



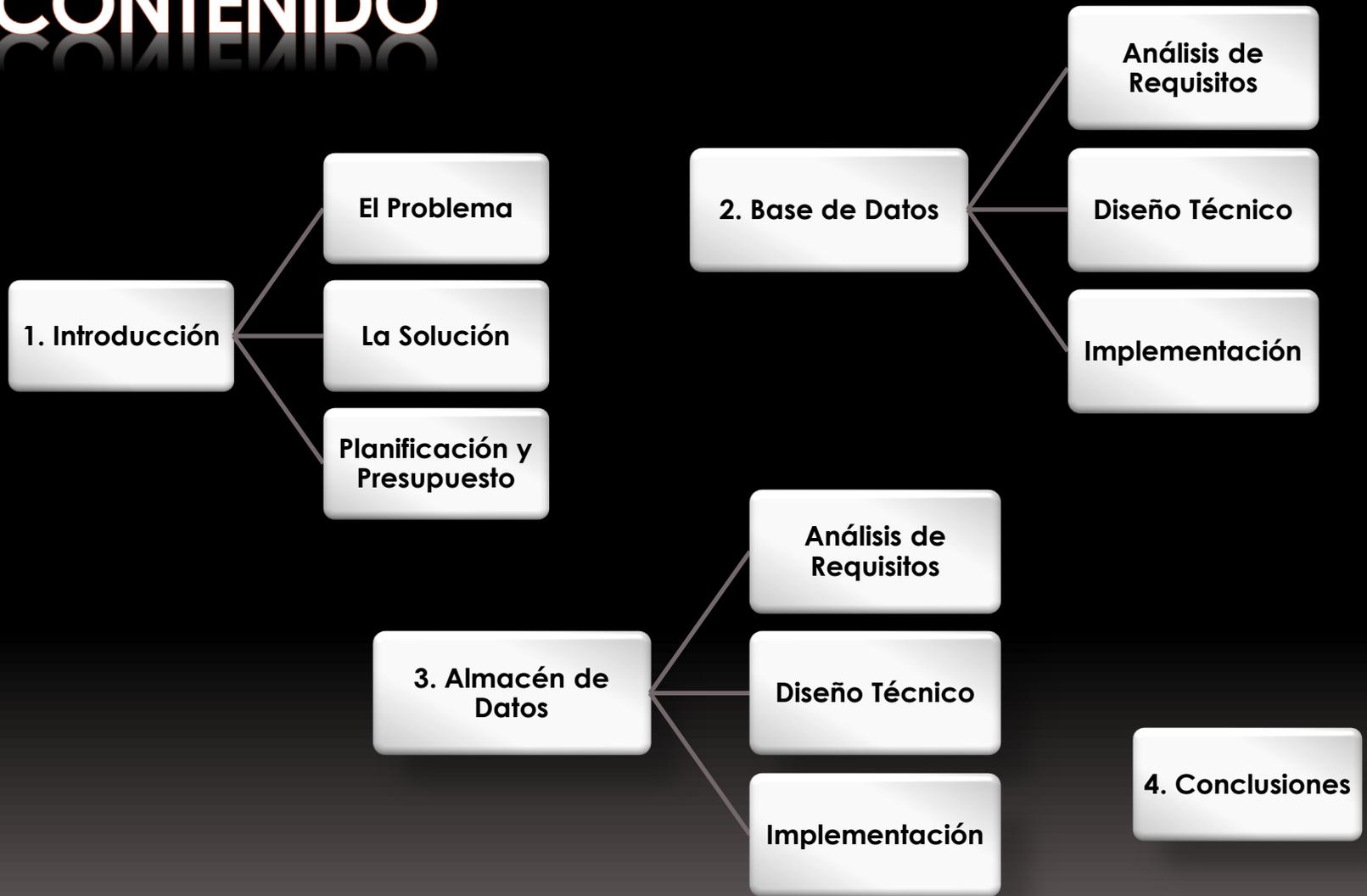
**DISEÑO E IMPLEMENTACIÓN DE UNA BASE DE DATOS  
RELACIONAL PARA LA GESTIÓN SANITARIA**  
PROYECTO FIN DE CARRERA – INGENIERÍA INFORMÁTICA

**2012-2013/2**

ALUMNO: DANIEL JESÚS RÖNNMARK CORDERO  
CONSULTOR: JUAN MARTÍNEZ BOLAÑOS



# CONTENIDO



# 1. INTRODUCCIÓN

# EL PROBLEMA

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** que dé solución al almacenamiento de información referente a médicos, pacientes, centros, medicamentos, etc.
- Un **almacén de datos** que permita extraer estadísticas y explotar grandes cantidades de datos de forma ágil de la forma más versátil posible.

