación no se incluyen en este apartado, sino en la fase de “Pruebas y Testeo”.

## 3. ALMACÉN DE DATOS

Una vez construida la base de datos, y tal y como solicita el cliente, en este capítulo quedarán recogidas todas las fases (análisis, diseño e implementación) necesarias para la construcción del almacén de datos.

### 3.1. Análisis de Requisitos

La herramienta que requiere el cliente facilitará la presentación de la información dispersa en la base de datos de una forma más comprensible y útil para el usuario, concretamente aquel que debe tomar decisiones basadas en el estudio de la información que desprende el sistema.

#### 3.1.1. Descripción General

El cliente ha referido que necesita estadísticas que le permitan saber:

- ✓ En qué época del año hay más urgencias.
- ✓ A qué edad se consumen más medicamentos.
- ✓ Cuál es el tiempo medio que una persona está de baja (media anual).

Además, se propone, como ampliación:

- ✓ Provincias donde se consumen más medicamentos.
- ✓ Municipios en que se producen más visitas médicas.

#### *Programación ETL*

El almacén de datos, por su naturaleza, puede ser consultado de manera concurrente por muchos usuarios, que ralentizarían la carga de datos en el mismo si se hiciera en tiempo real. Por este motivo, se ha decidido que los procesos de extracción, transformación y carga (ETL, del inglés “*Extract, Transform, Load*”) que cargarán los datos serán programados una vez al día y diferidos durante la noche, cuando no hay tanta demanda de información.

Todos los datos que se cargan tendrán relación con la tabla VISITA, ya que serán los registros de tablas que giran alrededor de las visitas médicas (como diagnósticos, pruebas...) del día anterior al momento de la carga los que deben ser extraídos, transformados y cargados en el almacén de datos. Por ejemplo, el día 6/3/2013, de madrugada, se cargarán todos los datos generados el día 5/6/2013, y así sucesivamente.

### 3.1.2. Diagramas de Casos de Uso

Podemos representar mediante notación UML un sencillo diagrama de estos requerimientos antes de pasar a su especificación textual más detallada.

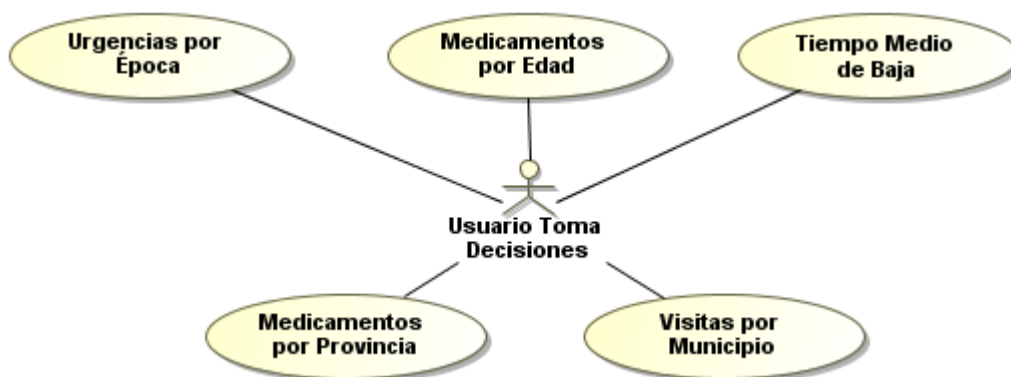


Ilustración 18: Casos de Uso (DW)

### 3.1.3. Especificación Textual

Pasamos a la descripción textual de los casos de uso del almacén de datos de nuestro sistema.

<b>Caso de uso</b>	<b>Urgencias por Época</b>
<b>Descripción</b>	Muestra un listado con las épocas del año (por trimestre) y el número de visitas de tipo urgente que se han producido en tal época
<b>Actor</b>	Usuario Toma Decisiones
<b>Precondición</b>	La tabla de visitas contiene datos y los campos fecha, tipo y paciente está correctamente cumplimentados
<b>Postcondición</b>	No
<b>Parámetros</b>	Siempre se muestra el listado completo, por lo que no se admiten parámetros
<b>Salida</b>	Estadísticas de urgencias por época del año
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para la consulta de esta estadística
<b>Flujo interacción</b>	Se consulta la tabla del almacén de datos donde se guardan las estadísticas de este tipo Si existen: se devuelven los datos solicitados se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existen: se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

Caso de uso	Medicamentos por Edad
Descripción	Muestra un listado agrupando los pacientes por sus edades, y se muestra el número de medicamentos consumidos por la totalidad de ellos en cada grupo, de manera que se puede identificar a qué edad se consumen más medicamentos
Actor	Usuario Toma Decisiones
Precondición	Las tablas de visitas, diagnósticos y tratamientos (con sus líneas) contienen datos. Los campos a consultar son claves, por lo que siempre estarán cumplimentados
Postcondición	No
Parámetros	Siempre se muestra el listado completo, por lo que no se admiten parámetros
Salida	Estadísticas de edades y número de medicamentos consumidos
Disparador	Se ejecuta el procedimiento almacenado para la consulta de esta estadística
Flujo interacción	Se consulta la tabla del almacén de datos donde se guardan las estadísticas de este tipo Si existen: se devuelven los datos solicitados se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existen: se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
Auditoría	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

Caso de uso	Tiempo Medio de Baja
Descripción	Muestra un listado con cada año y el número de días promedio que los pacientes (totales) están de baja. Así se puede observar si el número medio de días de baja asciende o desciende debido de causas aparentemente no relacionadas, por ejemplo en crisis económica
Actor	Usuario Toma Decisiones
Precondición	Las tablas de visitas, diagnósticos y procesos de baja contienen datos y los campos a consultar están cumplimentados correctamente
Postcondición	No
Parámetros	Siempre se muestra el listado completo, por lo que no se admiten parámetros
Salida	Estadísticas de tiempo medio que una persona está de baja (media anual)
Disparador	Se ejecuta el procedimiento almacenado para la consulta de esta estadística
Flujo interacción	Se consulta la tabla del almacén de datos donde se guardan las estadísticas de este tipo Si existen: se devuelven los datos solicitados se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existen: se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
Auditoría	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Medicamentos por Provincia</b>
<b>Descripción</b>	Muestra un listado con el número de medicamentos que se han prescrito en cada provincia, ya sea al año, al trimestre o al mes
<b>Actor</b>	Usuario Toma Decisiones
<b>Precondición</b>	Las tablas de visitas, diagnósticos, tratamientos y sus líneas están cumplimentados correctamente
<b>Postcondición</b>	No
<b>Parámetros</b>	Se debe seleccionar si el listado será por año, trimestre o mes
<b>Salida</b>	Estadísticas del número de medicamentos consumidos en cada provincia
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para la consulta de esta estadística
<b>Flujo interacción</b>	Se consulta la tabla del almacén de datos donde se guardan las estadísticas de este tipo Si existen:        se devuelven los datos solicitados se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existen:    se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no

<b>Caso de uso</b>	<b>Visitas por Municipio</b>
<b>Descripción</b>	Muestra un listado con el número de visitas médicas atendidas en cada municipio
<b>Actor</b>	Usuario Toma Decisiones
<b>Precondición</b>	Las tablas de visitas, pacientes, direcciones y sus tablas maestras contienen datos
<b>Postcondición</b>	No
<b>Parámetros</b>	Se debe seleccionar si el listado será por año, trimestre o mes
<b>Salida</b>	Estadísticas de número de visitas médicas en cada municipio
<b>Disparador</b>	Se ejecuta el procedimiento almacenado para la consulta de esta estadística
<b>Flujo interacción</b>	Se consulta la tabla del almacén de datos donde se guardan las estadísticas de este tipo Si existen:        se devuelven los datos solicitados se registra la acción en la tabla de auditoría ( <i>logs</i> )  Si no existen:    se devuelve un mensaje de error SQL explicando lo ocurrido se registra la acción en la tabla de auditoría ( <i>logs</i> )
<b>Auditoría</b>	Esta acción crea un registro en la tabla de auditoría, indicando lo ocurrido, tanto si se tiene éxito como si no



#### 3.1.4. Alternativa: Soluciones OLAP

En este análisis de requisitos cabe mencionar, aunque sea de manera breve y superficial, que para un sistema de dimensiones tan elevadas como es el sistema de información sanitario del Ministerio, se podría disponer de un servidor dedicado al procesamiento analítico en línea, más potente que el almacén de datos que aquí se pueda desarrollar.

Para ello existen productos comerciales como Microsoft Analysis Services, Oracle Essbase, Oracle Database OLAP Option, SAP NetWeaver BW, etc. e incluso software libre como Jedox Palo o Pentaho Mondrian OLAP server.

