

Diseño y Explotación de un almacén de UCI Sanitaria



Alejandro Silva Rodríguez

Marta Cuevas Rodríguez

Almacenes De Datos
Universidad de Málaga

Septiembre 2024

Índice

1. Introducción	2
2. Objetivos	2
3. Diseño Conceptual	2
3.1. Descripción del Modelo Conceptual	2
4. Diseño Lógico	4
4.1. Transformación a Modelo en Copo de Nieve	4
4.2. Definición de Claves	5
4.3. Instrucciones para Crear el Almacén de Datos	5
5. Dificultades Encontradas	5
6. Conclusión	6
7. Acceso al Repositorio	7

1. Introducción

En el contexto hospitalario actual, el monitoreo y la gestión eficiente de los recursos es una necesidad apremiante, especialmente en unidades como la de Cuidados Intensivos (UCI), donde la administración de medicamentos representa una parte sustancial de los costos. A nivel mundial, el incremento en el costo de los medicamentos y la presión financiera sobre los sistemas de salud han impulsado la búsqueda de soluciones que optimicen el uso de los recursos en entornos críticos. Sin embargo, muchas instituciones hospitalarias carecen de herramientas analíticas específicas para monitorizar y analizar de manera detallada el gasto en fármacos, lo que limita la capacidad de identificar patrones de consumo y optimizar la asignación de presupuestos.

En Estados Unidos, la situación es especialmente significativa debido a la gran cantidad de recursos destinados a la atención en las UCI y la disponibilidad de grandes volúmenes de datos clínicos. Actualmente, existen bases de datos como la proporcionada por el Instituto de Tecnología de Massachusetts (MIT) [3], que recogen información completa sobre los tratamientos, síntomas y diagnósticos de los pacientes, ofreciendo una oportunidad para profundizar en el análisis del gasto en medicamentos en pacientes críticos. No obstante, estas bases de datos no están estructuradas para un análisis enfocado exclusivamente en los costos de los medicamentos en el entorno de la UCI, lo que representa un vacío que este proyecto busca abordar.

2. Objetivos

Este proyecto tiene como objetivo desarrollar un almacén de datos específicamente diseñado para analizar el gasto en medicamentos en pacientes ingresados en la UCI en hospitales de EE.UU. A partir de la base de datos proporcionada por el MIT, se creará un modelo que permita identificar patrones de consumo y gasto en fármacos administrados a pacientes críticos. Con esto, se espera ofrecer una herramienta que aporte información valiosa para la gestión de recursos en las UCI, contribuya a optimizar los tratamientos y ayude en la planificación presupuestaria de los hospitales.

3. Diseño Conceptual

3.1. Descripción del Modelo Conceptual

El modelo conceptual (Figura 1) de nuestro almacén de datos toma como hecho principal el **Gasto en Medicamentos** en la UCI del hospital, y como medidas las variables de **cantidad de gasto**, **tasa de infusión** y **volumen de fluido administrado**.

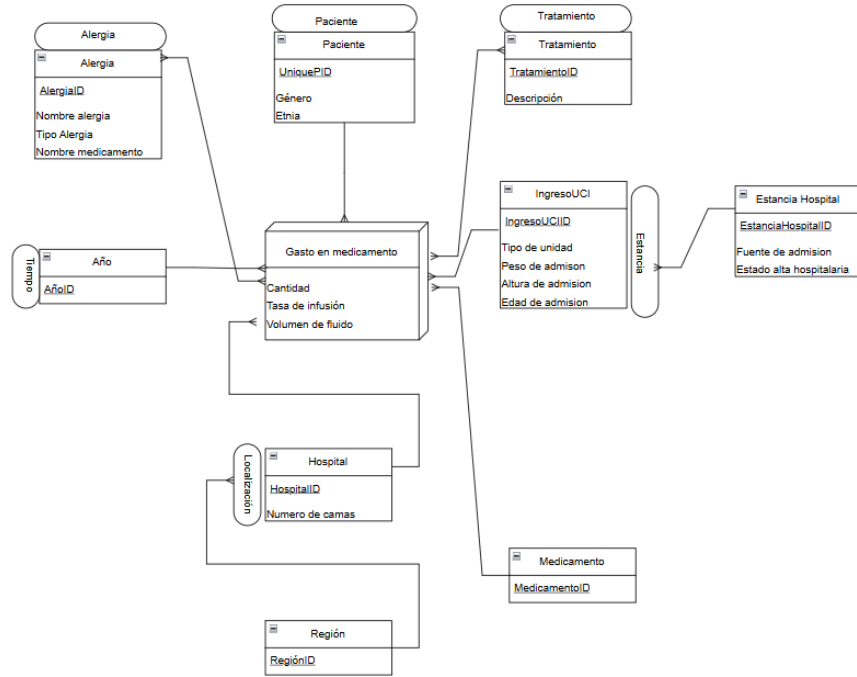


Figura 1: Modelo Conceptual elaborado con *draw.io* [4]