**Procedimientos almacenados:** 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, PL/SQL.

**Auditorías:** 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.

# LA SOLUCIÓN

Para encontrar una solución con calidad profesional, debemos ejecutar con precisión las siguientes tareas:

- Planificar el trabajo a realizar identificando y gestionando posibles riesgos.
- Analizar los requisitos impuestos por el cliente y definir, con claridad, el alcance.

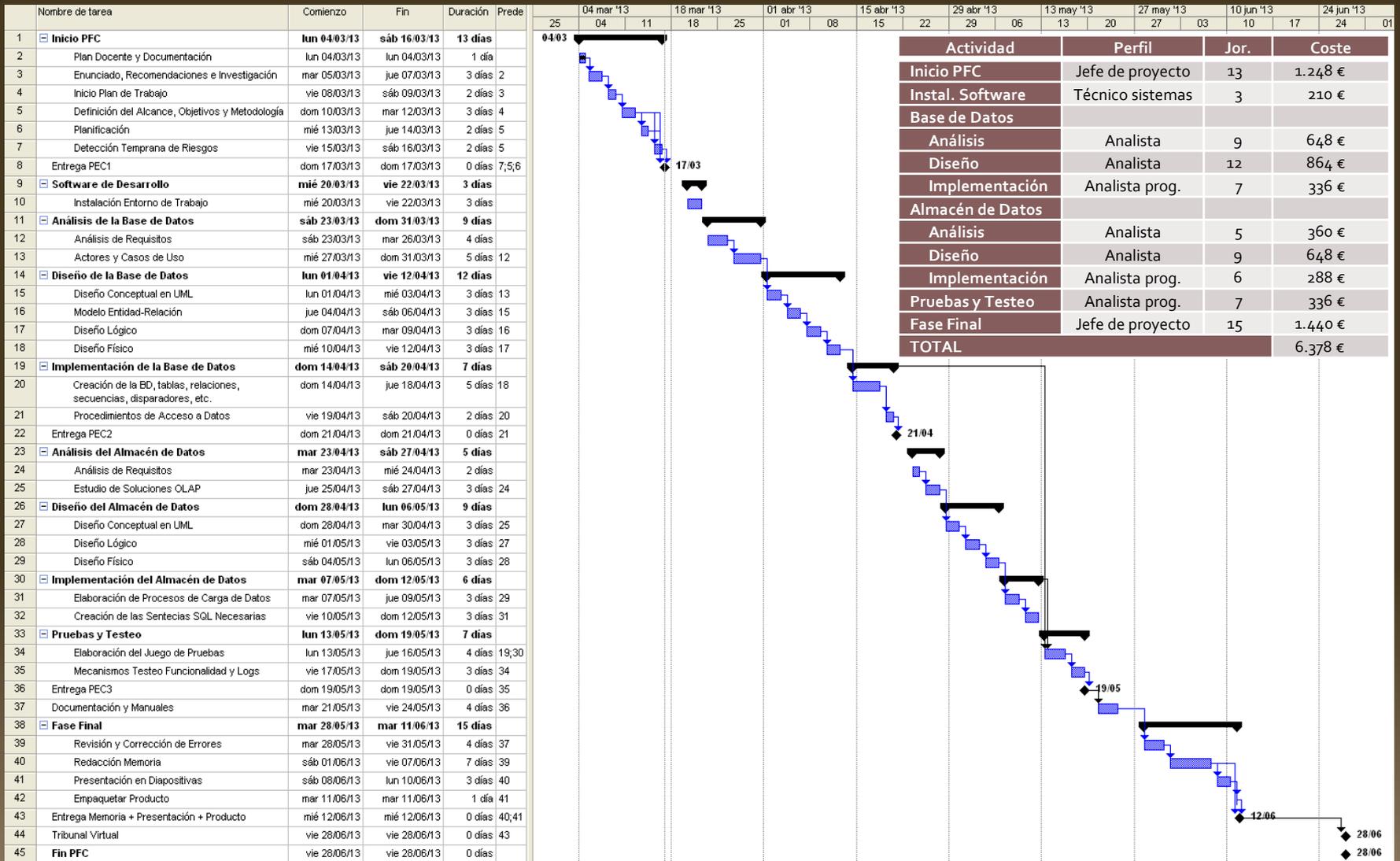
- Base de Datos
- Diseñar la base de datos a través de su diseño conceptual UML, identificando entidades, y poder obtener el modelo E/R, el diseño lógico y diseño físico.
  - Implementar la base de datos mediante el SGBD de Oracle: tablas, relaciones, secuencias, disparadores... así como los procedimientos de acceso a datos.

- Almacén de Datos
- Diseñar el almacén de datos, definiendo los hechos mensurables y dimensiones requeridos.
  - Implementar el almacén de datos: hechos, dimensiones, relaciones entre ellos, así como los procesos automáticos de carga del almacén de datos.