Mediante este tipo de soluciones se pueden desarrollar cuadros de control de negocio y de organización de manera más avanzada y especializada, ya que incluyen diversos modos de almacenamiento y de explotación de datos (MOLAP, ROLAP, HOLAP...). Cabe destacar la herramienta desarrollada por Microsoft, PowerPivot, que puede funcionar temporalmente sin conexión a la base de datos y que se integra con Excel, lo que da un paso a favor del usuario final no informático.

Sin embargo, no implantaremos ninguna de estas herramientas, aunque sí deben ser tenidas en cuenta en un proyecto real con más recursos temporales, económicos y humanos.

En síntesis, existen soluciones profesionales para la ayuda a la toma de decisiones, si bien en el presente proyecto se propondrá un almacén de datos más sencillo, que puede ser ampliable en un futuro según surjan nuevas necesidades.

### 3.2. Diseño Técnico

Para dar forma al almacén de datos, debemos definir sus dimensiones, entendiendo estas como los ejes del análisis que debemos ofrecer. Asimismo, definiremos las tablas de hechos mensurables (número de visitas, número de urgencias, número de medicamentos, número de días en baja, etc.).

#### 3.2.1. Dimensiones y Hechos

Las dimensiones encontradas como necesarias para satisfacer los requerimientos observados en los casos de uso son:

Dimensión	Jerarquía	Rango Valores
<b>TIEMPO</b>	Año	2013 en adelante
	Trimestre	1 - 4
	Mes	1 - 12
	Semana	1 - 52
	Día	Cualquier fecha
<b>ZONA</b>	País	Tabla PAIS
	Provincia	Tabla PROVINCIA
	Municipio	Tabla MUNICIPIO
<b>EDAD</b>	Grupo	0 a 2 años (bebés) 3 a 12 años (niños) 13 a 18 años (adolescentes) 19 a 26 años (jóvenes) 27 a 59 años (adultos) 60 en adelante (mayores)
	Años	0 en adelante

Ilustración 19: Dimensiones del Almacén de Datos

Por otro lado, los hechos mensurables son:

Hecho	Indicador	Descripción
<b>VISITAS</b>	totalVisitas	Número total de visitas recibidas
	numOrdinarias	Número de vistas ordinarias recibidas
	numUrgentes	Número de vistas urgentes recibidas
<b>BAJAS</b>	numDiasTotales	Número total de días de baja de todos los pacientes de esta dimensión
	numDiasIT	Número de días de baja de tipo IT
	numDiasAT	Número de días de baja de tipo AT
<b>MEDICAMENTOS</b>	numMedicamentos	Número de medicamentos prescritos a todos los pacientes de esta dimensión

Ilustración 20: Hechos del Almacén de Datos

Como veremos en el modelo relacional, los hechos deben incluir como clave el valor del atributo de menor jerarquía de cada dimensión.

### 3.2.2. Diseño Conceptual en UML

Podemos representar las dimensiones y hechos en UML:

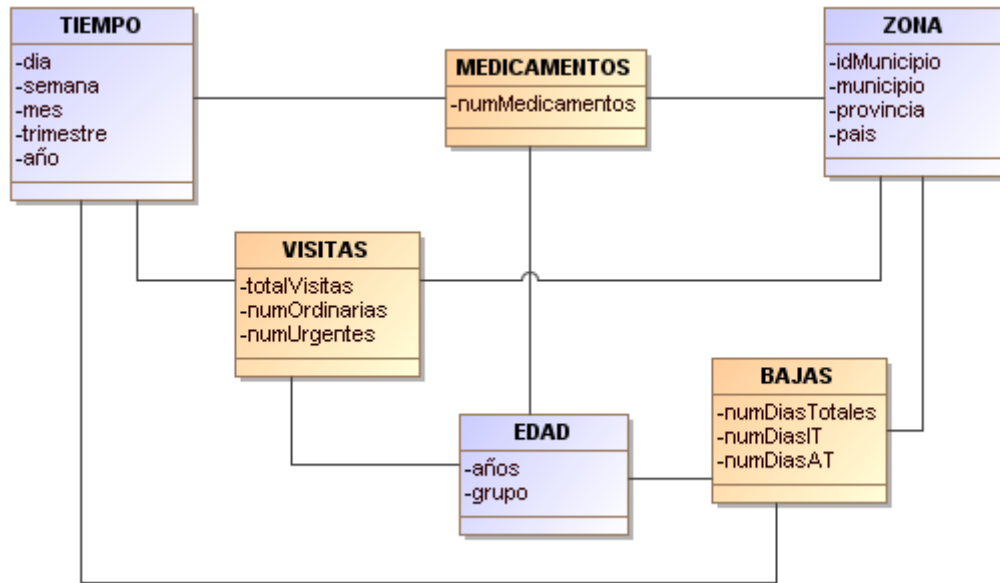


Ilustración 21: Diagrama Conceptual en UML (DW)

Cabe destacar que todos los hechos son consultables desde todas las dimensiones.

### 3.2.3. Modelo Entidad-Relación

La aproximación del modelo estático al modelo relacional es la siguiente.

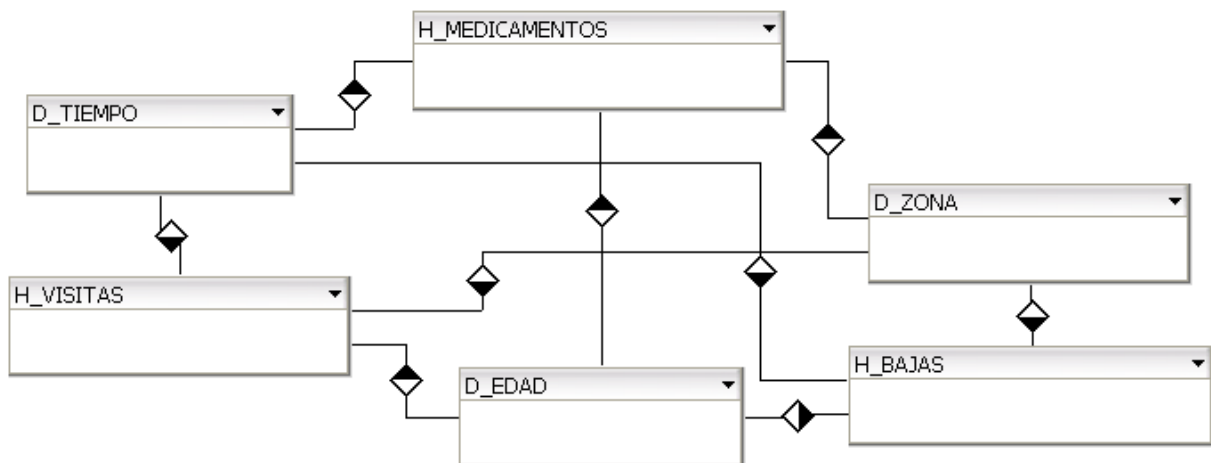


Ilustración 22: Modelo Entidad-Relación a Nivel Entidad (DW)

Nomenclatura: las tablas "dimensión" tienen el prefijo "D\_", mientras que las que contienen los "hechos" tienen "H\_".

### 3.2.4. Diseño Lógico

En este caso, obtenemos de manera trivial el modelo E/R a partir del nivel entidad y del diagrama UML presentados más arriba.

