



UNIVERSIDAD DE MÁLAGA



Trabajo Almacenes de Datos

# Análisis de datos (MDX)

Crear un cubo multidimensional del almacén para  
poder realizar consultas en MDX.

Realizado por  
De Pablo Diego y Soriano Juan

Profesor encargado:  
Luque Baena, Rafael Marcos

Departamento  
Lenguajes y Ciencias de la Computación

MÁLAGA, DICIEMBRE de 2024



UNIVERSIDAD  
DE MÁLAGA



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA  
ESTUDIANTES DE INGENIERÍA BIOINFORMÁTICA

# **Análisis de datos (MDX)**

## **Almacenes de Datos**

Realizado por  
**De Pablo Diego y Soriano Juan**

Profesor encargado:  
**Luque Baena, Rafael Marcos**

Departamento  
**Lenguajes y Ciencias de la Computación**

UNIVERSIDAD DE MÁLAGA  
MÁLAGA, DICIEMBRE DE 2024

# Contents

<b>1</b>	<b>Introducción</b>	<b>3</b>
<b>2</b>	<b>Objetivos</b>	<b>3</b>
<b>3</b>	<b>Modificación del almacén de datos</b>	<b>3</b>
<b>4</b>	<b>Creación del cubo</b>	<b>5</b>
4.1	Instrucciones para desplegar el proyecto en el equipo . . . . .	5
<b>5</b>	<b>Consultas en MDX</b>	<b>6</b>
<b>6</b>	<b>Instrucciones para ejecutar las consultas.</b>	<b>6</b>
<b>7</b>	<b>Problemas encontrados</b>	<b>7</b>
<b>8</b>	<b>Conclusión</b>	<b>7</b>
<b>9</b>	<b>Github y conjunto de instrucciones para su correcto despliegue en SQL Server.</b>	<b>7</b>

# 1 Introducción

Este documento es una continuación de las fases anteriores del proyecto, donde se desarrollaron el **diseño conceptual y lógico** de un almacén de datos basado en la información proporcionada por la *Base de Datos de Investigación Colaborativa eICU* [1], y la **integración de datos** mediante un proceso de Extracción, Transformación y Carga (ETL). Para la creación de un almacén enfocado a **pacientes con patologías respiratorias**.

En esta fase, se busca construir un **cubo multidimensional** utilizando los datos del almacén previamente desarrollado, En el ámbito hospitalario, la creación de un cubo multidimensional ofrece ventajas al permitir analizar datos clínicos complejos de manera ágil. Posteriormente se realizarán consultas en *MDX* (Multidimensional Expressions) sobre dicho cubo. Permitiendo extraer información de manera eficiente al navegar por las dimensiones y medidas del cubo.

Además, se describirán los problemas encontrados durante el desarrollo y las soluciones aplicadas, asegurando que las consultas generen resultados útiles para el análisis de datos.

## 2 Objetivos

Los objetivos principales de este informe son los siguientes:

- Realizar las correcciones pertinentes en el proceso ETL que permitan el desarrollo del cubo multidimensional.
- Implementar un cubo multidimensional adaptado para el análisis de **pacientes con patologías respiratorias**.
- Desarrollar al menos 8 consultas en *MDX* que exploren diferentes dimensiones y hechos del cubo, aplicando funciones avanzadas cuando sea necesario.
- Documentar de manera clara y replicable el proceso de creación del cubo multidimensional y las consultas *MDX*, proporcionando instrucciones detalladas para su ejecución.
- Describir los problemas encontrados durante el desarrollo del cubo y las consultas, y detallar las soluciones empleadas.

## 3 Modificación del almacén de datos

- Se eliminaron las columnas ‘Age’ y ‘PatientUnitStayID’ de la tabla de pacientes.
- Se seleccionaron únicamente los pacientes únicos, conservando solo su primera aparición en la base de datos. Anteriormente, el número total de pacientes era 1849, pero después de aplicar esta modificación, el total es de 1841, lo que es correcto.