- Desarrollar mecanismos de testeo sobre los problemas de integración con el resto del sistema (logs, auditorías, etc.).
- Probar el trabajo realizado mediante un juego de pruebas sobre la base y almacén de datos.

# PLANIFICACIÓN Y PRESUPUESTO

Cada día planificado supone dos horas de trabajo

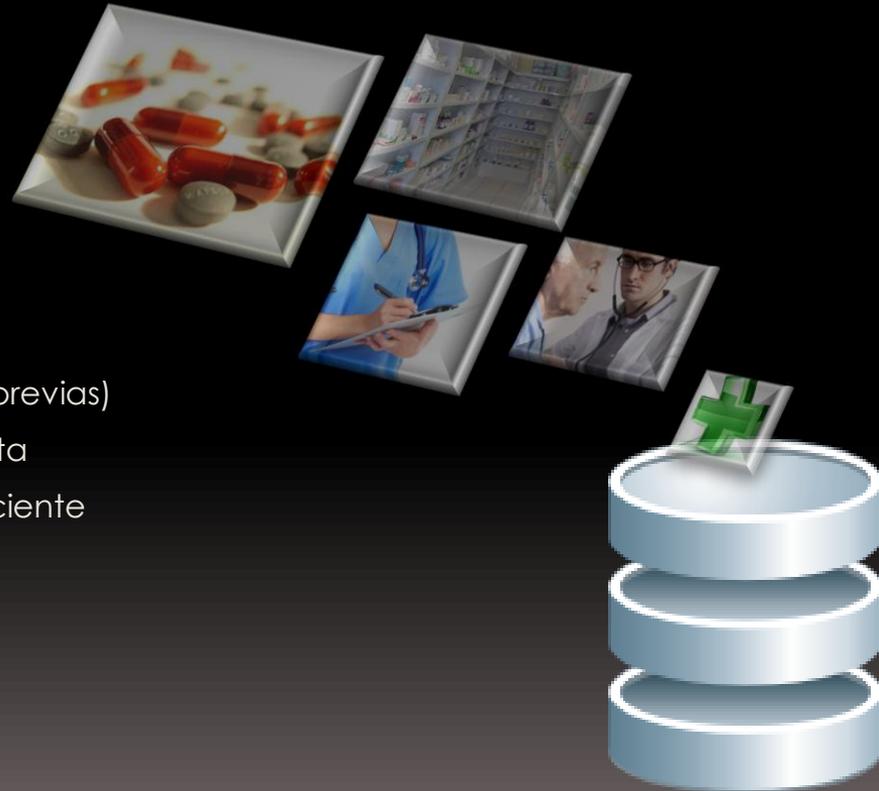


# 2. BASE DE DATOS

# ANÁLISIS DE REQUISITOS

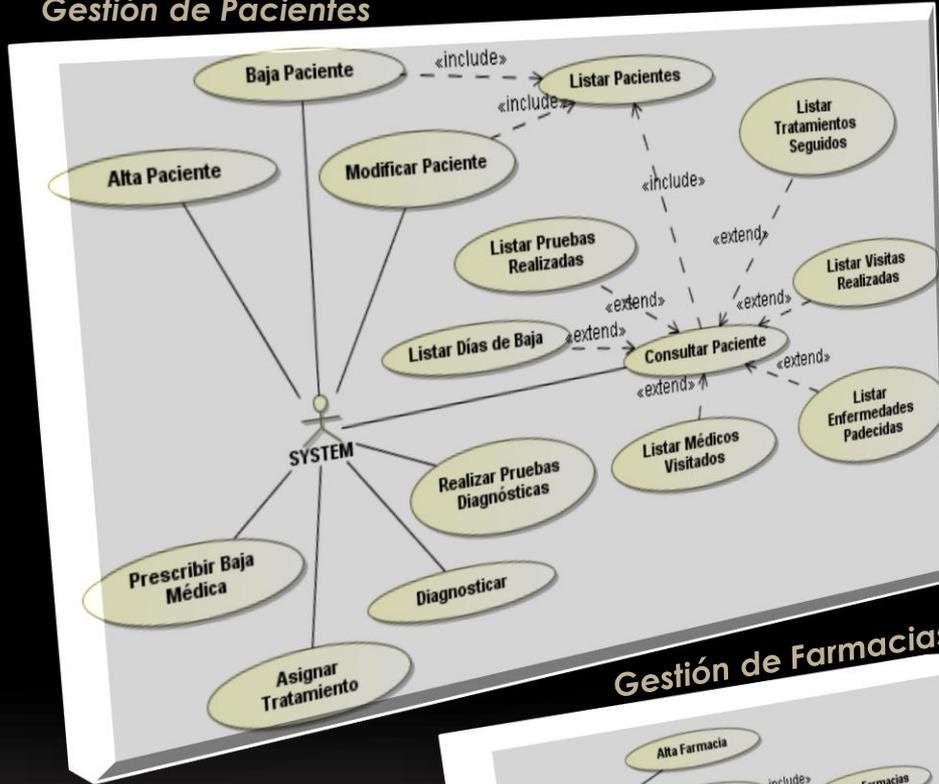
Los requisitos funcionales de la base de datos no incluyen descripciones sobre interfaces de usuario, sino que los actores que interactuarán con nuestro sistema serán *scripts* y otras aplicaciones que accederán a los datos mediante procedimientos almacenados que deben permitir realizar el alta, baja, 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)