El modelo está compuesto por cinco dimensiones, descritas a continuación:

1. **Tiempo:** Esta dimensión organiza los datos temporalmente mediante el atributo *Año*.
2. **Hospital:** La dimensión *Hospital* incluye los niveles *Región* y *Hospital*, lo que permite agrupar los datos por área geográfica y hospital específico. Además, el hospital contiene como atributo una variable categórica que indica el número de camas (por ejemplo, <500 camas).
3. **Estancia:** La dimensión *Estancia* describe el tiempo de permanencia del paciente en el hospital. Está estructurada en los niveles *Ingreso en UCI* y *Estancia en el Hospital*, ya que un mismo paciente puede ingresar varias veces en la UCI durante una sola estancia en el hospital. Para el nivel *Ingreso en UCI*, los atributos incluyen *Tipo de Unidad*, *Peso*, *Altura*, *Edad de Admisión* y *Género del Paciente*. En cuanto a la *Estancia en el Hospital*, los atributos incluyen la *Fuente de Admisión* (departamento donde fue admitido) y el *Estado de Alta Hospitalaria* (indicando si el paciente fue dado de alta vivo o fallecido).
4. **Medicamento:** Esta dimensión permite categorizar los datos según el *Medicamento* administrado al paciente en la UCI.
5. **Tratamiento:** Esta dimensión clasifica el gasto en medicamentos según el tipo de *Tratamiento* para el que se administran, permitiendo analizar el consumo de fármacos en función de las diferentes intervenciones clínicas en la UCI.
6. **Alergia:** Esta dimensión agrupa los datos según los tipos de *Alergias* presentes en los pacientes, indicando la relación entre las alergias y los medicamentos administrados, lo cual permite analizar cómo estas condiciones impactan en el gasto en fármacos.
7. **Paciente:** Esta dimensión identifica al *Paciente* en quien se produce el gasto de medicamentos, permitiendo un análisis individualizado del consumo de fármacos en función de las características demográficas y clínicas de cada paciente.

4. Diseño Lógico

4.1. Transformación a Modelo en Copo de Nieve

s

Para construir el modelo lógico 2 a partir del modelo conceptual, realizamos una transformación en la que cada dimensión y el hecho principal, **Gasto en Medicamentos**, fueron convertidos en entidades relacionales en forma de tablas. Este modelo lógico, desarrollado con *Oracle SQL Developer Data Modeler* [2], sigue una estructura en estrella, donde la dimensión de tiempo se encuentra denormalizada para optimizar el análisis temporal. La estructura planteada permite almacenar el gasto en medicamentos junto a los atributos de las diferentes dimensiones relevantes para el análisis.