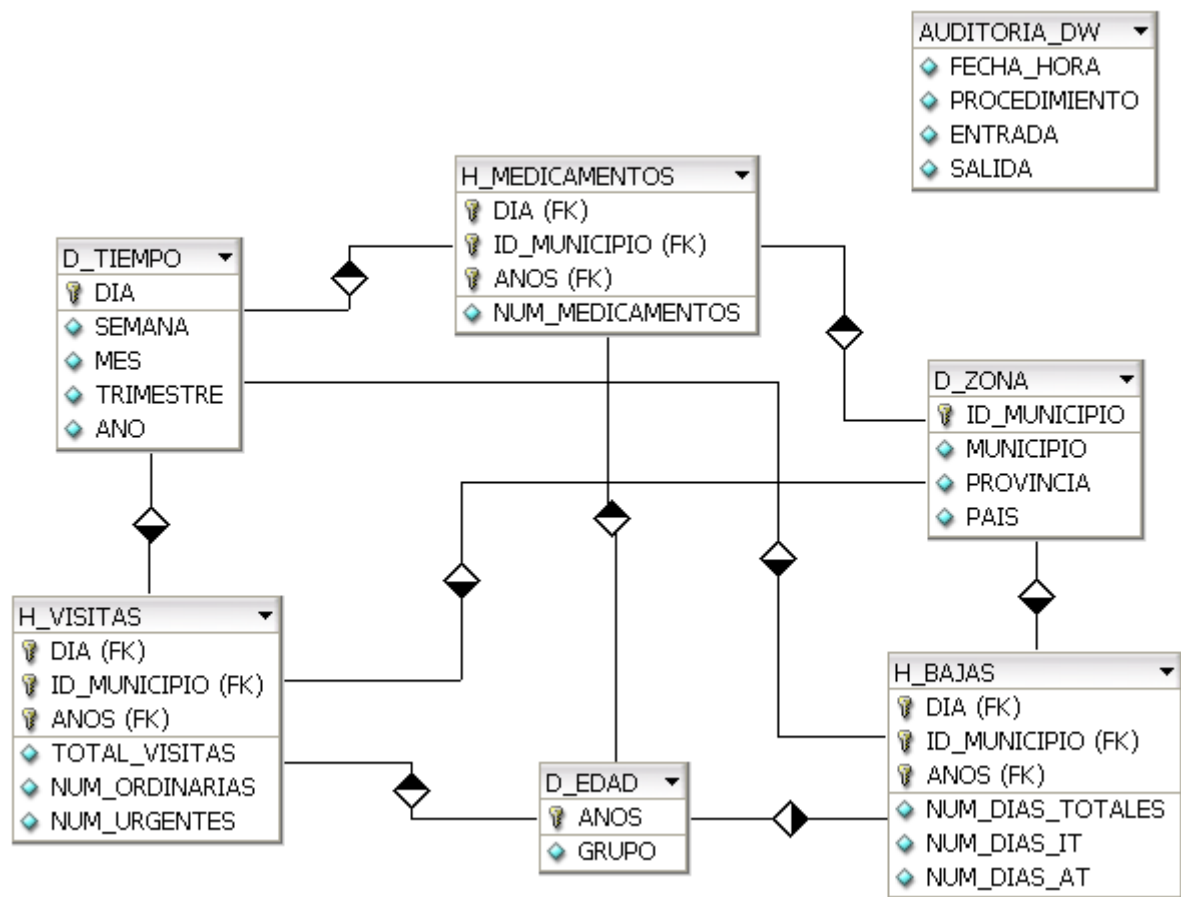


Ilustración 23: Modelo Entidad-Relación a Nivel de Atributos (DW)

### 3.2.5. Diseño Físico

Por último, al igual que en el diseño de la base datos, podemos elaborar el esquema físico del almacén de datos, contemplando los tipos de datos que permite Oracle. Por consiguiente, este diagrama se corresponde con el anterior, pero incluye la definición del tipo de datos de cada atributo.

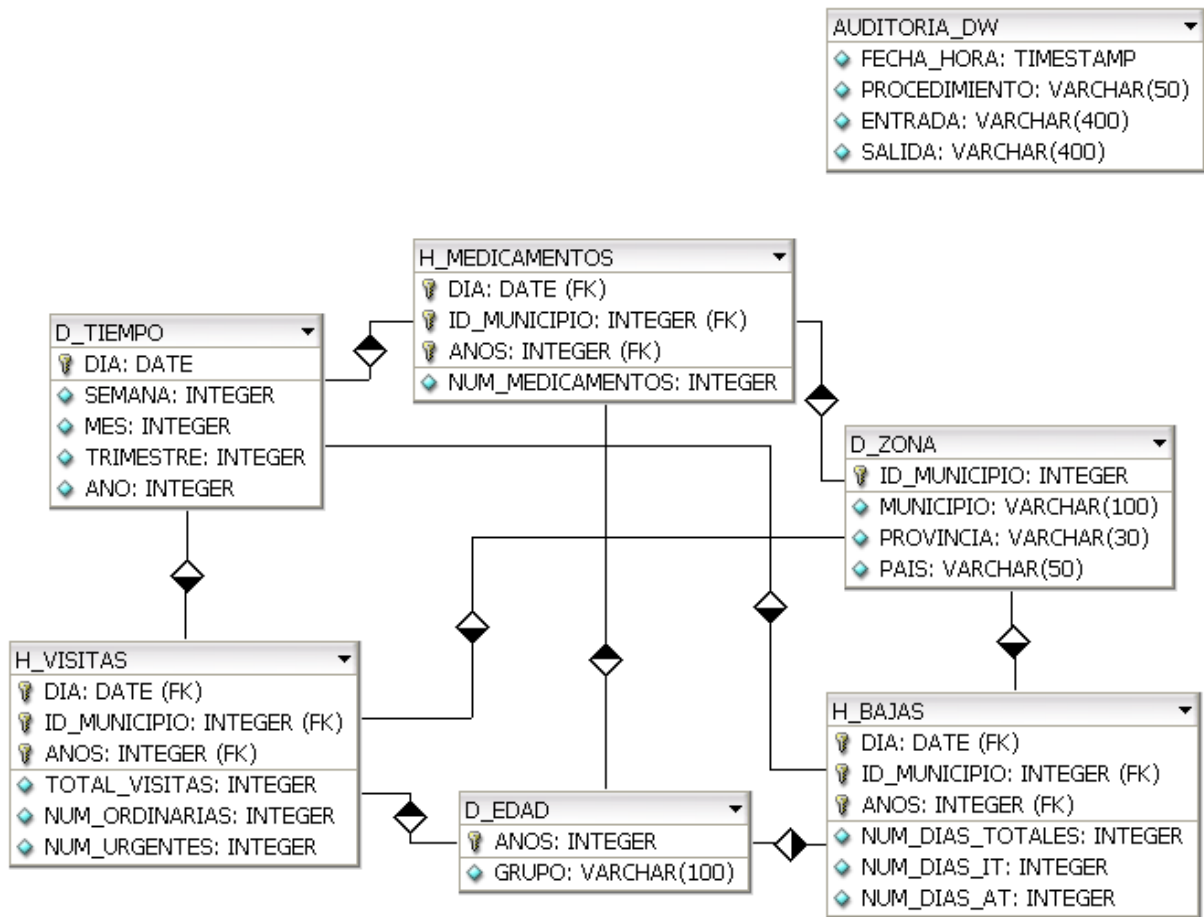


Ilustración 24: Modelo Entidad-Relación a Nivel de Esquema Físico (DW)

### 3.3. Implementación del Almacén de Datos

Ahora ya tenemos el diseño físico detallado de la base de datos para Oracle, en su versión 10g Express Edition. De este modo, podemos comenzar a codificar los *scripts* de creación de *tablespaces*, esquemas... que serán propios para el almacén de datos, es decir, no usaremos los mismos que en la base de datos. Por otro lado, no será necesario definir secuencias o disparadores. El proceso detallado de implementación se presenta a continuación.

#### 3.3.1. Codificación de Scripts de Creación

Todos los *scripts* se encuentran en la carpeta “Producto” adjunta a esta memoria. Veamos cada uno de ellos.

##### *Tablespaces Para Tablas e Índices*

En el archivo 11\_TABLESPACES\_DW.sql tenemos las sentencias necesarias para crear nuestros propios espacios de tablas, para el almacén de datos de nuestro sistema, donde se almacenarán todos los elementos de este. Diferenciaremos un *tablespace* para las tablas y otro para los índices, cada uno con 100 MB. iniciales, que podrán incrementarse de forma automática gracias a la instrucción “EXTENT MANAGEMENT LOCAL AUTOALLOCATE”.

##### *Usuario Administrador del Almacén de Datos*

En el archivo 12\_USUARIO\_DW.sql se encuentra el *script* de creación del usuario “SANIDAD\_DW”, al que se le otorgan todos los permisos necesarios sobre las tablas del esquema de la base de datos mediante un bucle que recorre todas las tablas cuyo propietario es SANIDAD y otorga permisos de selección (no son necesarios otros permisos) al usuario SANIDAD\_DW. Asimismo, se identifican los espacios de tabla por defecto. Su contraseña es “sanidad\_dw”.

##### *Creación de Tablas*

En el archivo 13\_CREACION\_TABLAS\_DW.sql se encuentra el *script* de creación de todas las tablas del almacén, con sus claves primarias, foráneas y restricciones. Además, debe tenerse en cuenta que las tablas e índices se crean en los espacios definidos a tal efecto, TS\_SANIDAD\_TABLAS\_DW y TS\_SANIDAD\_INDICES\_DW.

##### *Creación de Índices Adicionales*

A pesar de que la creación de tablas anterior ha creado los índices de las claves primarias de todas las tablas, tanto si son campos propios como si son foráneos, es interesante crear índices para todas las claves foráneas individuales que tenga la tabla, ya que esto puede mejorar considerablemente la velocidad del acceso a los datos, máxime considerando que se trata de un almacén de datos, donde la velocidad de acceso es uno de los principales objetivos. A tal efecto, en el archivo 14\_INDICES\_DW.sql encontramos la creación de todos estos índices adicionales.