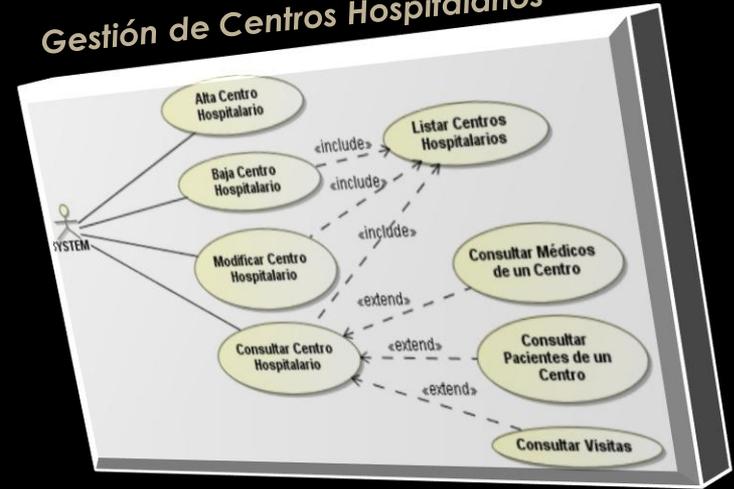


# ANÁLISIS DE REQUISITOS. CASOS DE USO

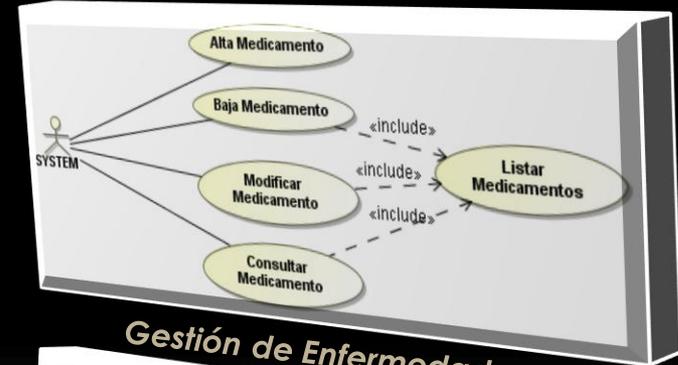
## Gestión de Pacientes



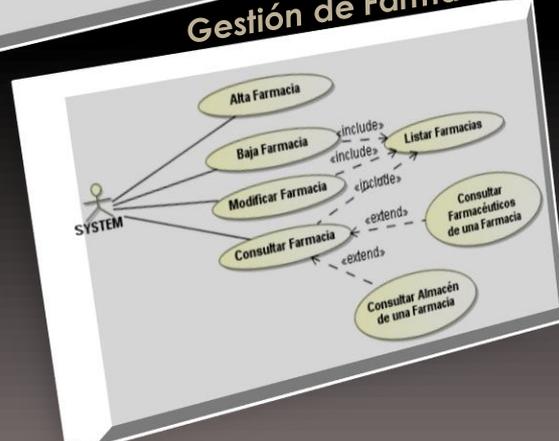
## Gestión de Centros Hospitalarios



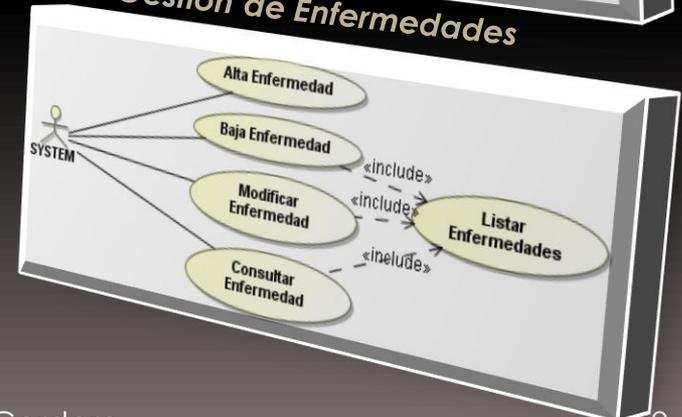
