



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>4</b>
<b>2</b>	<b>Objetivos</b>	<b>4</b>
<b>3</b>	<b>Modificación del almacén de datos</b>	<b>4</b>
<b>4</b>	<b>Creación del cubo</b>	<b>6</b>
4.1	Orígenes de datos . . . . .	6
4.2	Crear el cubo . . . . .	9
4.3	Modificación de las dimensiones . . . . .	11
4.3.1	Ethnicity - Paciente . . . . .	11
4.3.2	Gender - Paciente . . . . .	13
4.3.3	Hospital . . . . .	14
4.3.4	IngresoUCI . . . . .	16
4.3.5	Allergy . . . . .	16
4.3.6	Diagnosis . . . . .	17
4.3.7	Medication . . . . .	19
4.3.8	Tiempo . . . . .	20
4.4	Modificaciones del cubo . . . . .	23
4.4.1	Nuevas medidas . . . . .	24
4.4.2	Despliegue del cubo . . . . .	24
4.5	Instrucciones para desplegar el proyecto en el equipo . . . . .	25
<b>5</b>	<b>Consultas en MDX</b>	<b>26</b>
5.1	Consulta 1: Total de pacientes ingresados en hospital del oeste por género y edad . . . . .	26
5.2	Consulta 2: Número de medicamentos usados por cada etnia y región de hospital. . . . .	27
5.3	Consulta 3: Promedio de PaO2, FiO2 y Frecuencia Respiratoria por Región Hospitalaria y Etnia para Pacientes de 20 a 30 Años en UCI . . . . .	28
5.4	Consulta 4: Cálculo de Porcentaje Acumulado de Ingresos UCI por Región Hospitalaria . . . . .	30
5.5	Consulta 5: Recuento de Ingresos a UCI por Hora y Total Anual . . . . .	31
5.6	Consulta 6: Recuento de Diagnósticos por Edad y prioridad en un Intervalo de Tiempo Específico (00:00:00 a 12:00:00 del 2014) . . . . .	32
5.7	Consulta 7: Cantidad de alergias, para los pacientes ingresados y Tasa Respiratoria Media por Edad, Tipo de Alergia y Diagnóstico para Pacientes Hispanos en 2015 . . . . .	34
5.8	Consulta 8: Análisis de Medicación y Parámetros Respiratorios por Género y Año, Excluyendo Pacientes de Etnia Hispánica y con Ventilación Distinta de Cero . . . . .	35
<b>6</b>	<b>Instrucciones para ejecutar las consultas MDX</b>	<b>37</b>
6.1	Acceso al servidor de Analysis Services . . . . .	37
6.2	Ejecución de consultas MDX . . . . .	37
6.3	Recomendaciones para la ejecución de consultas . . . . .	37

<b>7</b>	<b>Problemas encontrados</b>	<b>38</b>
<b>8</b>	<b>Conclusión</b>	<b>38</b>
<b>9</b>	<b>Github y conjunto de instrucciones para su correcto despliegue en SQL Server.</b>	<b>38</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’ para evitar **fechas duplicadas en el ingreso**.

- 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

### 4.1 Orígenes de datos

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

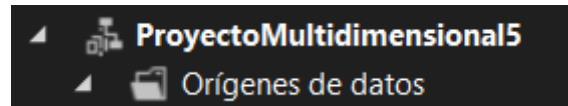


Figure 1: Origen de datos

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.

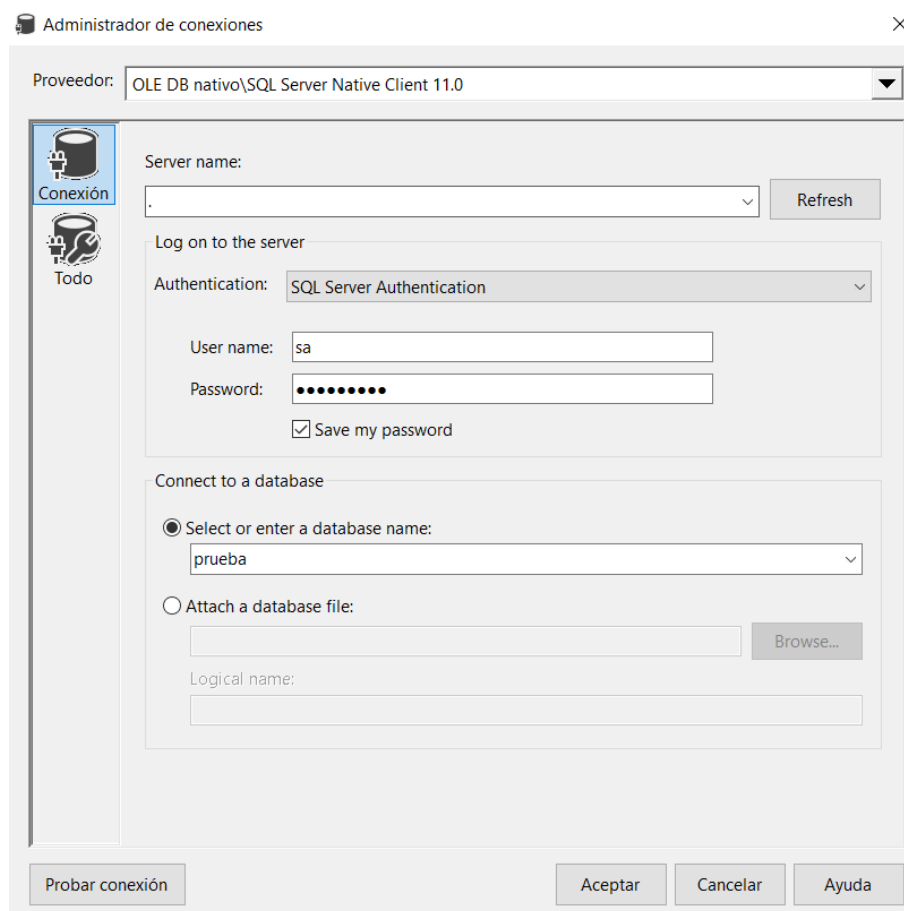


Figure 2: Conexión de datos

Cabe destacar que en la creación de un cubo se puede seleccionar mejores nombre

que "prueba", pero por simplicidad y facilitar el seguimiento de este tutorial, en caso de crear una base de datos se recomendaría usar un nombre más coherente con los datos trabajados, pero al igual que muchos lenguajes de programación se le puede asignar el nombre deseado mientras se mantenga la lógica y la correcta llamada de la misma.

Una vez realizados los pasos anteriores, procederemos a hacer clic derecho sobre las **vistas del origen de datos** y seleccionaremos la opción "Nueva Vista". Esto nos mostrará todos los orígenes de datos disponibles, siendo en este caso solo la base de datos "Prueba", como se muestra en la Figura 3.