- Para el tiempo, se utilizó la cláusula ‘DISTINCT’ y, para la información de diagnóstico, se extrajo el primer valor de la columna ‘DiagnosisString’ (primera dimensión). La consulta SQL utilizada para obtener los datos de diagnóstico de la base de datos fue la siguiente:

```
SELECT
DiagnosisID,
PatientUnitStayID,
ActiveUponDischarge,
DiagnosisOffset,
ICD9Code,
DiagnosisPriority,
SUBSTRING(DiagnosisString, 1, CHARINDEX('|'
, DiagnosisString + '|') - 1) AS FirstDiagnosis
FROM Diagnosis;
```

- Se corrigió la creación de las tablas intermedias para las relaciones ‘NxM’. Anteriormente, se utilizaba como origen de datos la tabla ‘Diagnosis’ de la base de datos en lugar de la del ‘Data Warehouse’. Posteriormente, se realizaba un ‘lookup’ con ‘IngresoUCI’ utilizando el ‘patientID\_og’.
- Se realizó una transformación en la columna ‘Age’, cambiando su tipo de dato de ‘string’ a ‘integer’ para permitir su uso en consultas. Durante este proceso, se transformó el valor ‘> 89’ a ‘90’ por defecto. La consulta SQL utilizada para obtener los datos de la tabla ‘Patient’ en la base de datos fue la siguiente:

```
SELECT
T.TiempoID,
P.UniquePID,
H.HospitalID,
I.HospitalDischargeOffset,
I.PatientHealthSystemStayID,
I.PatientUnitStayID,
CASE
WHEN I.Age = '> 89' THEN 90
-- Cambiar "> 89" por 90
WHEN ISNUMERIC(I.Age) = 1 THEN CAST(I.Age AS INT)
-- Convertir valores numéricos a INT
ELSE NULL
-- Manejar otros casos como NULL
END AS AgeInt
FROM [eICU Collaborative Research Database].dbo.Patient I
LEFT JOIN prueba.dbo.Paciente P ON I.uniquePID = P.uniquePID_og
LEFT JOIN prueba.dbo.Hospital H ON I.HospitalID = H.hospitalID_og
LEFT JOIN prueba.dbo.Tiempo T
ON I.HospitalDischargeYear = T.HospitalDischargeYear
AND I.HospitalDischargeTime24 = T.HospitalDischargeTime24;
```

## 4 Creación del cubo

Para la creación del cubo, iniciaremos el proceso estableciendo la conexión con nuestra base de datos:

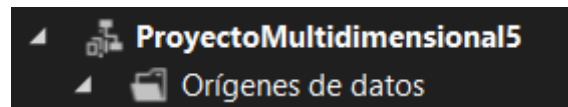


Figure 1: Origen de datos

A continuación, procederemos a establecer la conexión con la base de datos, donde especificaremos el proveedor a utilizar, como se muestra en la Figura 2. Dado que nuestro servidor se encuentra en el entorno local, se debe indicar únicamente . como la dirección del servidor. Es crucial, **y se debe prestar especial atención**, que la autenticación se configure utilizando SQL Server Authentication. Esto permitirá especificar el nombre de usuario como sa y la contraseña asociada, que en este caso será Almacenes. Finalmente, seleccionaremos la base de datos correspondiente, que en este escenario es prueba, como se ilustra en la Figura 2.