## Gestión de Medicamentos



## Gestión de Farmacias



## Gestión de Enfermedades



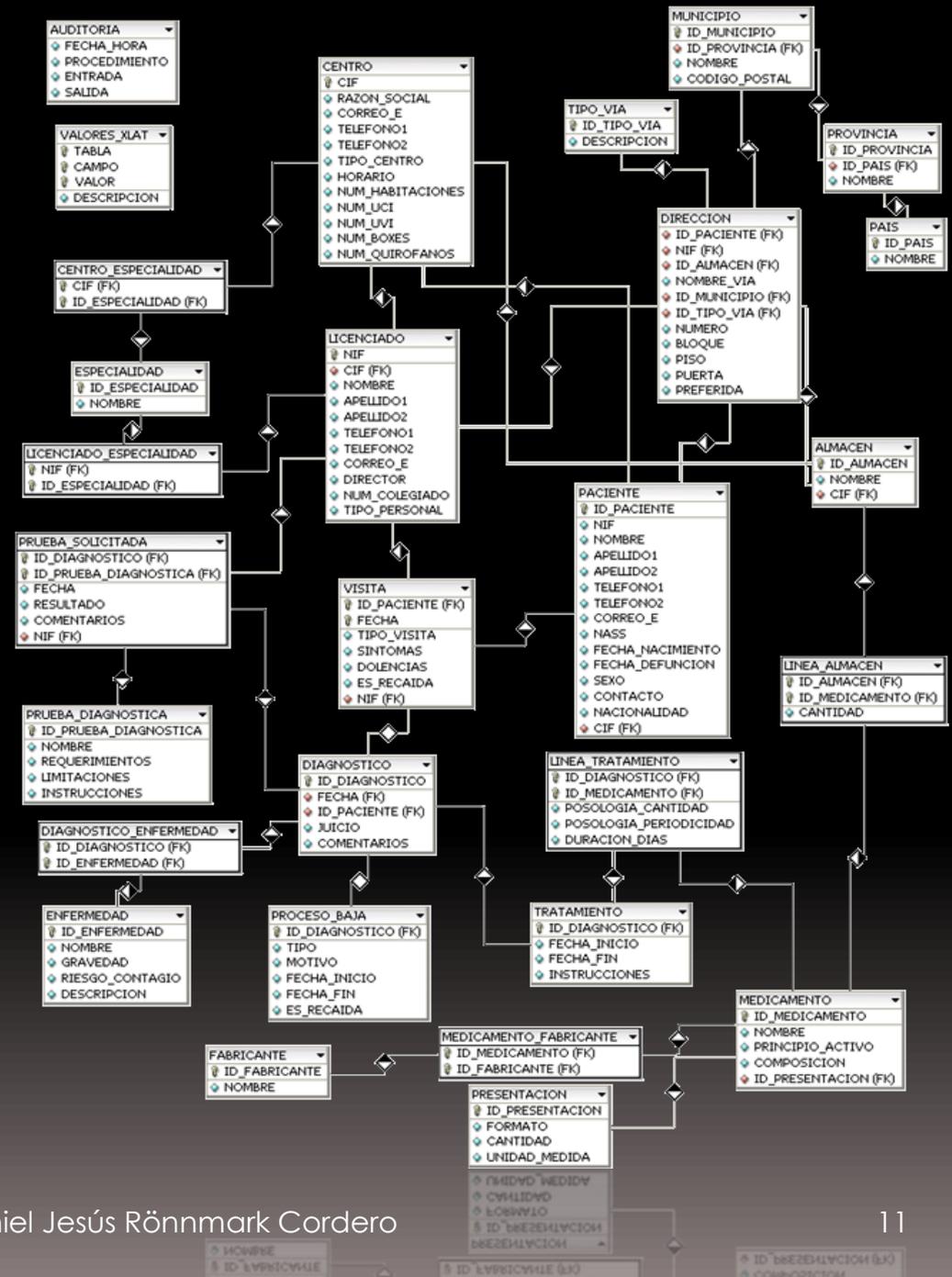


# DISEÑO TÉCNICO. E/R

Pasamos el diagrama UML a un modelo entidad-relación (E/R), más cercano al diseño físico que debemos definir en Oracle Database.