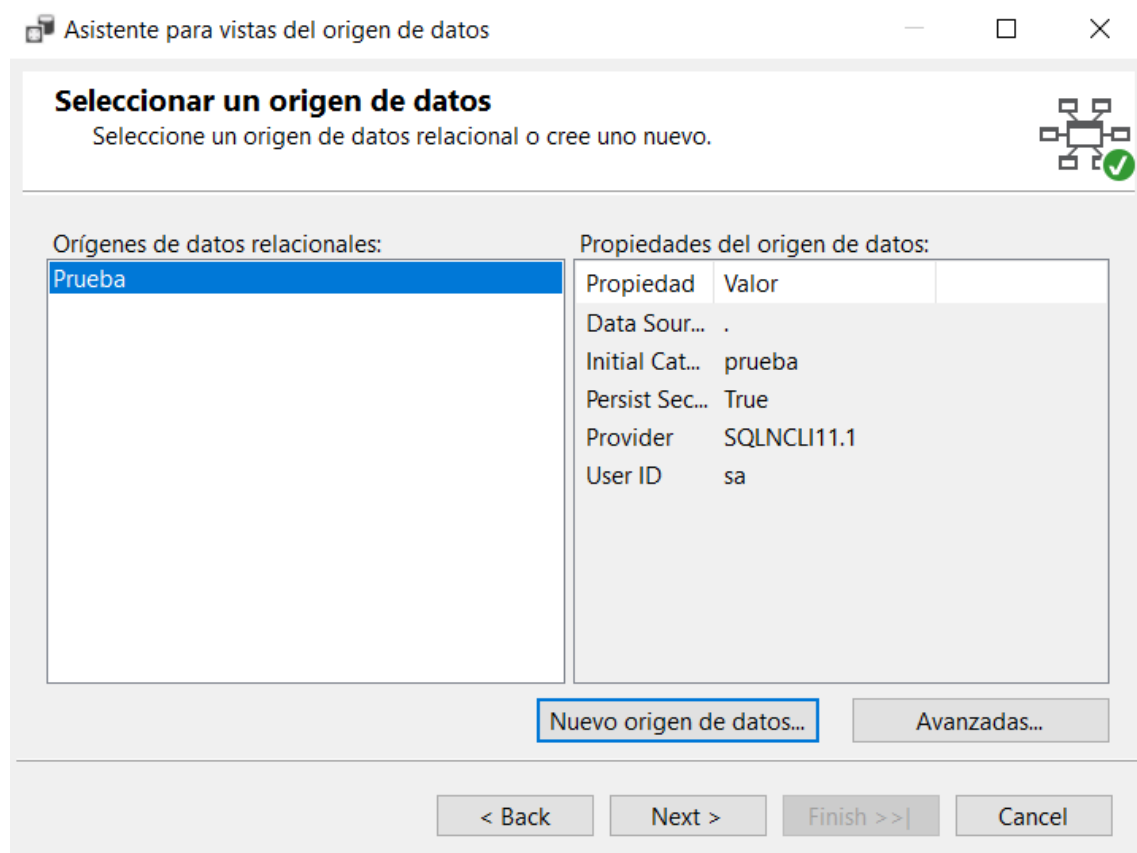


Figure 3: **Vista origen conexión de datos**

A continuación, procederemos a hacer clic en el botón "Next".

En la pantalla siguiente, tal como se muestra en la Figura 4, seleccionaremos el **botón resaltado en rojo**. Esta acción nos permitirá agregar **todas las tablas a la vista**, lo que es fundamental para asegurar que el almacén de datos incluya toda la información necesaria para la correcta construcción del cubo. Una vez seleccionadas todas las tablas, continuaremos haciendo clic en "Next" para avanzar al siguiente paso del proceso.



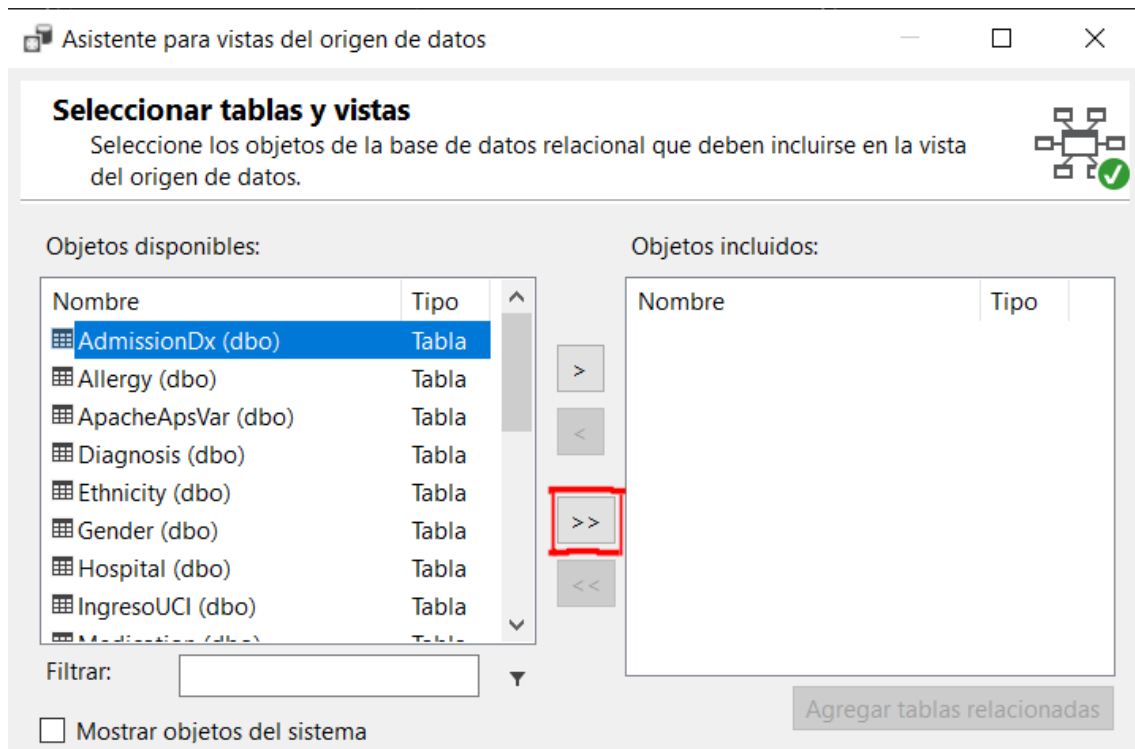


Figure 4: Selección de tablas

Posteriormente, aparecerá una nueva pantalla donde podremos **visualizar lo que contendrá la vista** y, por ende, con qué elementos podremos operar en el cubo, como se muestra en la Figura 5.

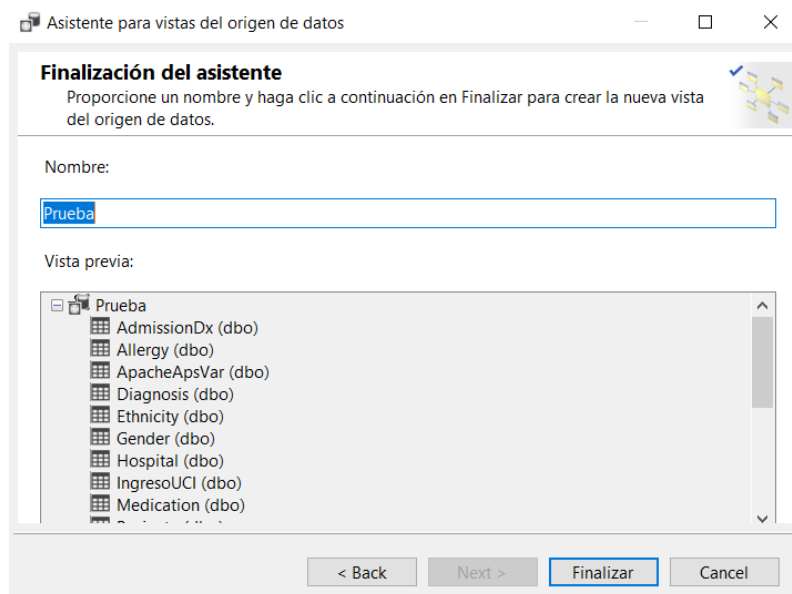


Figure 5: Vista finalización

Una vez revisado todo, seleccionaremos "Finalizar" como se muestra en la Figura 5.

Al hacer **doble clic en la vista recién creada**, podremos ver que contiene todas las tablas que teníamos en nuestro diagrama de SQL Server Management, como se ilustra en la Figura 6.