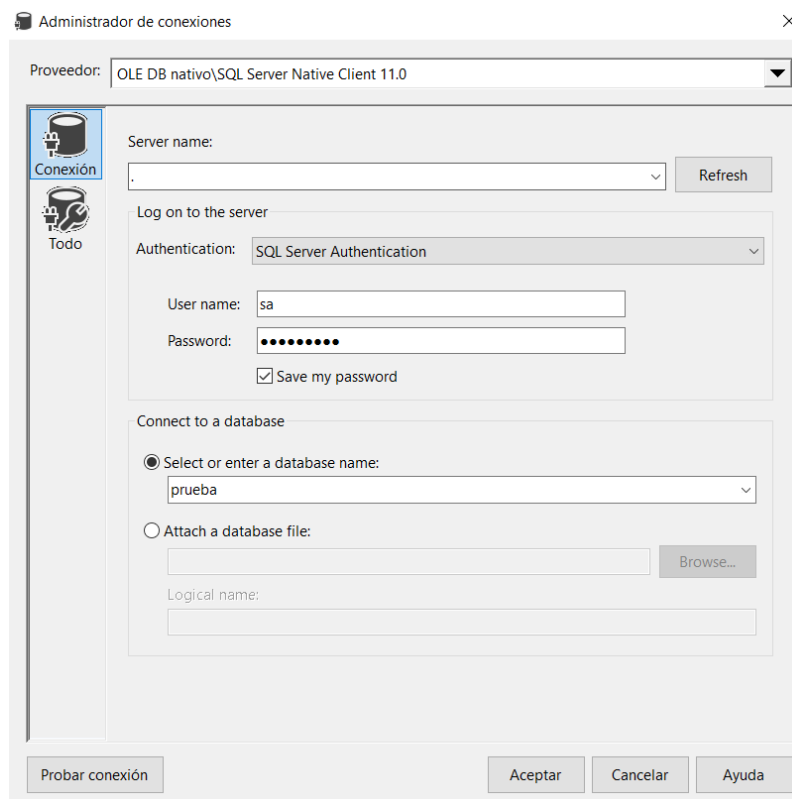


Figure 2: Conexión de datos

### 4.1 Instrucciones para desplegar el proyecto en el equipo

Para ejecutar la tarea, será suficiente con descomprimir el archivo. A continuación, se deberá hacer clic derecho sobre el proyecto multidimensional y seleccionar la opción "Propiedades". Una vez en el apartado de propiedades, se procederá a acceder a la sección de implementación, donde se deberá seleccionar el servidor

correspondiente. En este caso, se podrá elegir como servidor ‘localhost’ o bien ‘localhost\nuestroServidorSQL’. En mi situación particular, dado que disponía de dos instancias de ‘MSSQLSERVER’, fue necesario especificar que se utilizaría la instancia ‘MSSQLSERVER2’, que es la que posee la funcionalidad multidimensional.