## Decisiones de diseño:

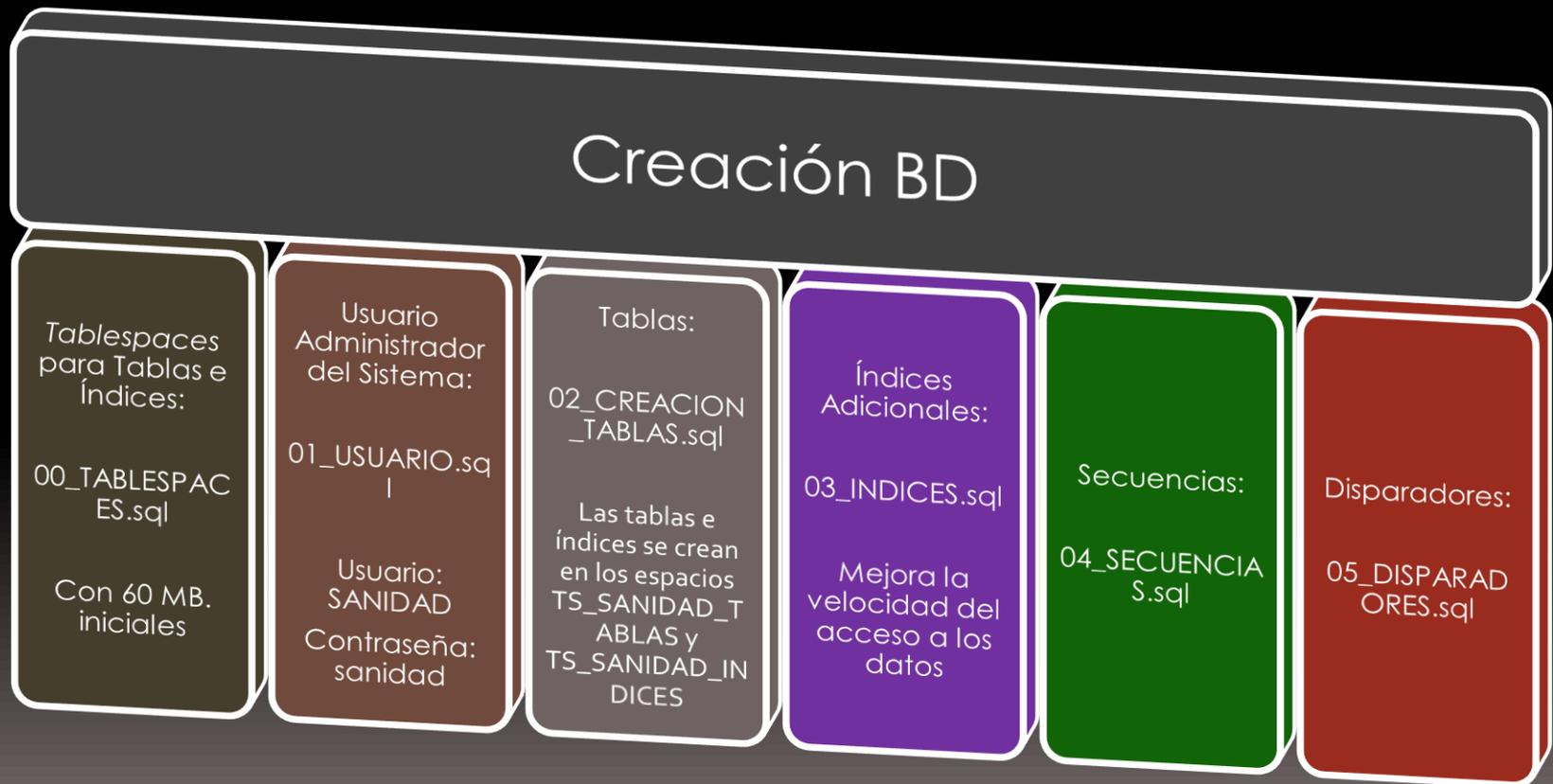
- Las sub-entidades 'CentroHospitalario' y 'Farmacia' pasarán sus atributos a la clase abstracta 'Centro'. Para diferenciarlos existe el campo 'tipoCentro'.
- Las sub-entidades de 'Persona', 'Médico' y 'Farmacéutico' pasan a generalizarse en la entidad 'Licenciado'. Asimismo, 'Paciente' queda desvinculada y adquiere su propia entidad.
- 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', etc.).
- Se crea una tabla AUDITORIA que registra todas las acciones realizadas en la base de datos.



# IMPLEMENTACIÓN. CREACIÓN BD

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...

Todos los *scripts* se encuentran en la carpeta 'Producto' adjunta a esta presentación.



# IMPLEMENTACIÓN. AUDITORÍA

Creamos 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.

Su signatura es la siguiente:

Procedimiento	Entrada	Salida
P_INSERTAR_AUDITORIA	PROCEDIMIENTO	Si correcto:
	ENTRADA	✓ COMMIT
	SALIDA	Si incorrecto por valor nulo: ✓ Lanza mensaje indicando que el campo no puede ser nulo
	<i>(El campo FECHA_HORA se rellena con la fecha y hora del sistema)</i>	Si incorrecto por otro motivo: ✓ Registro del error en tabla auditoría

Así, en todos los procedimientos podemos encontrar un código como este tras intentar su acción:

```
/* Operación CORRECTA */
LOG_SALIDA:= 'Operación CORRECTA';
P_INSERTAR_AUDITORIA('P_INSERTAR_CENTRO', LOG_ENTRADA, LOG_SALIDA);
COMMIT;
/* Operación INCORRECTA */
EXCEPTION
WHEN OTHERS THEN
ROLLBACK;
LOG_SALIDA := 'Error registrado: ' || SQLERRM;
P_INSERTAR_AUDITORIA('P_INSERTAR_CENTRO', LOG_ENTRADA, LOG_SALIDA);
RAISE_APPLICATION_ERROR (-20001, SQLERRM);
```

Script:  
06\_PROCEDIMIENTO\_AUDITORIA.sql

# IMPLEMENTACIÓN. PROCEDIMIENTOS ALMACENADOS

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 esto debemos emplear técnicas Oracle PL/SQL.