### 3.3.2. Proceso de Carga de Datos

La carga de información desde la base de datos hasta el almacén debe realizarse mediante procedimientos almacenados (*stored procedures*), que serán los encargados de extraer, transformar y cargar (ETL) la información en las tablas que hemos preparado a tal efecto. De nuevo, para ello, se emplearán técnicas de Oracle PL/SQL.

Como veremos, se harán uso de excepciones y de registros en la tabla de auditoría del almacén (AUDITORIA\_DW), que es diferente a la de la base de datos.

#### *Procedimiento de Auditoría*

De forma similar al funcionamiento de la base de datos, definimos un procedimiento almacenado que crea un registro en la tabla de auditorías. Este procedimiento será llamado desde los procedimientos de carga y consulta, y será el encargado de dejar huella en la tabla AUDITORIA\_DW de lo que se ha intentado hacer, tanto si ha habido éxito como si no.

Podemos encontrar este procedimiento para las auditorías en el archivo 15\_PROCEDIMIENTO\_AUDITORIA\_DW.sql.

Su signatura es la siguiente:

Procedimiento	Entrada	Salida
P_INSERTAR_AUDITORIA_DW	PROCEDIMIENTO ENTRADA SALIDA	Si correcto: ✓ COMMIT Si incorrecto por valor nulo: ✓ Lanza mensaje indicando que el campo no puede ser nulo Si incorrecto por otro motivo: ✓ Registro del error en tabla auditoría

Ilustración 25: Signatura del Procedimiento de Auditoría (DW)

Adicionalmente, el campo FECHA\_HORA se rellena con la fecha y hora del sistema (SYSDATE).

#### *Procedimientos de Carga de Datos*

Todas las dimensiones y hechos del almacén de datos deben tener su procedimiento de carga de datos. Las dimensiones actúan como tablas maestras, que deben cargarse en primer lugar, ya que los hechos tienen claves foráneas cuyos valores deben existir en todas las dimensiones (recordemos que todos los hechos son consultables desde todas las dimensiones).

Veamos la tabla descriptiva con la lista de procedimientos y algunas observaciones pertinentes:

PROCEDIMIENTO	ACCIÓN	OBSERVACIONES
P_CARGAR_D_TIEMPO	Inserta en la tabla de dimensión D_TIEMPO un nuevo registro para la fecha del día anterior al momento de carga.	La granularidad mínima de consulta es por día, y la máxima por año.
P_CARGAR_D_ZONA	Inserta en la tabla de dimensión D_ZONA tantos registros como municipios diferentes agrupen a los pacientes que han asistido a consulta el día anterior al momento de carga.	La granularidad mínima de consulta es por municipio, y la máxima por país. Dado que un paciente puede tener varias direcciones, el municipio seleccionado para la carga es el indicado como “preferido”.
P_CARGAR_D_EDAD	Inserta en la tabla de dimensión D_EDAD tantos registros como edades diferentes agrupen a los pacientes que han asistido a consulta el día anterior al momento de carga.	La granularidad mínima de consulta es por edad en años, y la máxima por grupo de edad.
P_CARGAR_H_VISITAS	Inserta en la tabla de hechos H_VISITAS las estadísticas sobre visitas de los pacientes que han asistido a consulta el día anterior al momento de carga.	El campo que determina el tipo de consulta es “TIPO_VISITA”, cuya descripción se encuentra en la tabla VALORES_XLAT.
P_CARGAR_H_MEDICAMENTOS	Inserta en la tabla de hechos H_MEDICAMENTOS las estadísticas sobre medicamentos recetados a los pacientes que han asistido a consulta el día anterior al momento de carga.	Se cuentan los medicamentos diferentes recetados en la visita del día anterior a la carga.
P_CARGAR_H_BAJAS	Inserta en la tabla de hechos H_BAJAS las estadísticas sobre procesos de baja finalizados el día anterior al momento de carga.	A diferencia del resto de cargas, no se toma la fecha de visita, pues podrían no conocerse los días exactos de baja, por lo que tomamos la fecha de fin del proceso, que sí nos asegura el número de días.

Ilustración 26: Listado de los Procedimientos de Carga

**PARÁMETRO DE ENTRADA:** Opcionalmente, todos los procedimientos de carga admiten como entrada una fecha, para cargar en el almacén las estadísticas de un día concreto. Esto podría conllevar riesgos de duplicidad de información y solo lo debería realizar el administrador de la base de datos. El comportamiento natural y adecuado es ejecutar los procedimientos sin parámetros de entrada, diariamente.

En el archivo 16\_PROCEDIMIENTOS\_CARGA\_DW.sql están todos los *scripts* de creación de estos procedimientos de carga.

Por otro lado, en el archivo 17\_EJECUTAR\_CARGAS\_DW.sql tenemos el *script* que debe ejecutarse cada noche, solo una vez al día. Necesitaremos programar la tarea, que se hará de una u otra forma dependiendo del sistema operativo de la máquina.



### 3.3.3. Sentencias de Consulta de Estadísticas

Una vez cargado el almacén de datos se pueden consultar todos los hechos filtrando y agrupando a través de las dimensiones según interese en cada momento.

Las consultas se pueden hacer mediante sentencias SQL directas a la base de datos, mediante informes, *grids* de datos en interfaces de usuario, gráficas, cuadros de mando, etc.

En todo caso, cabe destacar que una de las opciones de consulta es mediante procedimientos almacenados, de la misma forma que el cliente solicitó la gestión de datos en el apartado de base de datos. Para ver esto, se han desarrollado los procedimientos a tal efecto en el archivo 18\_PROCEDIMIENTOS\_ESTADISTICAS\_DW.sql, que habría que imitar para el resto de casos de consulta de estadísticas si se determinara este modo de acceso a los datos.

La signatura de estos procedimientos es la siguiente:

PROCEDIMIENTO	OBJETIVO	ENTRADA	SALIDA
P_EST_URGENCIAS_EPOCA	Listar las visitas urgentes en cada trimestre del año.	No	ANO, TRIMESTRE, NUM_URGENTES
P_EST_MEDICAMENTOS_EDAD	Listar el número de medicamentos que se consume en cada grupo de edad.	No	GRUPO, NUM_MEDICAMENTOS
P_EST_TIEMPO_MEDIO_BAJA	Listar el promedio de tiempo de las bajas.	No	ANO, PROMEDIO_TIEMPO_BAJA

Ilustración 27: Signatura de los Procedimientos Estadísticos

También se pueden programar para recibir argumentos de entrada, como se ha hecho en la base de datos.

Para ejecutar estos procedimientos, podemos lanzar estas instrucciones:

```
BEGIN
  P_EST_URGENCIAS_EPOCA; -- Urgencias por Época
END;

BEGIN
  P_EST_MEDICAMENTOS_EDAD; -- Medicamentos por Edad
END;

BEGIN
  P_EST_TIEMPO_MEDIO_BAJA; -- Tiempo Medio de Baja
END;
```

## 4. PRUEBAS Y TESTEO

En este capítulo se desarrollarán las pruebas de funcionalidad tanto de la base de datos como del almacén de datos. Así, se crearán *scripts* de inserción, modificación, consulta y borrado empleando los propios procedimientos almacenados elaborados en el presente proyecto.

### 4.1. Juego de Pruebas

#### 4.1.1. Base de Datos

Para probar el correcto funcionamiento de cada una de las funcionalidades de la base de datos, se ha preparado el archivo 19\_PRUEBAS\_BD.sql, que realiza pruebas unitarias de todos los procedimientos de inserción y de un buen número de procedimientos de borrado, modificación y consulta. Además, se han insertado líneas con acciones que provocan error, para corroborar el correcto manejo de las excepciones.

Este *script* dejará registro en la tabla AUDITORIA, que será la que consultaremos para verificar que todo ha quedado registrado rigurosamente.