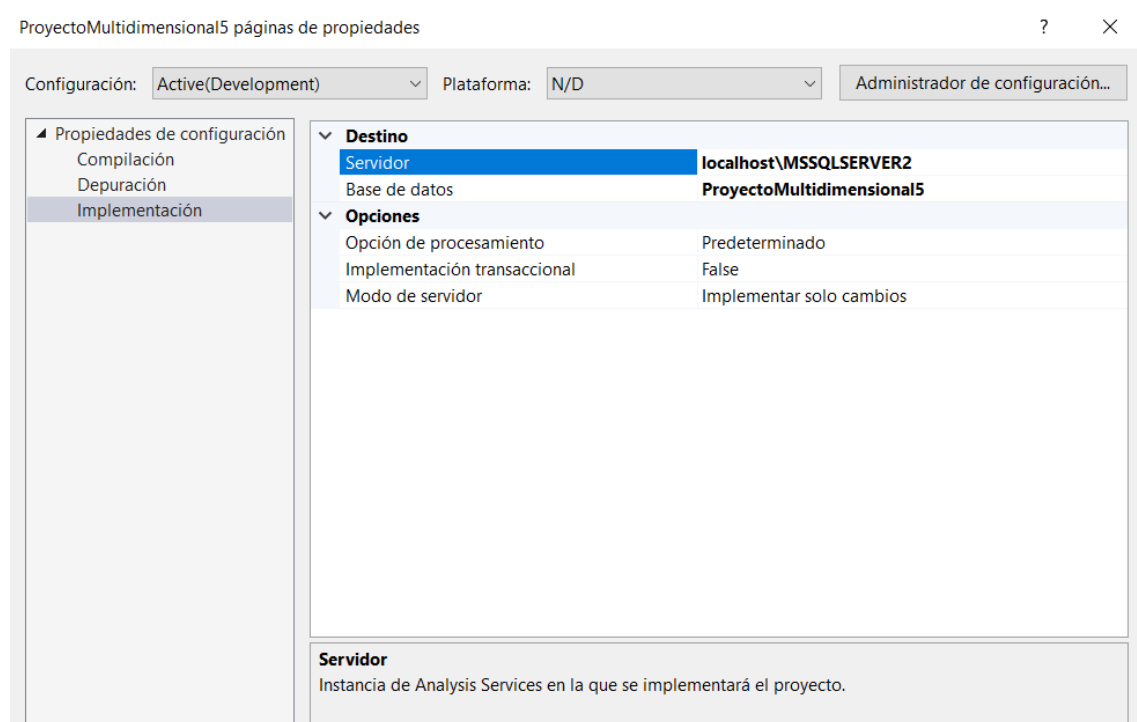


Figure 3: Configuración de despliegue

Una vez configurado todo correctamente, se procederá a hacer clic en “Iniciar” y el proceso se completará de manera satisfactoria.

Posteriormente, iniciaremos los servicios de Análisis (Analysis Services) en SQL Server Management Studio y verificaremos que tanto el proyecto *ProyectoMultidimensional5* como el cubo se han desplegado correctamente.

## 5 Consultas en MDX

- Una sección con las consultas en MDX y una captura con el resultado de cada una de ellas (la imagen capturada no tiene por qué mostrar todas las tuplas resultantes).

## 6 Instrucciones para ejecutar las consultas.

- Una sección con las instrucciones detalladas para que un evaluador pueda ejecutar las consultas en su máquina.

- Para cada consulta MDX, se muestra, además del enunciado de la consulta en sí, la consulta realizada en MDX y una captura del resultado generado.

- Indica el número de consultas MDX que se han intentando, es decir, que se ha escrito código MDX y se ha mostrado el resultado de la misma, independientemente de si son correctas o no:

(\*\*\*) 8 o más.

- Según tu experiencia, valora cómo están realizadas las consultas en MDX, seleccionando la opción más adecuada:

(\*\*\*\*) Todas las consultas parecen ser correctas, teniendo sentido la salida de cada una de ellas.

- En algunas consultas se han usado funciones más avanzada de MDX, que impliquen el uso de métodos para recorrer una jerarquía (PREVMEMBER, CURRENTMEMBER, PARENT, etc.).

## 7 Problemas encontrados

Maldita maquina virtual

- Una sección "problemas encontrados" que explique con cierto detalle los problemas que se han encontrado durante la realización de la práctica. Se permite que la información incluida en esta sección se encuentre dividida o dispersada a lo largo del documento.

- La sección de "problemas encontrados" (o similar) es coherente y explica con cierto detalle los problemas que se han encontrado durante la realización de la práctica (ojo, no tienen por qué haber sido resueltos). Se permite que la información incluida en esta sección se encuentre dividida o dispersada a lo largo del documento.

- Calificación final.

## 8 Conclusión

XD

## 9 Github y conjunto de instrucciones para su correcto despliegue en SQL Server.

Todo el proyecto está accesible en github [2] donde se detalla más específicamente como desplegar en SQL.

## References

- [1] eICU Collaborative Research Database. *eICU Collaborative Research Database*. <https://eicu-crd.mit.edu/about/eicu/>. Accessed: 2024-11-14. 2024. URL: <https://eicu-crd.mit.edu/about/eicu/>.
- [2] Diegodepab. *Almacén UCI Sanitaria*. [https://github.com/Diegodepab/almacen\\_UCI\\_Sanitaria](https://github.com/Diegodepab/almacen_UCI_Sanitaria). Accessed: 2024-11-14. 2024. URL: [https://github.com/Diegodepab/almacen\\_UCI\\_Sanitaria](https://github.com/Diegodepab/almacen_UCI_Sanitaria).





UNIVERSIDAD  
DE MÁLAGA

| **uma.es**

E.T.S. DE INGENIERÍA INFORMÁTICA

E.T.S de Ingeniería Informática  
Bulevar Louis Pasteur, 35  
Campus de Teatinos  
29071 Málaga