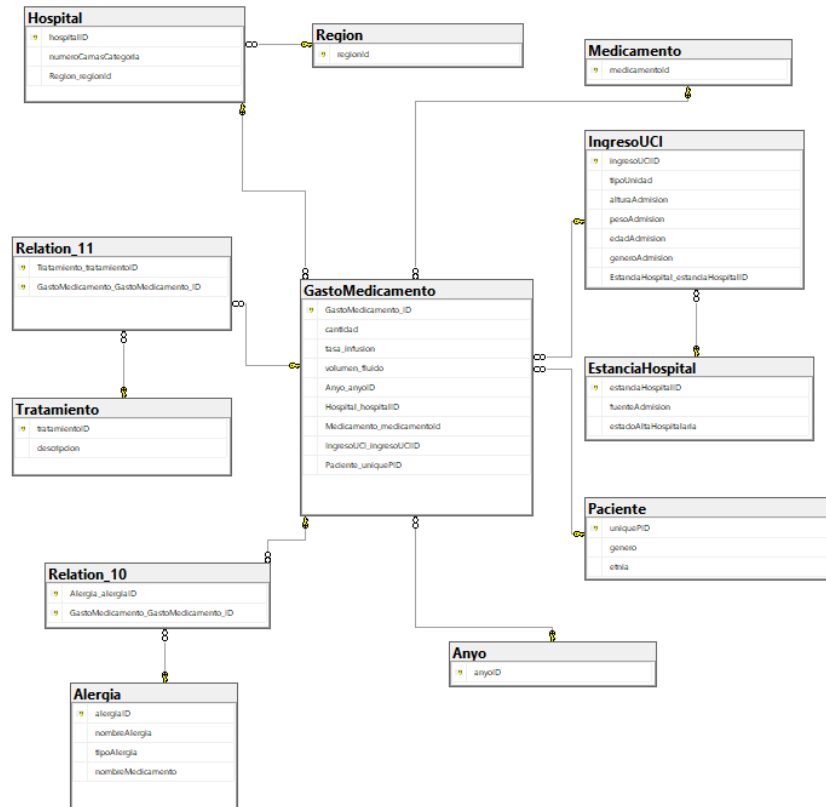


Figura 2: Modelo Lógico elaborado con *SQL Server Management Studio* [1]

Cada dimensión del modelo conceptual (*Paciente*, *Año*, *Hospital*, *Ingreso*, *Medicamento*, *Alergia* y *Tratamiento*) se transformó en una tabla individual en el modelo lógico. Las tablas de dimensión incluyen sus atributos clave, manteniendo las jerarquías definidas en el modelo conceptual. La tabla central **GastoMedicamento** representa el hecho principal del modelo y contiene las medidas de análisis, como *cantidad*, *tasa de infusión* y *volumen de fluido*. Esta tabla se conecta con las tablas de dimensión mediante claves foráneas, que reflejan las relaciones entre las dimensiones y el hecho central en la estructura relacional. De este modo, el modelo lógico permite realizar consultas eficientes sobre el gasto en medicamentos con base en las características de cada dimensión.

Para gestionar la relación de **GastoMedicamento** con las dimensiones *Alergia* y *Tratamiento*, que pueden tener una asociación de tipo **muchos-a-muchos**, fue necesario introducir tablas intermedias. Estas tablas, denominadas **Relation_10** (para la relación entre *GastoMedicamento* y *Alergia*) y **Relation_11**

(para la relación entre *GastoMedicamento* y *Tratamiento*), permiten descomponer la relación muchos-a-muchos en dos relaciones de uno-a-muchos. De esta manera, cada registro en **GastoMedicamento** puede asociarse a múltiples alergias y tratamientos sin redundancia de datos ni inconsistencias. Las tablas intermedias incluyen claves foráneas hacia **GastoMedicamento** y hacia las tablas *Alergia* o *Tratamiento*, asegurando la integridad referencial en estas relaciones.

4.2. Definición de Claves

En el modelo lógico, cada tabla de dimensión se ha diseñado con una clave primaria única, que permite identificar de manera exclusiva cada registro y facilita las consultas sobre datos específicos. Estas claves primarias son identificadores (como *PacienteID*, *HospitalID*, *MedicamentoID*, entre otros) y garantizan la integridad de los datos dentro de cada tabla.

La tabla de hechos **GastoMedicamento** utiliza, además, claves foráneas que apuntan a las claves primarias de las tablas de dimensión. Por ejemplo, *Paciente_uniquePID* se utiliza como clave foránea para enlazar la tabla de *Paciente* con **GastoMedicamento**, y de forma similar se manejan las relaciones con *Hospital*, *Medicamento*, y las demás dimensiones. Las tablas intermedias **Relation_10** y **Relation_11** también incluyen claves foráneas hacia las tablas de dimensión *Alergia* y *Tratamiento*, respectivamente, migrando las claves primarias de estas dimensiones a dichas tablas intermedias para mantener la relación de muchos-a-muchos.

Este uso de claves foráneas permite conectar el hecho del gasto en medicamentos con las distintas características de cada dimensión, proporcionando una estructura organizada y facilitando la integridad referencial en el almacén de datos. Al tener un esquema de este tipo, el modelo se encuentra normalizado y optimizado tanto para la integridad de los datos como para mejorar la eficiencia en las consultas.