Aunque en el archivo 19\_PRUEBAS\_BD.sql están todas las pruebas, veamos aquí, como ejemplos, un par de capturas de pantalla de la salida DBMS que los procedimientos de listado ofrecen:

P\_CONS\_LIST\_PACIENTES\_CENTRO (pacientes adscritos a un centro):

```
BEGIN
    P_CONS_LIST_PACIENTES_CENTRO ('B6666');
END;
```

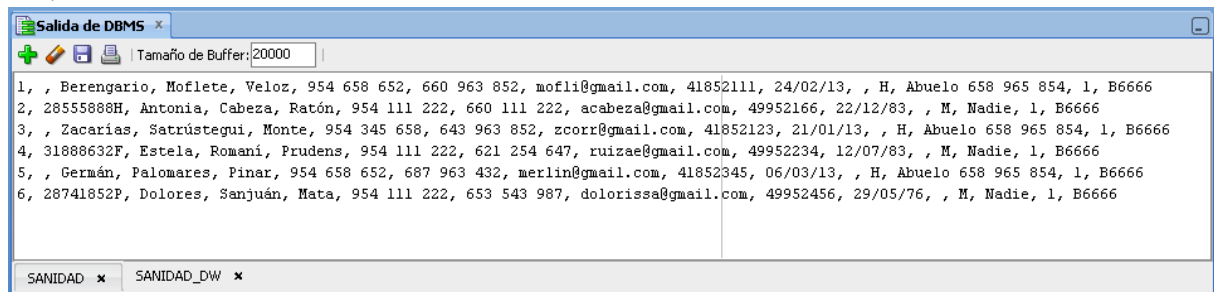


Ilustración 28: Captura de Pantalla - Ejemplo Listado 1

P\_CONS\_LIST\_PACIENTE\_ENFERMEDA (enfermedades que ha padecido un paciente):

```
BEGIN
    P_CONS_LIST_PACIENTE_ENFERMEDA ('1');
END;
```

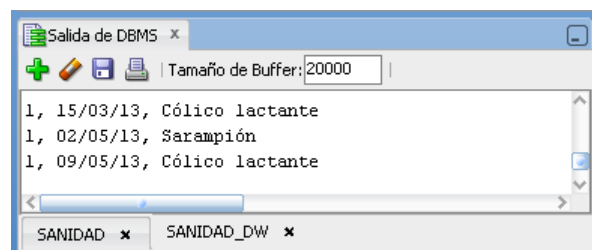


Ilustración 29: Captura de Pantalla - Ejemplo Listado 2

#### 4.1.2. Almacén de Datos

El funcionamiento normal del almacén implica lanzar el *script* del archivo de ejecución de las cargas “17\_EJECUTAR\_CARGAS\_DW.sql” cada día desde que entra en funcionamiento el sistema (base de datos y almacén). Dado que esto no ha ocurrido durante las pruebas, puesto que se han introducido datos correspondientes a fechas anteriores, se ha programado un bucle que lanza las cargas del almacén desde principios de año hasta la fecha actual.

En el archivo “20\_CARGA\_HISTÓRICA\_DW.sql” se encuentra el código que lanza los procedimientos de carga con carácter histórico, haciendo uso del parámetro de entrada que diariamente no se utiliza.

Una vez lazado este *script*, podemos consultar el archivo “21\_PRUEBAS\_DW.sql” y ver el resultado de los procedimientos de consultas estadísticas descritos anteriormente. Y aunque en este archivo están todas las pruebas, veamos aquí, como ejemplo, una captura de pantalla de la salida DBMS que los procedimientos de estadísticas ofrecen:

P\_EST\_MEDICAMENTOS\_EDAD (número de medicamentos que se consume en cada grupo de edad):

```
BEGIN  
    P_EST_MEDICAMENTOS_EDAD;  
END;
```

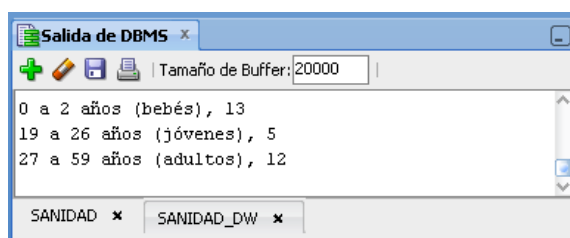


Ilustración 30: Captura de Pantalla - Ejemplo Estadística

## 4.2. Comprobación y Registro (*log*)

Como ya se ha explicado detenidamente en apartados anteriores, la forma de registrar (*log*) elegido para nuestro sistema, es mediante procesos de auditoría que registran toda la actividad tanto de la base de datos como del almacén, incluido los procesos programados de carga nocturna de este último.

Cabe mencionar que, si se desea, es labor del administrador de la base de datos complementar este nivel de control y registro con la auditoría de grano fino (FGA, del inglés *Fine Grained Auditing*) que nos ofrece el paquete DBMS. Su configuración es trivial, y podemos encontrar una buena explicación en el siguiente enlace:

[http://pic.dhe.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=%2Fcom.ibm.itcim.doc%2Ftcim85\\_install456.html](http://pic.dhe.ibm.com/infocenter/tivihelp/v2r1/index.jsp?topic=%2Fcom.ibm.itcim.doc%2Ftcim85_install456.html)

PFC-Bases de Datos – MEMORIA  
ALUMNO: DANIEL JESÚS RÖNNMARK CORDERO

Mostremos ahora unas capturas de pantalla de nuestras tablas de auditoría para comprobar que todo ha quedado correctamente registrado durante la fase de pruebas, incluso los distintos errores:

Auditoría de la Base de Datos

SELECT \* FROM SANIDAD.AUDITORIA ORDER BY FECHA\_HORA DESC;

FECHA_HORA	PROCEDIMIENTO	ENTRADA	SALIDA
227 18/05/13 11:41:35...	P_MODIFICAR_REGISTRO	TABLA: MUNICIPIO, CAMPO: ID_MUNICIPIO, VALOR: 5, PK1: ID_MUN...	Operación CORRECTA
228 18/05/13 11:41:08...	P_MODIFICAR_REGISTRO	TABLA: MUNICIPIO, CAMPO: ID_MUNICIPIO, VALOR: 6, PK1: ID_MUN...	Operación CORRECTA
229 18/05/13 11:28:30...	P_INSERTAR_MUNICIPIO	ID_PROVINCIA: 1, NOMBRE : Dos Hermanas, CODIGO_POSTAL: 41089	Operación CORRECTA
230 18/05/13 11:27:49...	P_INSERTAR_MUNICIPIO	ID_PROVINCIA: 1, NOMBRE : Lepe, CODIGO_POSTAL: 29108	Operación CORRECTA
231 18/05/13 11:27:49...	P_INSERTAR_MUNICIPIO	ID_PROVINCIA: 1, NOMBRE : Sevilla, CODIGO_POSTAL: 41001	Operación CORRECTA
232 18/05/13 11:27:49...	P_INSERTAR_MUNICIPIO	ID_PROVINCIA: 1, NOMBRE : Dos Hermanas, CODIGO_POSTAL: 410089	Error registrado: ORA-12899: el valor es demasiado grande para la col...
233 18/05/13 11:27:49...	P_INSERTAR_MUNICIPIO	ID_PROVINCIA: 1, NOMBRE : Bollullos, CODIGO_POSTAL: 41110	Operación CORRECTA
234 18/05/13 11:27:49...	P_INSERTAR_MUNICIPIO	ID_PROVINCIA: 1, NOMBRE : Bolonia, CODIGO_POSTAL: 11056	Operación CORRECTA
235 18/05/13 11:19:37...	P_INSERTAR_PROVINCIA	ID_PAIS: 1, NOMBRE : Huelva	Operación CORRECTA
236 18/05/13 11:19:37...	P_INSERTAR_PROVINCIA	ID_PAIS: 1, NOMBRE : Cádiz	Operación CORRECTA
237 18/05/13 11:18:09...	P_INSERTAR_PROVINCIA	ID_PAIS: 1, NOMBRE : Sevilla	Operación CORRECTA
238 18/05/13 11:16:49...	P_MODIFICAR_REGISTRO	TABLA: PAIS, CAMPO: ID_PAIS, VALOR: 2, PK1: ID_PAIS, ID1: 5	Operación CORRECTA
239 18/05/13 11:16:22...	P_MODIFICAR_REGISTRO	TABLA: PAIS, CAMPO: ID_PAIS, VALOR: 1, PK1: ID_PAIS, ID1: 4	Operación CORRECTA
240 18/05/13 11:15:43...	P_INSERTAR_PAIS	NOMBRE: Finlandia	Operación CORRECTA
241 18/05/13 11:15:43...	P_INSERTAR_PAIS	NOMBRE: España	Operación CORRECTA
242 18/05/13 11:10:21...	P_INSERTAR_PAIS	NOMBRE: España	Operación CORRECTA
243 18/05/13 11:10:21...	P_INSERTAR_PAIS	NOMBRE: Finlandia	Operación CORRECTA
244 18/05/13 11:07:59...	P_INSERTAR_PAIS	NOMBRE:	Error registrado: ORA-01400: no se puede realizar una inserción NULL ...
245 18/05/13 11:03:40...	P_BORRAR_REGISTRO	TABLA: CENTRO, PK1: CIF, ID1: F8798, PK2: , ID2: , PK3: , ID3:	Operación CORRECTA
246 18/05/13 11:03:17...	P_BORRAR_REGISTRO	TABLA: PAIS, PK1: ID_PAIS, ID1: 43, PK2: , ID2: , PK3: , ID3:	Operación CORRECTA
247 18/05/13 10:59:13...	P_CONS_CENTRO_CIF	CIF: B6666	Operación CORRECTA
248 18/05/13 10:59:08...	P_MODIFICAR_REGISTRO	TABLA: CENTRO, CAMPO: TELEFONO2, VALOR: 954 693 693, PK1: CI...	Operación CORRECTA
249 18/05/13 10:56:13...	P_CONS_CENTRO_CIF	CIF: B6666	Operación CORRECTA
250 18/05/13 10:55:56...	P_CONS_CENTRO_CIF	CIF: B6666	Operación CORRECTA