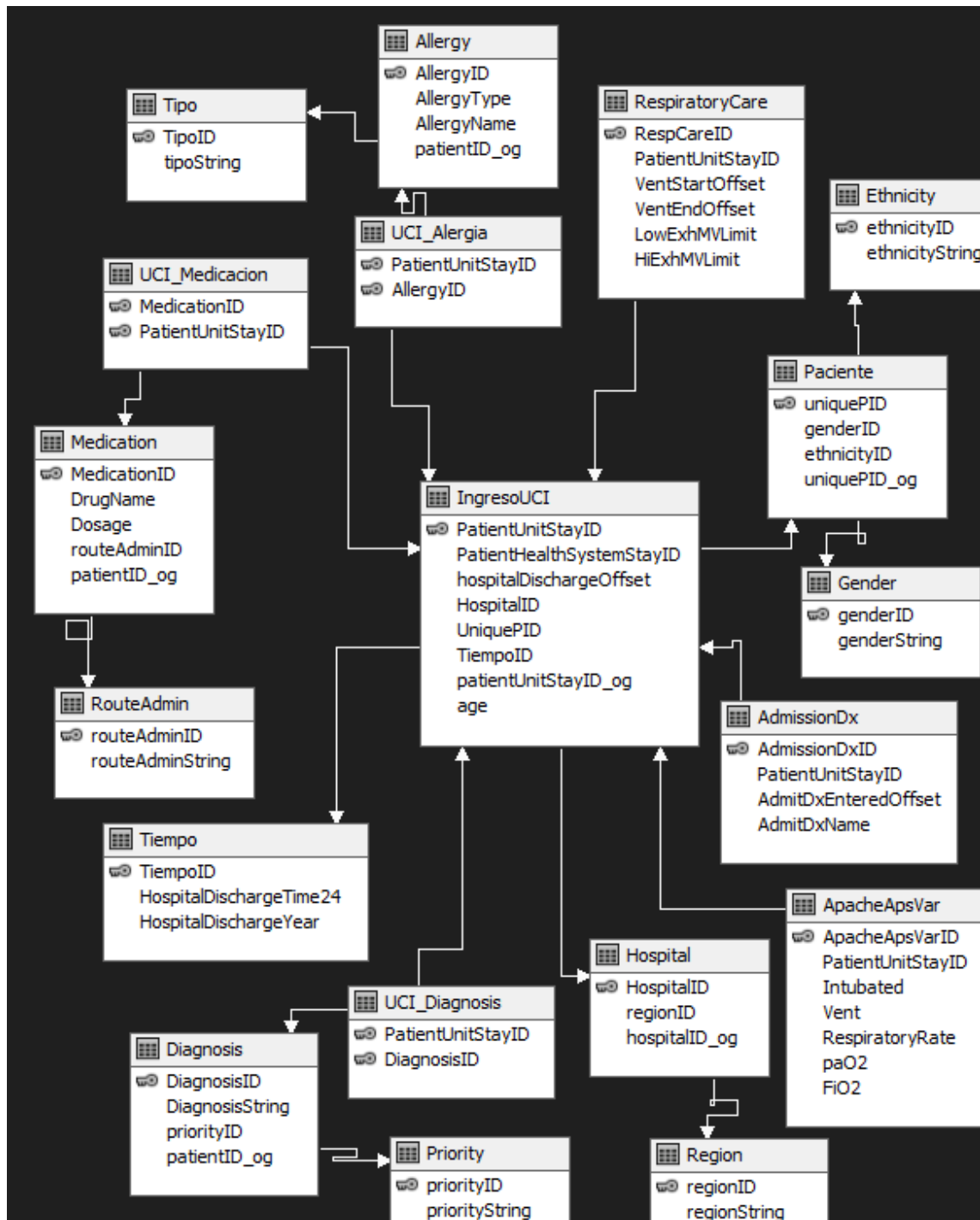


Figure 6: Vista

## 4.2 Crear el cubo

Una vez que la **vista esté configurada**, podemos proceder a **crear el cubo**. Para ello, haremos clic derecho sobre el proyecto y seleccionaremos "Nuevo Cubo". En la

ventana emergente, elegiremos la opción **"Usar tablas existentes"**, tal como se muestra en la Figura 7.

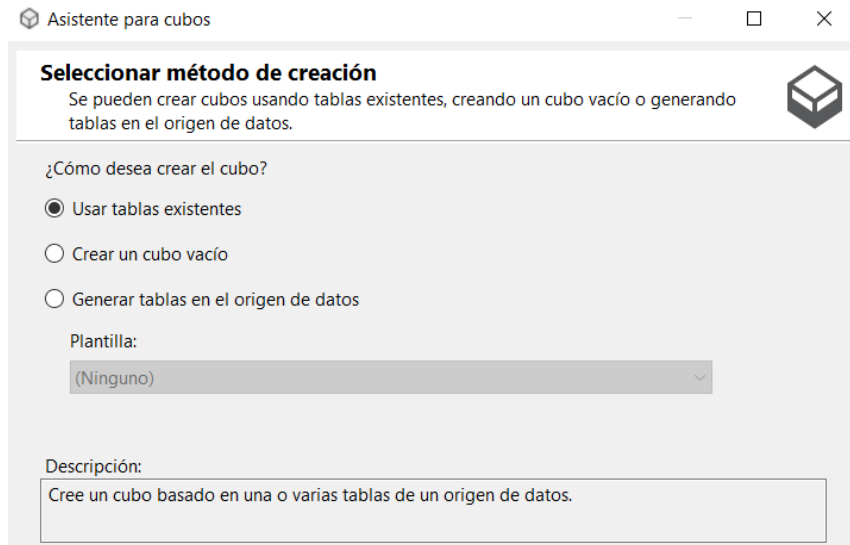


Figure 7: Creación de nuevo cubo

Luego, haremos clic en "Next" y aparecerá una pantalla como la que se muestra en la Figura 8.

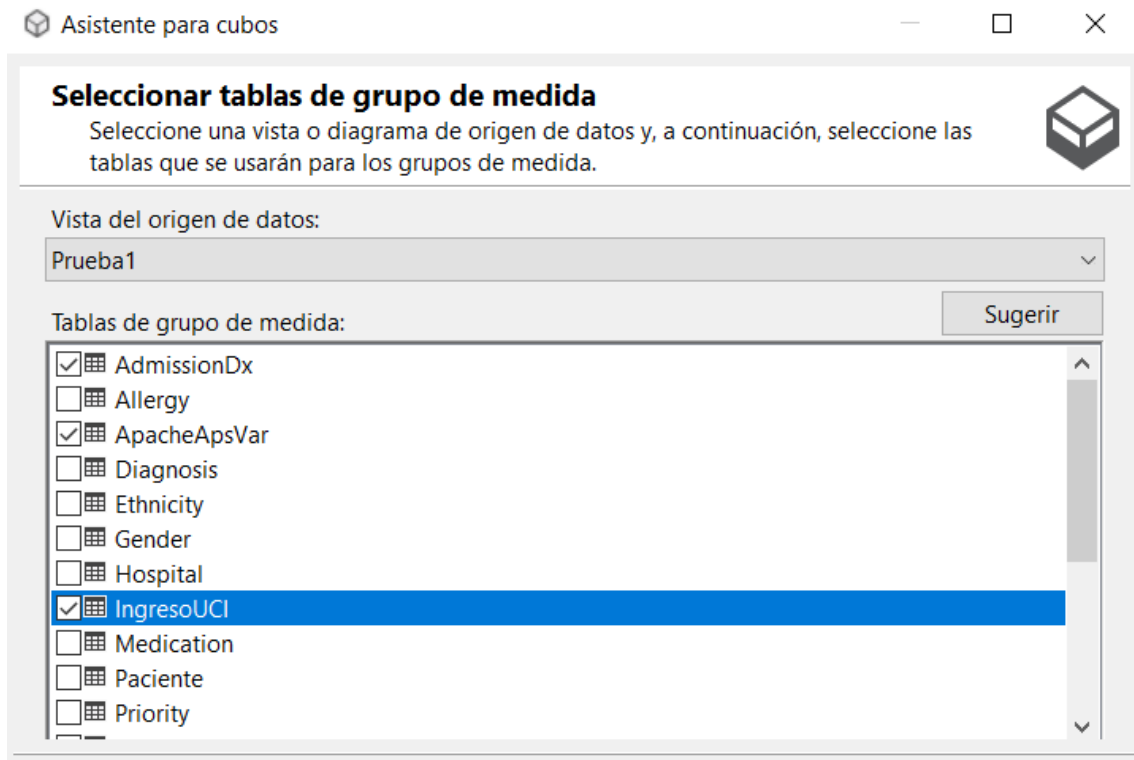


Figure 8: Selección tablas del Cubo

En esta pantalla, seleccionaremos únicamente aquellas **tablas que tengan una relación de tipo 1 a N** con la tabla de hechos, donde N representa las tablas rela-

cionadas y 1 es la tabla de hechos. Esto incluye las tablas *AdmissionDx*, *ApacheApsVar*, *RespiratoryCare*, así como las tablas intermedias (*UCI\_Medicacion*, *UCI\_Diagnosis* y *UCI\_Alergia*) resultantes de las relaciones NxM entre las tablas *Medication*, *Diagnosis* y *Allergy*.

Si hemos seguido correctamente los pasos anteriores, el resultado será similar al mostrado en la Figura 9.

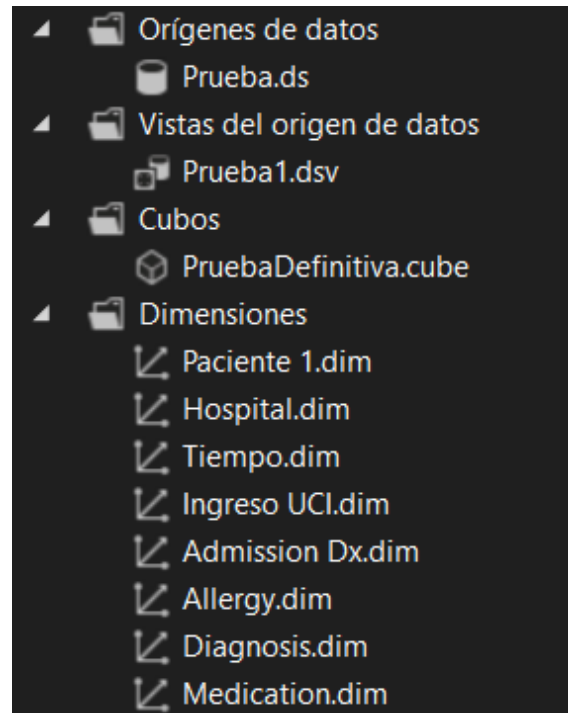


Figure 9: Proyecto Multidimensional

Cabe destacar que las dimensiones se generan automáticamente a partir de la creación del cubo.