Dividiremos estos procedimientos en cuatro grupos:

## Procedimientos de Inserción:

- Uno para cada tabla de la base de datos.
- Los parámetros de entrada son todos los campos que tiene la tabla destino, excepto los auto incrementales.

*Script: 07\_PROCEDIMIENTOS\_INSERTION.sql*

## Procedimiento de Borrado:

- 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 (aunque esté compuesta por varios campos).

*Script: 08\_PROCEDIMIENTO\_BORRADO.sql*

## Procedimiento de Modificación:

- 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 (aunque esté compuesta por varios campos).

*Script: 09\_PROCEDIMIENTO\_MODIFICACION.sql*

## Procedimientos de Consulta:

- La aplicación externa solicitante declara una variable del tipo de fila (ROWTYPE) de la tabla que va a consultar y la envía a la función junto con el valor de la clave del registro que va a consultar. Esto nos devuelve el registro coincidente.

*Script: 10\_PROCEDIMIENTOS\_CONSULTAS.sql*

# 3. ALMACÉN DE DATOS

# 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.

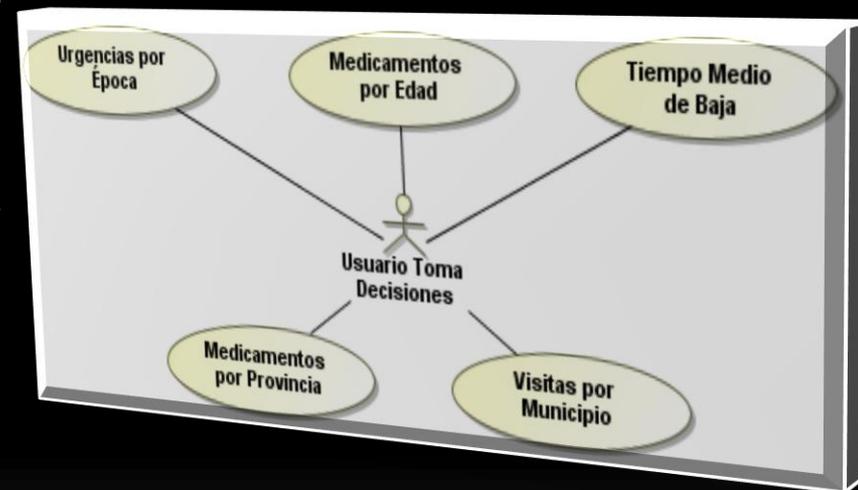
El cliente ha referido que necesita estadísticas que le permitan saber:

- 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.

## Casos de Uso



### ETL:

- *Los procesos de extracción, transformación y carga 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.*