Ilustración 31: Captura de Pantalla - Auditoría (BD)

Auditoría del Almacén de Datos

SELECT \* FROM SANIDAD\_DW.AUDITORIA\_DW ORDER BY FECHA\_HORA DESC;

FECHA_HORA	PROCEDIMIENTO	ENTRADA	SALIDA
1 18/05/13 17:56:33...	P_EST_TIEMPO_MEDIO_BAJA	ESTADÍSTICA: MEDICAMENTOS POR EDAD	Operación CORRECTA
2 18/05/13 17:53:00...	P_EST_MEDICAMENTOS_EDAD	ESTADÍSTICA: MEDICAMENTOS POR EDAD	Operación CORRECTA
3 18/05/13 17:52:50...	P_EST_URGENCIAS_EPOCA	ESTADÍSTICA: URGENCIAS POR ÉPOCA	Operación CORRECTA
4 18/05/13 17:52:36...	P_EST_MEDICAMENTOS_EDAD	ESTADÍSTICA: MEDICAMENTOS POR EDAD	Operación CORRECTA
5 18/05/13 17:52:31...	P_CARGAR_D_ZONA	FECHA_CARGA: 02/04/13, CAMPOS DE VARIAS ZONAS CARGADAS: ID_MUNICIPIO, MUNICIPIO, PROVINCIA, PAIS	Operación CORRECTA
6 18/05/13 17:52:31...	P_CARGAR_D_EDAD	FECHA_CARGA: 02/04/13, CAMPOS DE VARIAS EDADES CARGADAS: ANOS, GRUPO	Operación CORRECTA
7 18/05/13 17:52:31...	P_CARGAR_H_VISITAS	FECHA_CARGA: 02/04/13, CAMPOS DE VARIAS VISITAS CARGADAS: DIA, ID_MUNICIPIO, ANOS, TOTAL_VISITAS, NUM_ORDINA...	Operación CORRECTA
8 18/05/13 17:52:31...	P_CARGAR_H_MEDICAMENTOS	FECHA_CARGA: 02/04/13, CAMPOS DE VARIOS MEDICAMENTOS PRESCRITOS CARGADOS: DIA, ID_MUNICIPIO, ANOS, NUM_MEDIC...	Operación CORRECTA
9 18/05/13 17:52:31...	P_CARGAR_H_BAJAS	FECHA_CARGA: 02/04/13, CAMPOS DE VARIAS BAJAS CARGADAS: DIA, ID_MUNICIPIO, ANOS, NUM_DIAS_TOTALES, NUM_DIAS_...	Operación CORRECTA
10 18/05/13 17:52:31...	P_CARGAR_D_TIEMPO	FECHA_CARGA: 03/04/13, DIA : 03/04/13, SEMANA: 14, MES: 4, TRIMESTRE: 2, AÑO: 2013	Operación CORRECTA
11 18/05/13 17:52:31...	P_CARGAR_D_ZONA	FECHA_CARGA: 03/04/13, CAMPOS DE VARIAS ZONAS CARGADAS: ID_MUNICIPIO, MUNICIPIO, PROVINCIA, PAIS	Operación CORRECTA
12 18/05/13 17:52:31...	P_CARGAR_D_EDAD	FECHA_CARGA: 03/04/13, CAMPOS DE VARIAS EDADES CARGADAS: ANOS, GRUPO	Operación CORRECTA
13 18/05/13 17:52:31...	P_CARGAR_H_VISITAS	FECHA_CARGA: 03/04/13, CAMPOS DE VARIAS VISITAS CARGADAS: DIA, ID_MUNICIPIO, ANOS, TOTAL_VISITAS, NUM_ORDINA...	Operación CORRECTA
14 18/05/13 17:52:31...	P_CARGAR_H_MEDICAMENTOS	FECHA_CARGA: 03/04/13, CAMPOS DE VARIOS MEDICAMENTOS PRESCRITOS CARGADOS: DIA, ID_MUNICIPIO, ANOS, NUM_MEDIC...	Operación CORRECTA
15 18/05/13 17:52:31...	P_CARGAR_H_BAJAS	FECHA_CARGA: 03/04/13, CAMPOS DE VARIAS BAJAS CARGADAS: DIA, ID_MUNICIPIO, ANOS, NUM_DIAS_TOTALES, NUM_DIAS_...	Operación CORRECTA
16 18/05/13 17:52:31...	P_CARGAR_D_TIEMPO	FECHA_CARGA: 04/04/13, DIA : 04/04/13, SEMANA: 14, MES: 4, TRIMESTRE: 2, AÑO: 2013	Operación CORRECTA
17 18/05/13 17:52:31...	P_CARGAR_D_ZONA	FECHA_CARGA: 04/04/13, CAMPOS DE VARIAS ZONAS CARGADAS: ID_MUNICIPIO, MUNICIPIO, PROVINCIA, PAIS	Operación CORRECTA
18 18/05/13 17:52:31...	P_CARGAR_D_EDAD	FECHA_CARGA: 04/04/13, CAMPOS DE VARIAS EDADES CARGADAS: ANOS, GRUPO	Operación CORRECTA
19 18/05/13 17:52:31...	P_CARGAR_H_VISITAS	FECHA_CARGA: 04/04/13, CAMPOS DE VARIAS VISITAS CARGADAS: DIA, ID_MUNICIPIO, ANOS, TOTAL_VISITAS, NUM_ORDINA...	Operación CORRECTA
20 18/05/13 17:52:31...	P_CARGAR_H_MEDICAMENTOS	FECHA_CARGA: 04/04/13, CAMPOS DE VARIOS MEDICAMENTOS PRESCRITOS CARGADOS: DIA, ID_MUNICIPIO, ANOS, NUM_MEDIC...	Operación CORRECTA
21 18/05/13 17:52:31...	P_CARGAR_H_BAJAS	FECHA_CARGA: 04/04/13, CAMPOS DE VARIAS BAJAS CARGADAS: DIA, ID_MUNICIPIO, ANOS, NUM_DIAS_TOTALES, NUM_DIAS_...	Operación CORRECTA
22 18/05/13 17:52:31...	P_CARGAR_D_TIEMPO	FECHA_CARGA: 05/04/13, DIA : 05/04/13, SEMANA: 14, MES: 4, TRIMESTRE: 2, AÑO: 2013	Operación CORRECTA
23 18/05/13 17:52:31...	P_CARGAR_D_ZONA	FECHA_CARGA: 05/04/13, CAMPOS DE VARIAS ZONAS CARGADAS: ID_MUNICIPIO, MUNICIPIO, PROVINCIA, PAIS	Operación CORRECTA
24 18/05/13 17:52:31...	P_CARGAR_D_EDAD	FECHA_CARGA: 05/04/13, CAMPOS DE VARIAS EDADES CARGADAS: ANOS, GRUPO	Operación CORRECTA

Ilustración 32: Captura de Pantalla - Auditoría (DW)

## 5. DOCUMENTACIÓN Y MANUAL

Para el desarrollo de este proyecto se han empleado el siguiente software:

- DBDesigner Fork
- MagicDraw UML Personal Edition
- SQL Developer
- Oracle Database 10g Express Edition

De estos, *DBDesigner Fork* y *MagicDraw UML Personal Edition* han sido empleados solo con finalidades de diseño técnico.

Por otro lado, *SQL Developer* es una herramienta gráfica para el desarrollo de base de datos que puede ser sustituida por cualquier otra, como TOAD, SQL Plus o incluso el propio editor *web* de *Oracle Database 10g Express Edition*, que permite explorar objetos, ejecutar instrucciones SQL y secuencias de comandos SQL...

Por último, *Oracle Database 10g Express Edition*, es el gestor de base de datos mínimo que debe tener la máquina donde realicemos la instalación del producto (con 'mínimo' nos referimos a que también son válidas versiones superiores de Oracle Database). Por ello, a continuación se explica con más detalle en qué consiste este sistema gestor de bases de datos y se muestra la forma en que puede obtenerse, de forma totalmente gratuita.