### 4.3 Modificación de las dimensiones

Ahora procederemos con la modificación de las dimensiones para su uso correcto en el cubo.

Si damos doble clic en la dimensión **Paciente** y nos vamos al apartado de "Estructura de dimensión", veremos en la columna de atributos las claves almacenadas en **Paciente**, como se muestra en la Figura 10. Ahora, lo que haremos será crear jerarquías para utilizarlas posteriormente en las consultas.

Nota: **Para cada cambio relacionado con las dimensiones, será necesario procesar y actualizar la dimensión.** Para ello, tendremos que estar en Browser o Explorador en Español.

#### 4.3.1 Ethnicity - Paciente

En el caso de **Paciente**, teníamos una doble jerarquía. Por un lado, era el género y por otro, la etnia. Para crear estas jerarquías, arrastraremos la clave de la jerarquía

que queramos empezar, ya sea GenderID o EthnicityID, al área donde aparece "Jerarquías". Luego, arrastraremos la clave UniquePID. Esto dará lugar a dos jerarquías similares a las mostradas en la Figura 10.

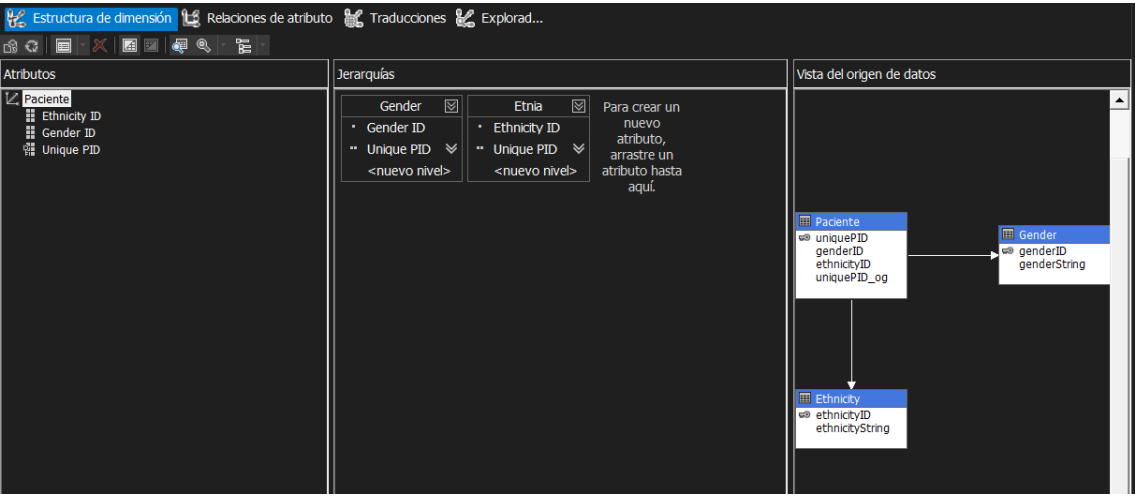


Figure 10: Dimensión Paciente

Posteriormente, para que las jerarquías se entiendan correctamente y no se utilicen solo ID's, será necesario hacer los siguientes cambios.

Daremos clic en EthnicityID y en el recuadro inferior derecho nos aparecerán las propiedades de dicho atributo, como se muestra en la Figura 11.

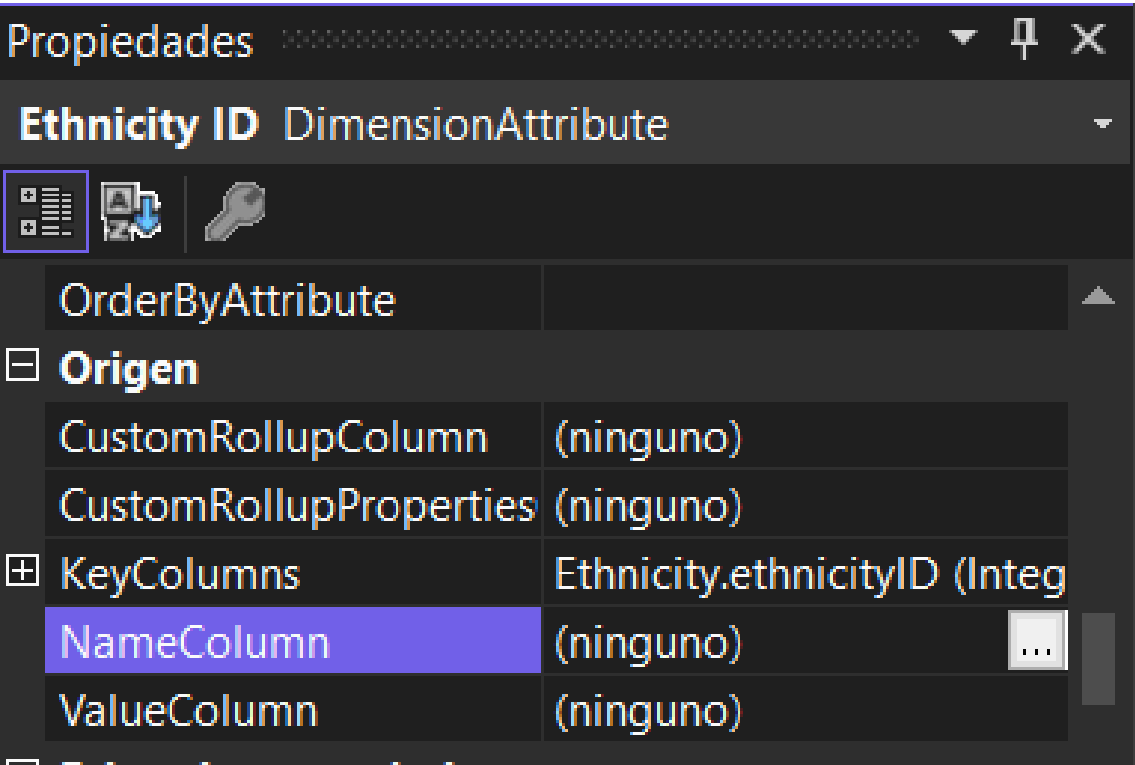


Figure 11: EthnicityID

Nos iremos al apartado **NameColumn**, le daremos a los tres puntos y nos saldrá una ventana parecida a la Figura 12:

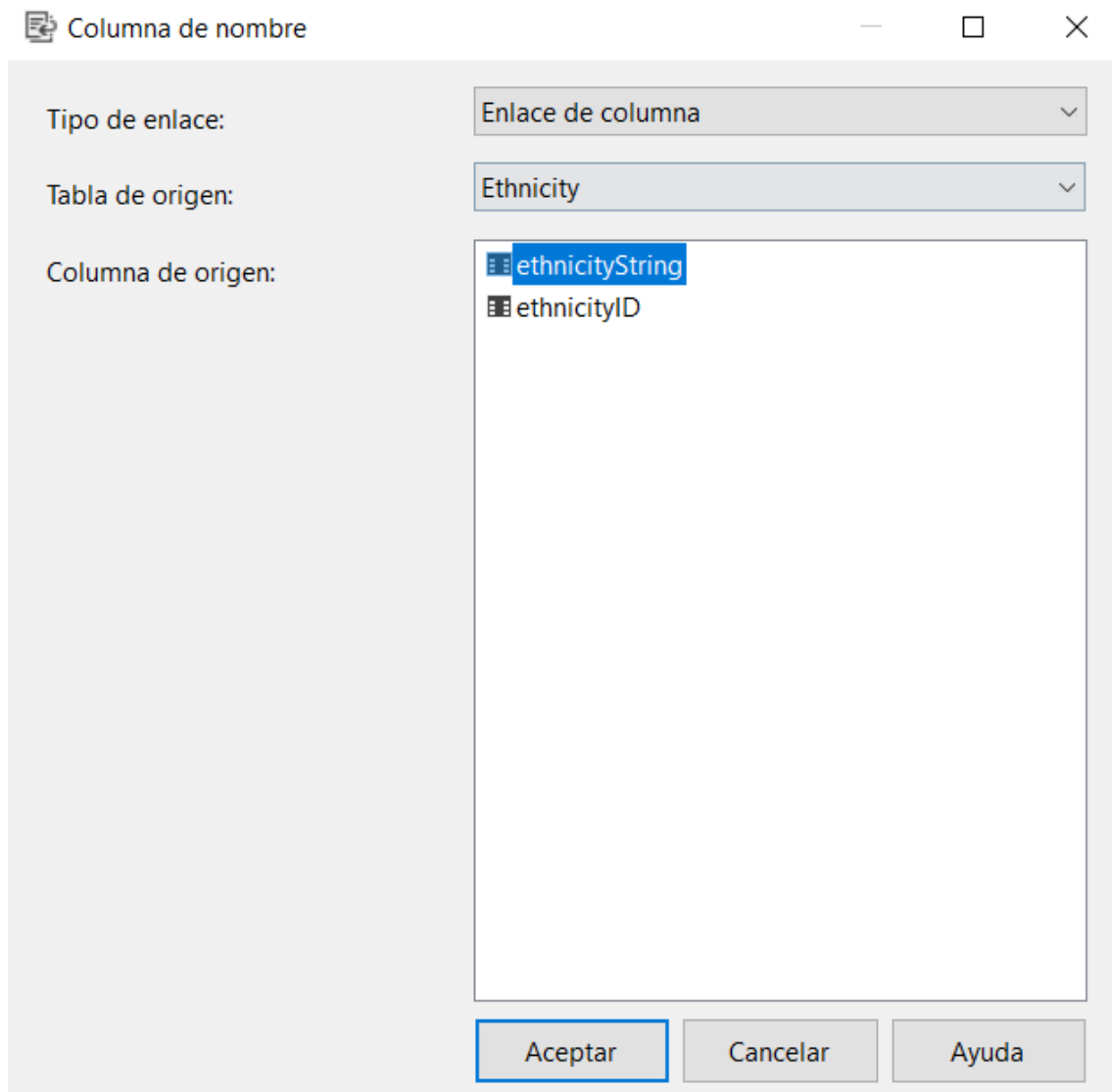


Figure 12: EthnicityID

Ahora seleccionaremos **ethnicityString** como **NameColumn**. Ya por último, siguiendo en el apartado de propiedades, tendremos que poner en **ValueColumn** el valor **ethnicityString**, también, ya que no queremos realmente que las jerarquías sean por el ID sino por el string asociado a ellos.

Haremos exactamente el mismo proceso para el resto de tablas, **exceptuando Tiempo**, que la comentaré más tarde.

#### 4.3.2 Gender - Paciente

**ValueColumn** y **NameColumn**:

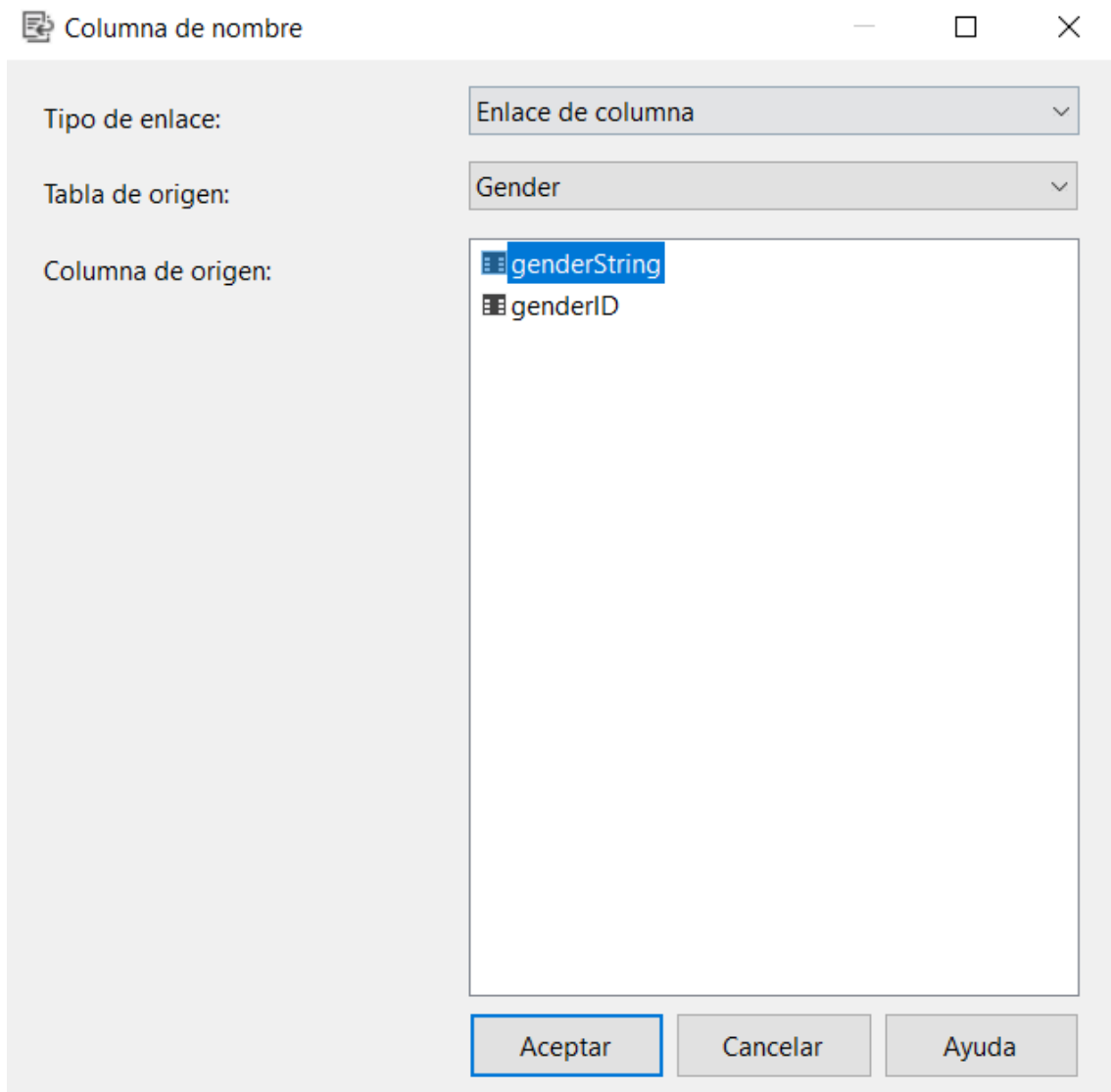


Figure 13: GenderID

Seleccionamos **genderString**. Al igual que antes esto es un paso "opcional" pero que mejora la calidad de nuestro cubo, si se ha probado desplegar el cubo sin cambiar estos puntos, dará resultados y se podrá hacer consultas, el problema es que estas serán más confusas, donde en lugar de hacer consultas con una etiqueta como podría ser ".male" sería correspondiente al id autogenerado para male, que vendría a ser un número que no refleja correctamente su uso.

#### 4.3.3 Hospital

En el caso del hospital al ser parecido al caso anterior se empezará a obviar pasos para evitar redundancia, siendo las principales selecciones:

**Jerarquía:** Se asigna que los hospitales tengan otro nivel donde se puedan organizar según la región del hospital.

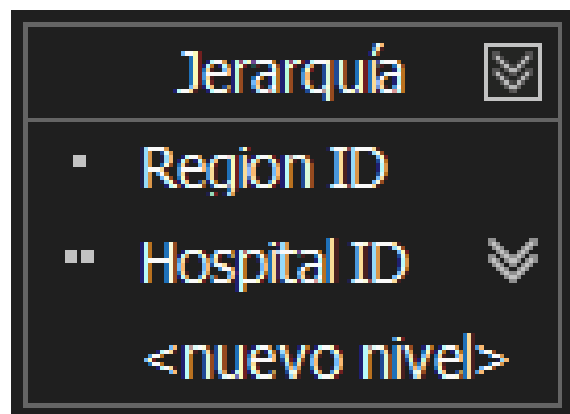


Figure 14: Jerarquía de Region

**ValueColumn** y **NameColumn** son cambiados por su versión en string que de mayor contexto.

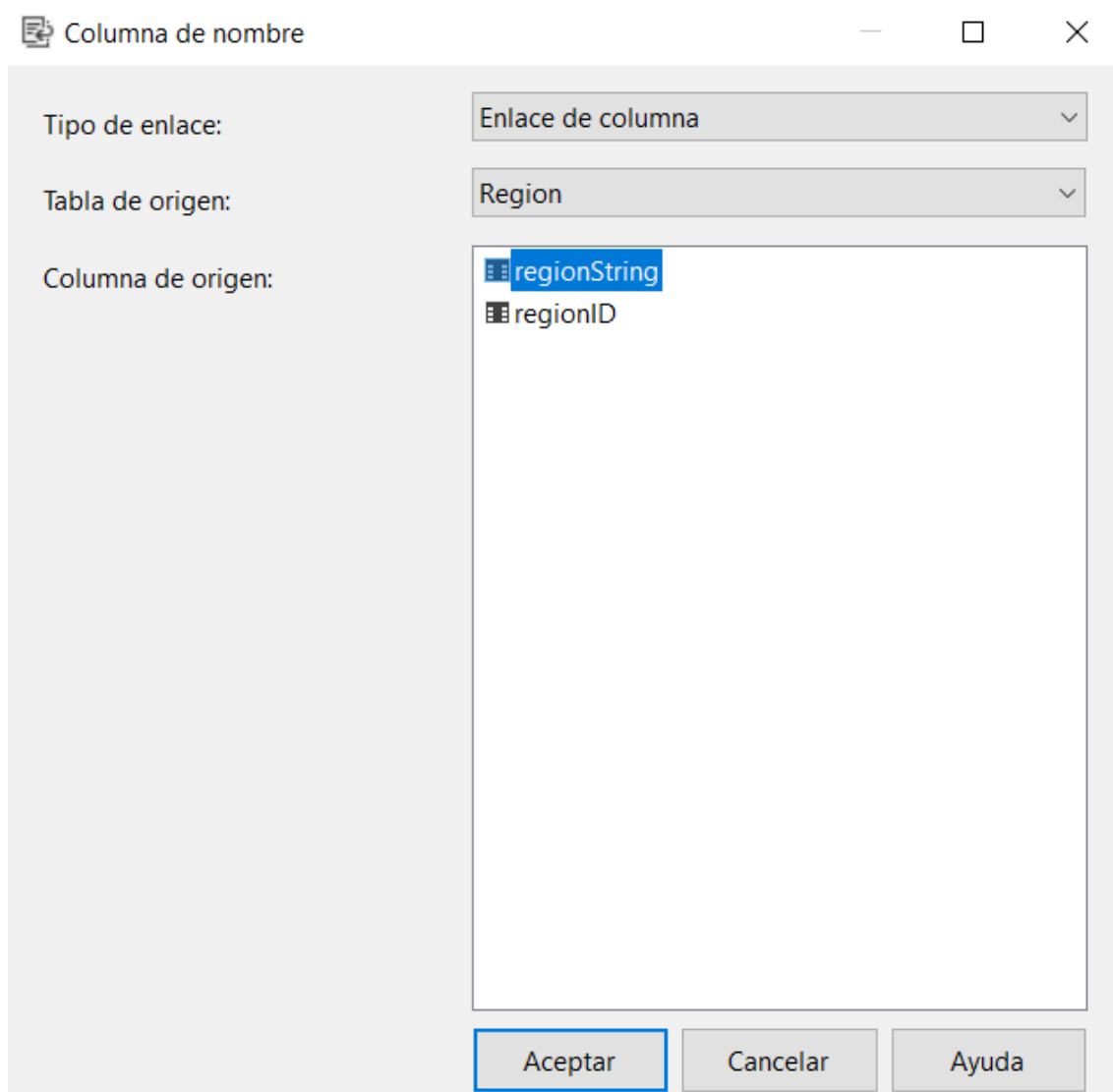


Figure 15: RegionID



En otras bases de datos el nivel más bajo de granularidad probablemente cuente con su respectivo String, como puede ser en los almacenes de comercio donde los productos, empresas, tiendas, etc. Tienen sus nombres. En el caso de la salud contamos con el inconveniente de la sensibilidad de los datos, llegando a tener el RegionString que indica la zona del hospital, pero de los hospitales solo se cuenta con los IDs asociados, que vendrían a ser un número autogenerated, al no contar con sus nombres solo se podrá cambiar los nombres del nivel de región.

#### 4.3.4 IngresoUCI

Jerarquía:

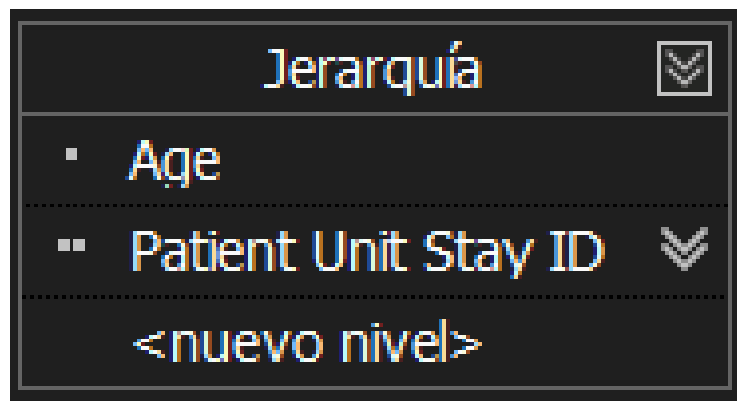


Figure 16: Jerarquía de IngresoUCI

En este nivel se tomo la decisión de tomar la edad como medidas y como dimensión para dar mayor facilidad a la hora de hacer consultas demográficas.

#### 4.3.5 Allergy

Jerarquía:

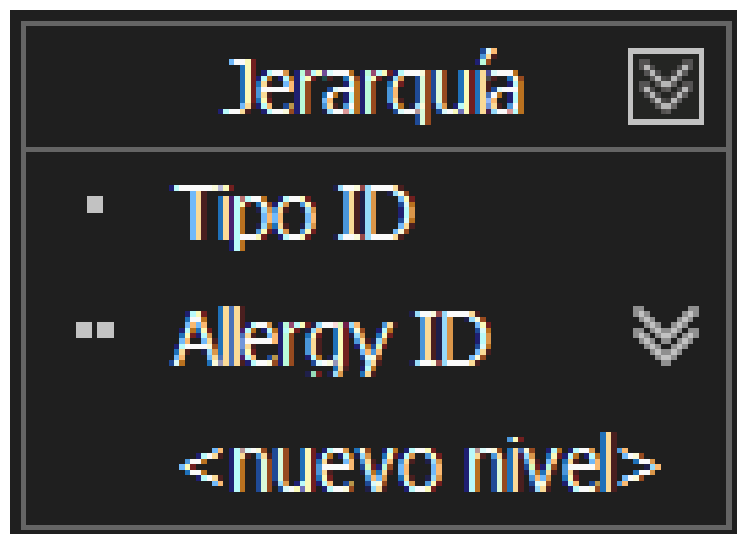
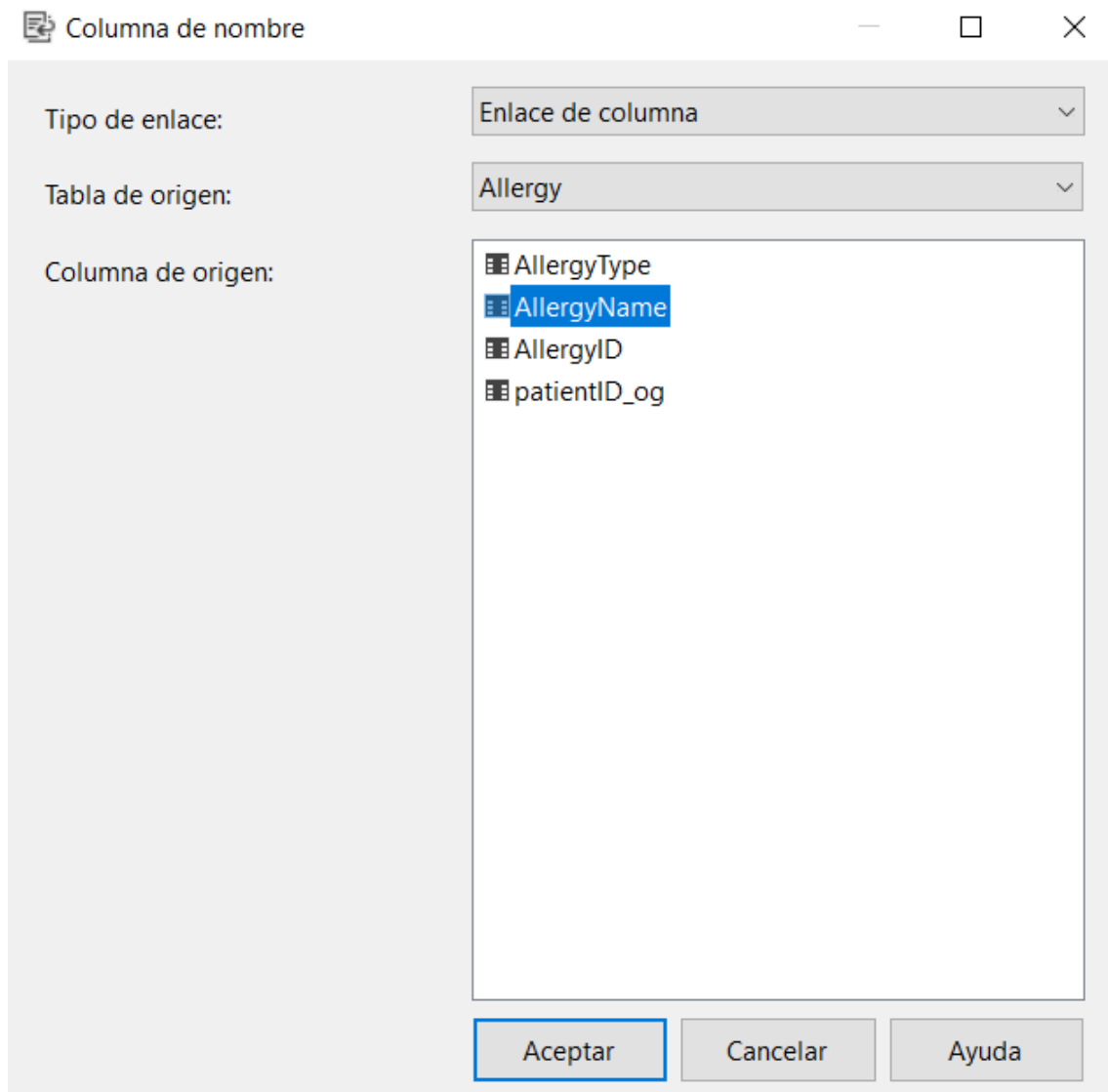