# 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.).

## Dimensiones

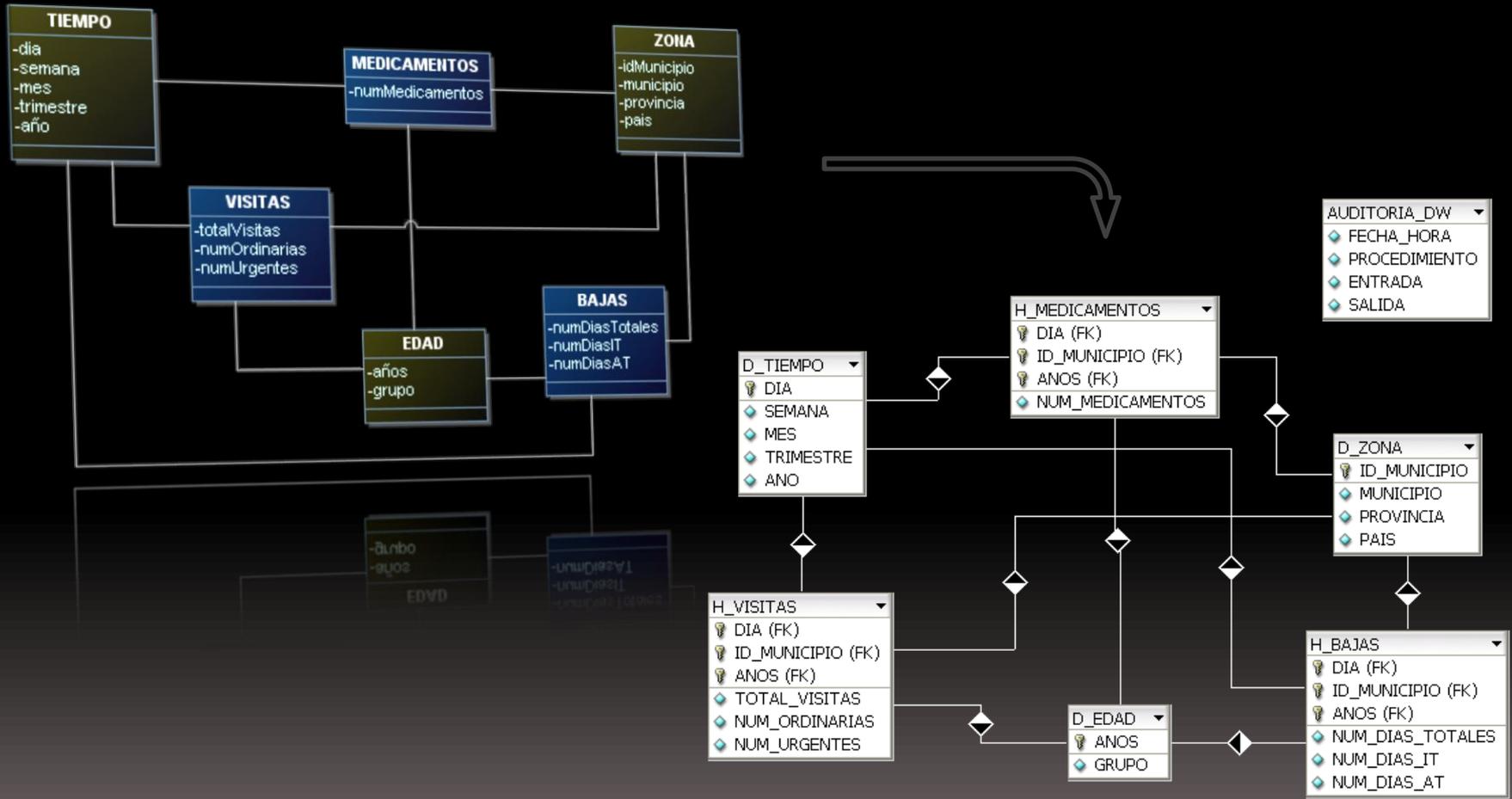
Dimensión	Jerarquía	Rango Valores
TIEMPO	Año	2013 en adelante
	Trimestre	1 - 4
	Mes	1 - 12
	Semana	1 - 52
	Día	Cualquier fecha
ZONA	País	Tabla PAIS
	Provincia	Tabla PROVINCIA
	Municipio	Tabla MUNICIPIO
EDAD	Grupo	0 a 2 años (bebés)
		3 a 12 años (niños)
		13 a 18 años (adolescentes)
		19 a 26 años (jóvenes)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

## Hechos Mensurables

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

# DISEÑO TÉCNICO. UML Y E/R

Es fácil diseñar el diagrama UML de este sistema, y a partir de este, encontrar el modelo E/R.



# IMPLEMENTACIÓN. CREACIÓN ALMACÉN DE DATOS

## Creación Almacén de Datos

Tablespaces para  
Tablas e Índices:

11\_TABLESPACES\_DW.sql

Con 100 MB. iniciales

Usuario Administrador  
del Almacén de Datos:

12\_USUARIO\_DW.sql

Usuario: SANIDAD\_DW

Contraseña:  
sanidad\_dw

Tablas:

13\_CREACION\_TABLAS\_  
DW.sql

Las tablas e índices se crean  
en los espacios  
TS\_SANIDAD\_TABLAS\_DW  
y  
TS\_SANIDAD\_INDICES\_DW

Índices Adicionales:

14\_INDICES\_DW.sql

Mejora la velocidad del  
acceso a los datos

Auditoría:

15\_PROCEDIMIENTO\_AU  
DITORIA\_DW.sql

### Procedimientos de Carga de Datos:

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

Script: 16\_PROCEDIMIENTOS\_CARGA\_DW.sql

PROCEDIMIENTO	ACCIÓN
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.
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.
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.
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.
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.
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.

# 4. CONCLUSIONES

Finalmente, habiendo llevado a cabo un profundo estudio e investigación sobre las herramientas seleccionadas, 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.**

Por favor, contacten conmigo si tienen cualquier duda: [dronnmark@uoc.edu](mailto:dronnmark@uoc.edu)

**GRACIAS POR SU ATENCIÓN**

**Reciban un cordial saludo,**

**Daniel Jesús Rönmark Cordero (Alumno Ingeniería Informática\_UOC).**