### *Oracle Database 10g Express Edition*

¿En qué consiste Oracle Database 10g Express Edition?

Desarrollo, implementación y distribución sin cargo.

Oracle Database 10g Express Edition (Oracle Database XE) es una base de datos de entrada de *footprint* pequeño, creada sobre la base de código Oracle Database 10g Release 2 que puede desarrollarse, implementarse y distribuirse sin cargo; es fácil de descargar y fácil de administrar. Oracle Database XE es una excelente base de datos inicial para:

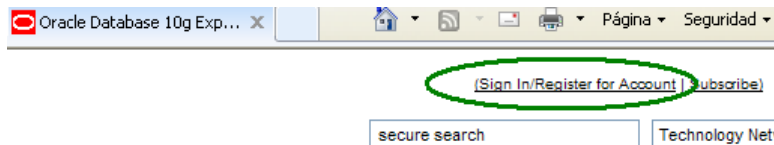
- Desarrolladores que trabajan en PHP, Java, .NET, XML, y aplicaciones de Código Abierto.
- DBAs que necesitan una base de datos inicial y sin cargo para la capacitación e implementación.
- Proveedores Independientes de Software (ISVs) y proveedores de hardware que quieren una base de datos inicial para distribuir sin cargo.
- **Instituciones educativas y estudiantes que necesitan una base de datos sin cargo para su plan de estudios.**

Fuente: [http://www.oracle.com/lang/es/database/express\\_edition.html](http://www.oracle.com/lang/es/database/express_edition.html)

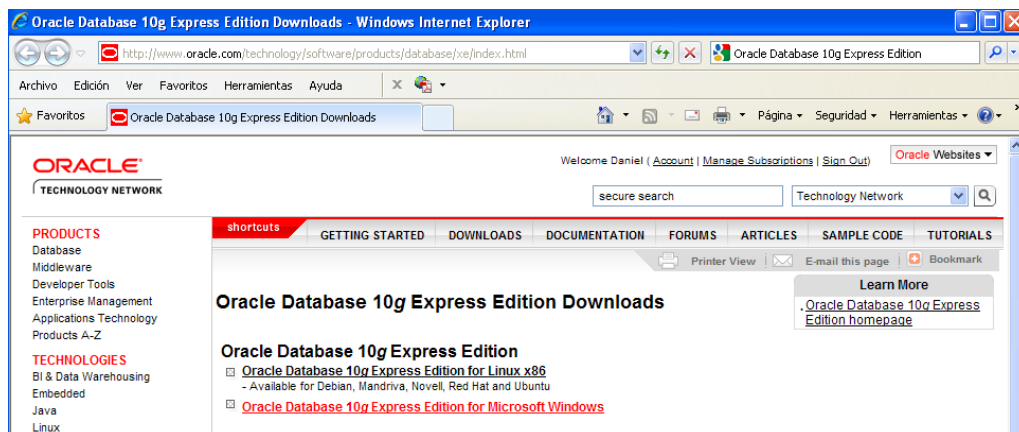
Para su descarga debemos acceder a la siguiente dirección, desde donde podremos registrarnos de forma totalmente gratuita:

<http://www.oracle.com/technology/software/products/database/xe/index.html>

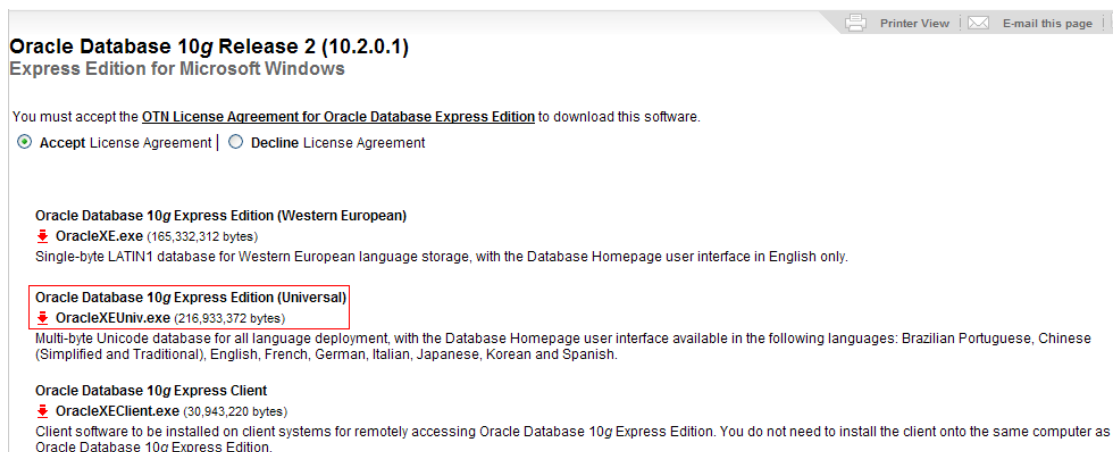
Registro:

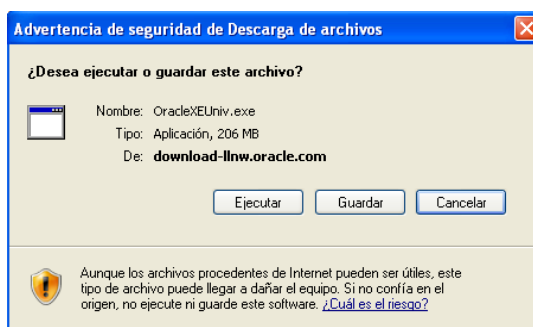


Una vez nos hayamos registrado podremos realizar la descarga:



Deberemos aceptar los términos de licencia y entonces podremos iniciar la descarga:





Instalación:

Ejecutamos el instalador descargado y procedemos a lo largo del proceso, que no permite más configuración que la contraseña de la cuenta SYS y SYSTEM. El usuario que emplearemos inicialmente es SYSTEM, con el que crearemos los usuarios SANIDAD y SANIDAD\_DW, que son los que ejecutarán todos los *scripts* de creación del producto.

Podemos encontrar información detallada para la instalación a través del siguiente enlace:

<http://www.ajpdsoft.com/modules.php?name=News&file=article&sid=231>

### *Instalación del Producto*

Para la instalación del producto, deben ejecutarse, en orden, los *scripts* que se encuentran en la carpeta Producto, teniendo en cuenta que:

- El archivo 17\_EJECUTAR\_CARGAS\_DW.sql no se ejecuta en el momento de creación, sino que es el que debe lanzarse cada noche para la carga del almacén de datos.
- Los archivos 19\_PRUEBAS\_BD.sql, 20\_CARGA\_HISTÓRICA\_DW.sql y 21\_PRUEBAS\_DW.sql solo tienen sentido si se desean cargar los datos de prueba. Es decir, para una instalación comercial, no deben ejecutarse estos bloques de código.

### *Ejecución de Procedimientos*

Como se ha explicado en los apartados de implementación, los procedimientos almacenados pueden ejecutarse desde cualquier editor SQL, con su signatura correspondiente. Veamos un par de ejemplos:

Modificación de una tabla con una clave primaria compuesta por tres campos:

```
EXECUTE P_MODIFICAR_REGISTRO ('VALORES_XLAT', 'DESCRIPCION', 'ml', 'TABLA', 'PRESENTACION', 'CAMPO', 'UNIDAD_MEDIDA', 'VALOR', '2')
```

Consulta de estadísticas a través de salida DBMS:

```
BEGIN  
P_EST_URGENCIAS_EPOCA; -- Urgencias por Época  
END;
```

## 6. EPÍLOGO

En este último apartado de la memoria veremos temas ajenos al aspecto técnico del producto elaborado, como su valoración económica, conclusiones, glosario y bibliografía.

### 6.1. Valoración Económica

Para la siguiente propuesta económica nos basaremos en los precios orientativos facilitados en la asignatura “Gestión de Organizaciones y Proyectos Informáticos” de la UOC.

Recurso	Coste/hora	Coste/jornada
Jefe de proyecto	48 €	384 €
Analista	36 €	288 €
Analista programador	24 €	192 €
Técnico de sistemas	35 €	280 €

En base a estos precios y al cálculo de horas dedicados, adoptando cada perfil profesional según la tarea realizada, y considerando una jornada laboral de dos horas, tenemos la siguiente propuesta económica.