4.3. Instrucciones para Crear el Almacén de Datos

Para facilitar la creación del almacén de datos, se ha adjuntado un archivo en lenguaje de definición de datos (DDL) que contiene las instrucciones necesarias para crear las tablas, claves y restricciones del esquema.

A continuación, se detallan los pasos para su ejecución en SQL Server Management Studio:

1. **Crear la base de datos:** Inicie el entorno de SQL Server Management Studio y cree una nueva base de datos.
2. **Cargar el archivo DDL:** Abra el archivo DDL adjunto. Este archivo contiene todas las definiciones de tablas y restricciones.
3. **Seleccionar la base de datos:** Incluya el comando `USE <nombre_base_de_datos>;` seguido de `GO` al inicio del archivo, reemplazando `<nombre_base_de_datos>` por el nombre de la base de datos creada.
4. **Ejecutar el script:** Ejecute el archivo DDL completo como un script. Esto creará todas las tablas, claves y restricciones requeridas para el almacén de datos.

Al finalizar, todas las estructuras necesarias para el almacén de datos estarán definidas en la base de datos seleccionada.

5. Dificultades Encontradas

A lo largo del proyecto nos hemos encontrado con varios obstáculos que nos han hecho replantear algunos aspectos del diseño del almacén de datos para la gestión hospitalaria:

1. **Comprender la base de datos de UCI:** La base de datos de UCI es bastante compleja, con muchas tablas y una gran cantidad de atributos. Nos llevó tiempo entender cómo se relacionan todos estos datos entre sí, lo cual ha sido clave para seleccionar las tablas y atributos realmente necesarios, evitando redundancias.
2. **Manejo de relaciones muchos a muchos (n:m):** Encontramos bastantes relaciones n:m, lo que complicó el diseño, ya que tuvimos que crear tablas intermedias y asegurarnos de que todo estuviera correctamente enlazado.
3. **Elegir software adecuado para el modelado conceptual:** Nos costó encontrar una herramienta que realmente se adaptara al diseño conceptual que queríamos. Muchas de las opciones no ofrecían las funcionalidades necesarias para trabajar con relaciones complejas y hechos como se necesita en un almacén de datos.

A pesar de estas dificultades, cada uno de estos retos nos ha ayudado a entender mejor las particularidades del sector salud y a preparar un diseño más sólido para el almacén de datos.

6. Conclusión

El desarrollo de este proyecto ha permitido estructurar un almacén de datos enfocado en el análisis del gasto en medicamentos en la UCI, proporcionando una herramienta que facilita el análisis eficiente y detallado de esta información crucial. Este modelo permite integrar y organizar grandes volúmenes de datos de pacientes, tratamientos y hospitalización, ofreciendo una visión completa y coherente que puede ser aprovechada para mejorar la gestión de recursos en el hospital.

Gracias a la organización del almacén de datos, se optimiza la consulta de información relevante para la toma de decisiones, facilitando la identificación de patrones de gasto asociados a diversas variables, como tratamientos específicos o características de los pacientes. Esta estructura, además, sienta las bases para futuras ampliaciones o análisis más complejos, promoviendo la escalabilidad y adaptabilidad del sistema. En conclusión, el proyecto aporta un modelo sólido y adaptable para la gestión y análisis de datos en el ámbito hospitalario, contribuyendo a una administración más eficiente de los recursos y a una mejora potencial en la atención a los pacientes.

7. Acceso al Repositorio

Toda la información adicional, incluyendo el código fuente y la documentación completa de este proyecto, está disponible en el repositorio de GitHub [5].

Referencias

- [1] Microsoft Corporation. *SQL Server Management Studio*, 2024. Herramienta de gestión para SQL Server, utilizada para visualizar el modelo lógico.
- [2] Oracle Corporation. *Oracle SQL Developer Data Modeler*, 2024. Herramienta para modelado de bases de datos, permite diseñar diagramas ER.
- [3] MIT Laboratory for Computational Physiology. eICU Collaborative Research Database. <https://eicu-crd.mit.edu/>, 2020. Último acceso: 8 noviembre 2024.
- [4] JGraph Ltd. *draw.io: Diagram Software*, 2024. Versión en línea para diagramas de flujo y diseño de bases de datos.
- [5] Alex Silva. *Healthcaredatawarehouse*. <https://github.com/AlexSilvaa9/HealthcareDataWarehouse>, 2024. Último acceso: 1 octubre 2024.