Actividad	Perfil	Jornadas	Horas	Coste
Inicio PFC	Jefe de proyecto	13	26	1.248 €
Instalación Software	Técnico de sistemas	3	6	210 €
Base de Datos				
Análisis	Analista	9	18	648 €
Diseño	Analista	12	24	864 €
Implementación	Analista programador	7	14	336 €
Almacén de Datos				
Análisis	Analista	5	10	360 €
Diseño	Analista	9	18	648 €
Implementación	Analista programador	6	12	288 €
Pruebas y Testeo	Analista programador	7	14	336 €
Fase Final (Revisión, redacción, presentación, producto...)	Jefe de proyecto	15	30	1.440 €
<b>TOTAL</b>				<b>6.378 €</b>

Ilustración 33: Valoración Económica



## 6.2. Conclusiones

El trabajo presentado es el fruto de muchas horas de trabajo en las que se ha buscado la calidad del producto elaborado, planteando una estrategia no solo teórica sino pragmática, como si se tratara de una herramienta comercial real que debiera ser implementada, siendo conscientes, por supuesto, de que de ser así, posiblemente los requerimientos hubieran sido mayores – pensemos en todo el sistema sanitario español.

Por un lado, en la base de datos no se han tenido grandes dificultades en su desarrollo, ya que las asignaturas de la carrera preparan bastante bien para ello, y el bagaje profesional de este autor también ha facilitado mucho la tarea.

Por otro lado, el almacén de datos, del que no se poseían tan amplios conocimientos, ha exigido más tiempo de estudio e investigación, que han merecido la pena, ya que ha resultado ser un tema muy interesante y útil para las organizaciones, lo que puede establecer una ventaja profesional para el escribiente, que ya ha entrado en contacto con estos sistemas y en los que, a partir de ahora, tiene la obligación intelectual de profundizar.

Cabe mencionar que en la ejecución no han existido variaciones respecto a la planificación, y se han podido cumplir todos los plazos exhaustivamente. Tanto es así que, habiendo existido dificultades que hubieran podido retrasar el progreso del proyecto, los días de contingencia establecidos han sido capaces de paliar el problema.

En síntesis, se han conseguido todos los objetivos, puesto que el sistema desarrollado cumple con creces lo solicitado por el cliente, con una calidad profesional, un tiempo de elaboración razonable y un presupuesto bastante asequible. Asimismo, ha servido para su fin académico, que es el de reforzar los conocimientos adquiridos en la carrera y plasmarlos en un trabajo final que haga uso de las habilidades técnicas, de gestión y de redacción. Este es un reto que, al cumplirse, llena de satisfacción al alumno y, con suerte, al tribunal evaluador.

## 6.3. Glosario de Términos y Siglas

**Almacén de datos (*Data Warehouse*):** conjunto de datos organizados como hechos mensurables y consultables según dimensiones de un sistema de coordenadas. Su objetivo principal es que la consulta de información para la ayuda a la toma de decisiones sea lo más veloz y ágil posible.

**Bases de datos (BBDD):** conjuntos de datos, almacenados y organizados en forma de tuplas, que pertenecen a un mismo ámbito y que son explotados para obtener información útil.

**ETL (*'Extract, Transform and Load', 'Extraer, Transformar y Cargar'*):** proceso que permite mover datos desde distintas fuentes, transformarlos, limpiarlos, organizarlos... y cargarlos en otra base de datos, como por ejemplo un almacén de datos, para analizarlos y utilizarlos como mejor convenga en un proceso de negocio o en la toma de decisiones.

**Modelo E/R (Entidad/Relación):** modelado de datos que muestra las entidades relevantes de un sistema de información, sus propiedades y la relación entre ellas.

**NIF:** Número de Identificación Fiscal.

**OLAP (*On-Line Analytical Processing*, Procesamiento analítico en línea):** idea nacida del campo de la Inteligencia Empresarial (*Business Intelligence*) para agilizar la consulta de grandes cantidades de datos mediante estructuras multidimensionales, llamados cubos OLAP, que resumen los datos de grandes bases de datos.

**PL/SQL (*Procedural Language/Structured Query Language*):** lenguaje de programación incrustado en Oracle y que amplía, con nuevas características, al lenguaje SQL para la creación de bloques de código en procedimientos, funciones u otros *scripts*.

**Procedimiento almacenado (*Stored Procedure*):** programa almacenado físicamente en la base de datos, que es ejecutado directamente en su motor de datos, es decir, en el servidor, no en la máquina que lo lanza, de manera que solo envía al usuario los resultados, sin que haya transporte de grandes cantidades de datos.

**Script:** texto plano que puede ser leído e interpretado por un sistema. En el ámbito de las bases de datos son líneas de instrucciones SQL que pueden ser interpretados por el SGBD.

**Sistema de Gestión de Bases de Datos (SGBD):** conjunto de programas que permiten el almacenamiento, modificación, extracción y análisis de la información en una base de datos.

**SQL (*Structured Query Language*):** lenguaje estructurado de acceso a bases de datos.

**UML (*Unified Modeling Language*, Lenguaje Unificado de Modelado):** lenguaje de modelado gráfico para visualizar, especificar, construir y documentar un sistema.

## 6.4. Bibliografía

**Almacén de datos** [en línea]

[http://es.wikipedia.org/wiki/Almac%C3%A9n\\_de\\_datos](http://es.wikipedia.org/wiki/Almac%C3%A9n_de_datos)

**Cubos OLAP** [en línea]

[http://es.wikipedia.org/wiki/Cubo\\_OLAP](http://es.wikipedia.org/wiki/Cubo_OLAP)

[http://en.wikipedia.org/wiki/Comparison\\_of\\_OLAP\\_Servers](http://en.wikipedia.org/wiki/Comparison_of_OLAP_Servers)

**Cursoros** [en línea]

<http://www.techonthenet.com/oracle/questions/cursor1.php>

<http://www.techonthenet.com/oracle/questions/cursor3.php>

[http://docs.oracle.com/cd/B19306\\_01/appdev.102/b14261/cursor\\_variables.htm](http://docs.oracle.com/cd/B19306_01/appdev.102/b14261/cursor_variables.htm)

[http://docs.oracle.com/cd/B14117\\_01/appdev.101/b10807/13\\_elems033.htm](http://docs.oracle.com/cd/B14117_01/appdev.101/b10807/13_elems033.htm)

<http://mioracle.blogspot.com.es/2008/07/cursores-explicitos-en-plsql.html>  
<http://www.devjoker.com/contenidos/catss/39/Cursores-Explicitos-en-PLSQL.aspx>

**Moral Rubia, Jorge** (2009). Artículo: Introducción a los almacenes de datos [en línea]  
[https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CEQQFjAB&url=https%3A%2F%2Fwww.ica.es%2Fpublicaciones%2Fanales\\_get.php%3Fid%3D1653&ei=tz6RUby4Icjm7AaP1oDgDw&usg=AFQjCNESm6BOQLGRgoI5bWRjMjTe-tbhfQ&bvm=bv.46340616,d.ZGU&cad=rja](https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0CEQQFjAB&url=https%3A%2F%2Fwww.ica.es%2Fpublicaciones%2Fanales_get.php%3Fid%3D1653&ei=tz6RUby4Icjm7AaP1oDgDw&usg=AFQjCNESm6BOQLGRgoI5bWRjMjTe-tbhfQ&bvm=bv.46340616,d.ZGU&cad=rja)

**Oracle Functions** [en línea]  
<http://psoug.org/reference/functions.html>

**Result Sets from Stored Procedures In Oracle** [en línea]  
[http://asktom.oracle.com/pls/asktom/ASKTOM.download\\_file?p\\_file=6551171813078805685](http://asktom.oracle.com/pls/asktom/ASKTOM.download_file?p_file=6551171813078805685)

**Using Procedures and Packages** [en línea]  
[http://docs.oracle.com/cd/B10501\\_01/appdev.920/a96590/adg10pck.htm](http://docs.oracle.com/cd/B10501_01/appdev.920/a96590/adg10pck.htm)

## 6.5. Anexos

Tal y como ya se ha visto en la presente memoria, anexa a ella se encuentra la carpeta **Producto**, en donde podemos encontrar los siguientes *scripts*:

- Definición de espacios de tablas (*tablespaces*) de la base de datos
- Creación de usuarios
- Creación de la base de datos
- Procedimientos almacenados de la base de datos
- Definición de espacios de tablas (*tablespaces*) del almacén de datos
- Creación del almacén de datos
- Procedimientos almacenados del almacén de datos
- Cargas del almacén
- Procedimientos relaciones con auditorías y registro (*logs*)
- Juego de pruebas de la base de datos
- Juego de pruebas de almacén de datos

---

Por favor, contacten conmigo si tienen cualquier duda:

[dronmark@uoc.edu](mailto:dronmark@uoc.edu)

**Daniel Jesús Rönmark Cordero (Alumno UOC - Ingeniería Informática)**