Figure 17: Jerarquía de Alergia

**ValueColumn y NameColumn:**



Columna de nombre

Tipo de enlace: Enlace de columna

Tabla de origen: Allergy

Columna de origen:

- AllergyType
- AllergyName
- AllergyID
- patientID\_og

Aceptar Cancelar Ayuda

Figure 18: allergyID

Seleccionamos AllergyName.

#### 4.3.6 Diagnosis

**Jerarquía:**

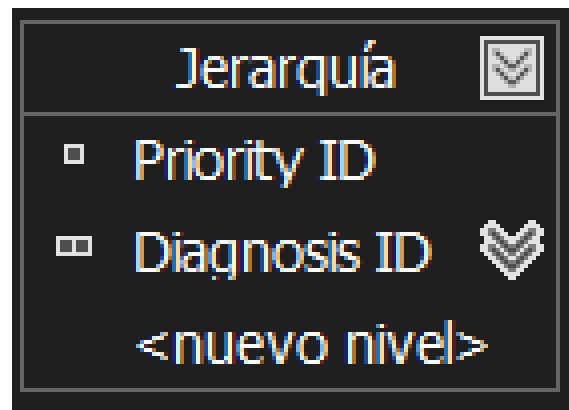


Figure 19: Jerarquía de Diagnosis

**ValueColumn y NameColumn:**

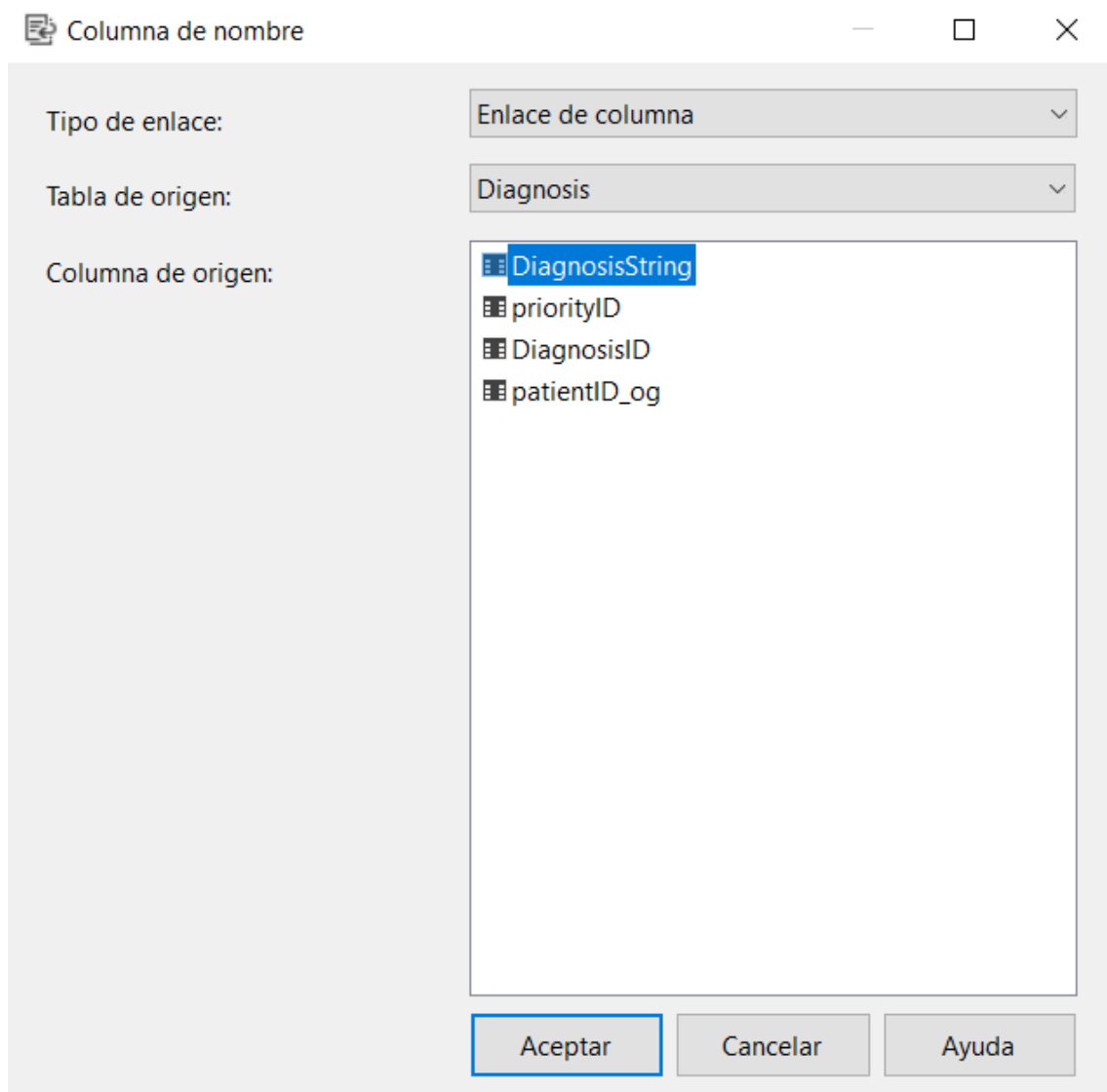


Figure 20: DiagnosisID

Seleccionamos `DiagnosisString`.

#### 4.3.7 Medication

Jerarquía:

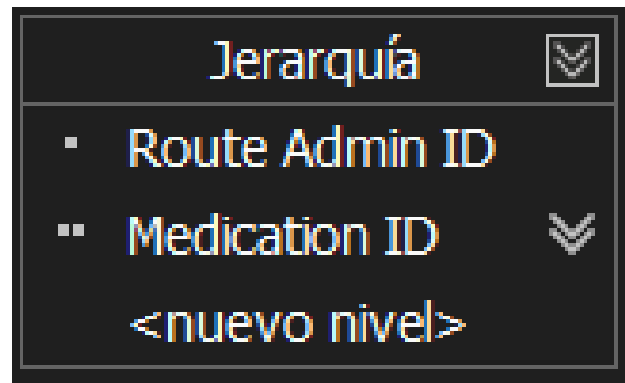


Figure 21: Jerarquía de Hospital

ValueColumn y NameColumn para `RouteAdminID`:

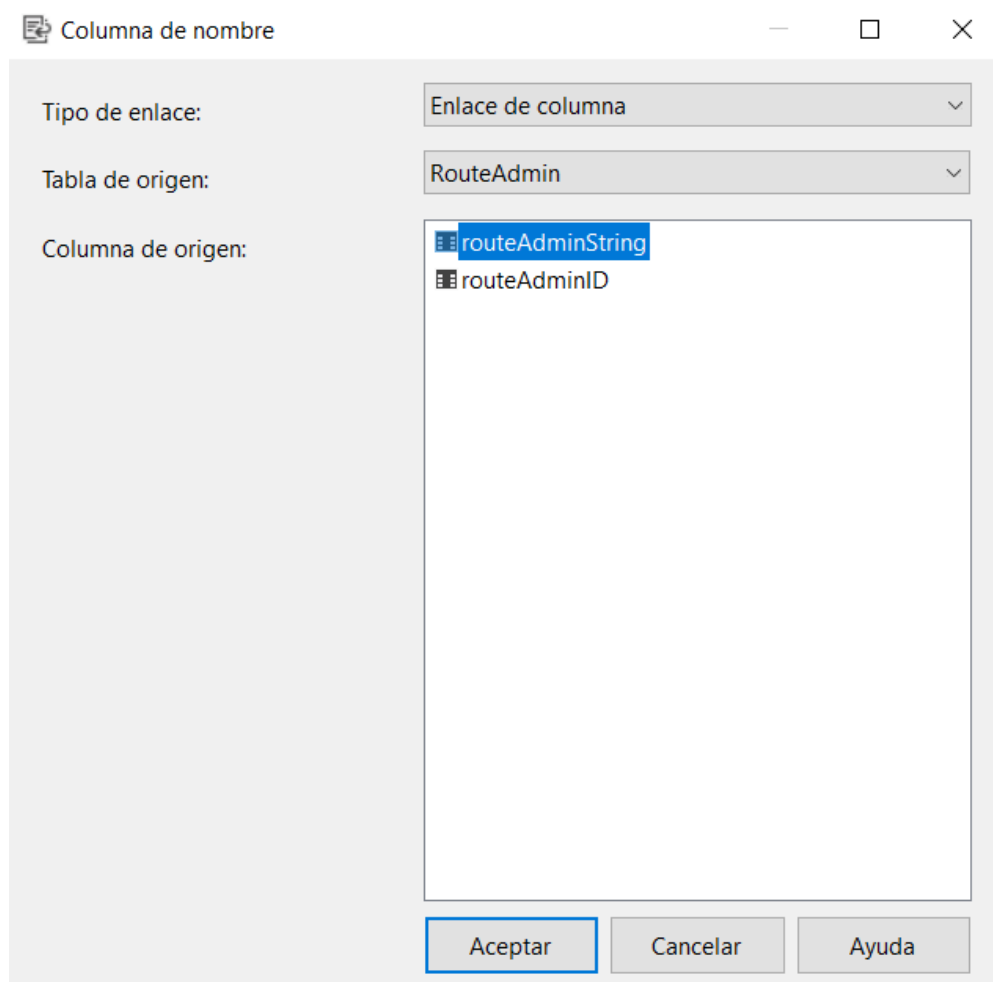


Figure 22: RouteAdminID

Seleccionamos `routeAdminString`.

**ValueColumn** y **NameColumn** para `MedicationID`:

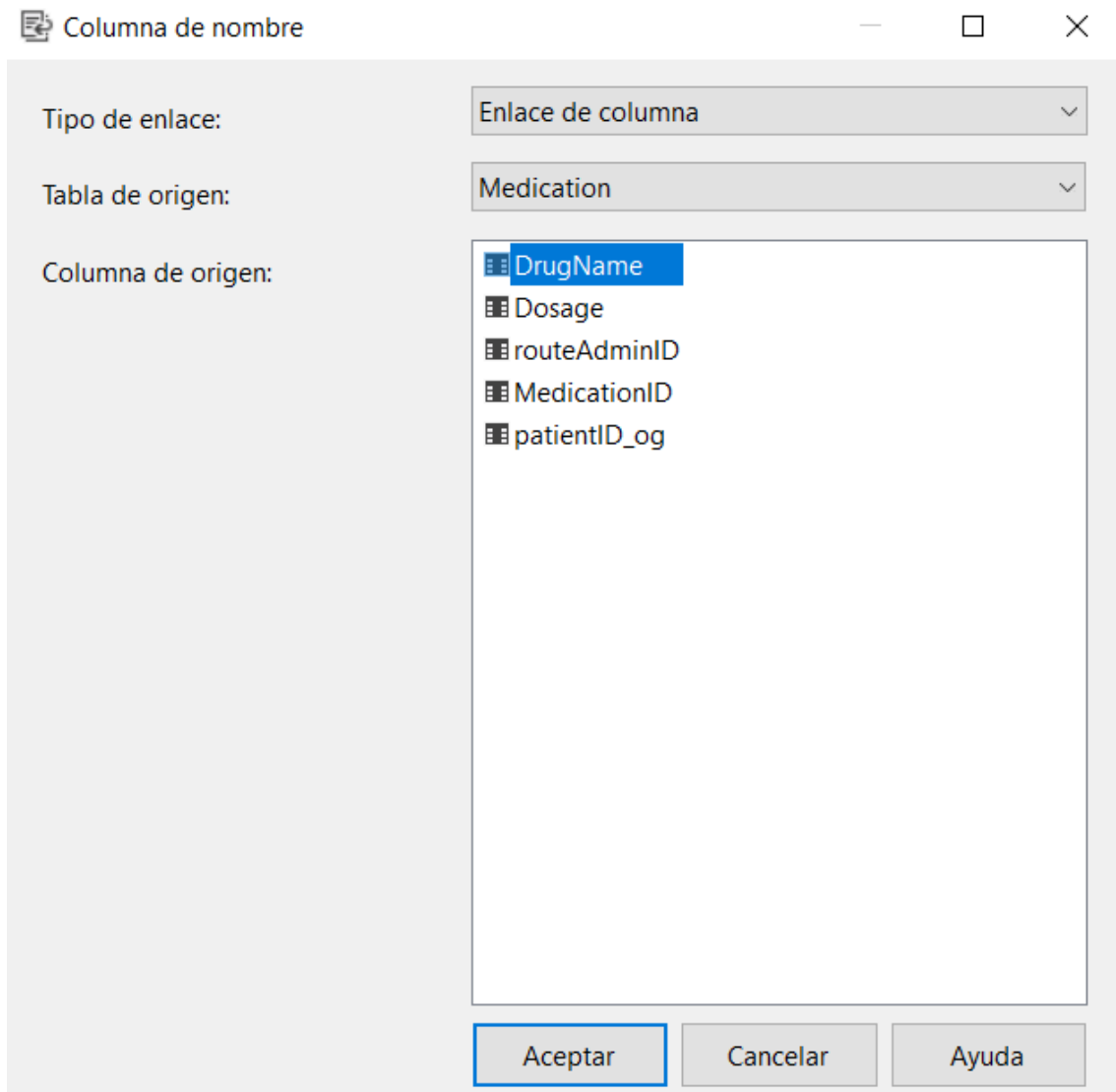


Figure 23: MedicationID

Seleccionamos `DrugName`.

#### 4.3.8 Tiempo

Para la dimensión Tiempo, será necesario primero arrastrar los atributos `HospitalDischargeTime24` y `HospitalDischargeYear` de la vista del origen de datos al apartado de atributos. Luego, arrastraremos primero `Year` y luego `Time24`, para tener dentro de los años las horas de ingreso asociadas.

Quedando algo parecido a la Figura 24:

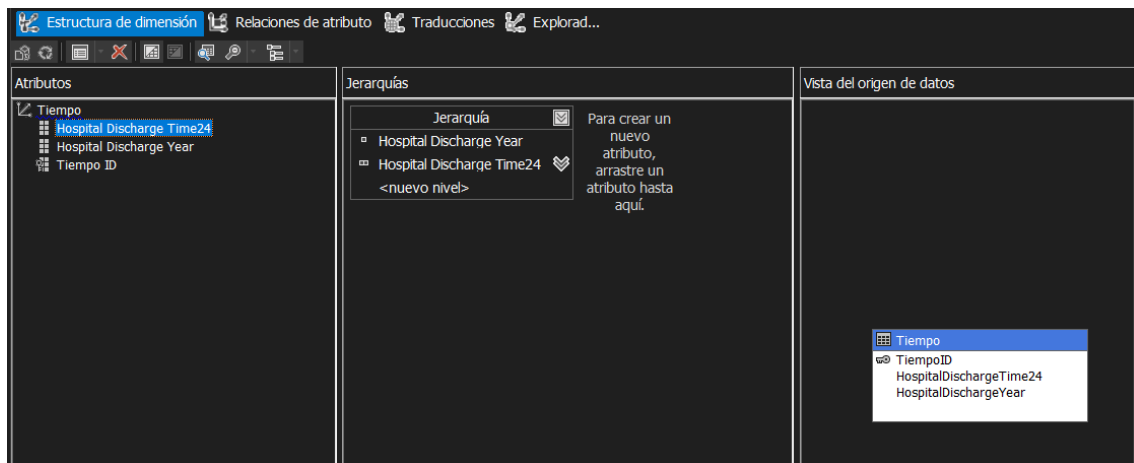


Figure 24: Jerarquía de Tiempo

Posteriormente, para evitar errores de valor duplicado en la hora, haremos lo siguiente:

1. Daremos clic en el atributo `HospitalDischargeTime24`.
2. En el apartado de propiedades, seleccionaremos los tres puntos en **key-Columns**.

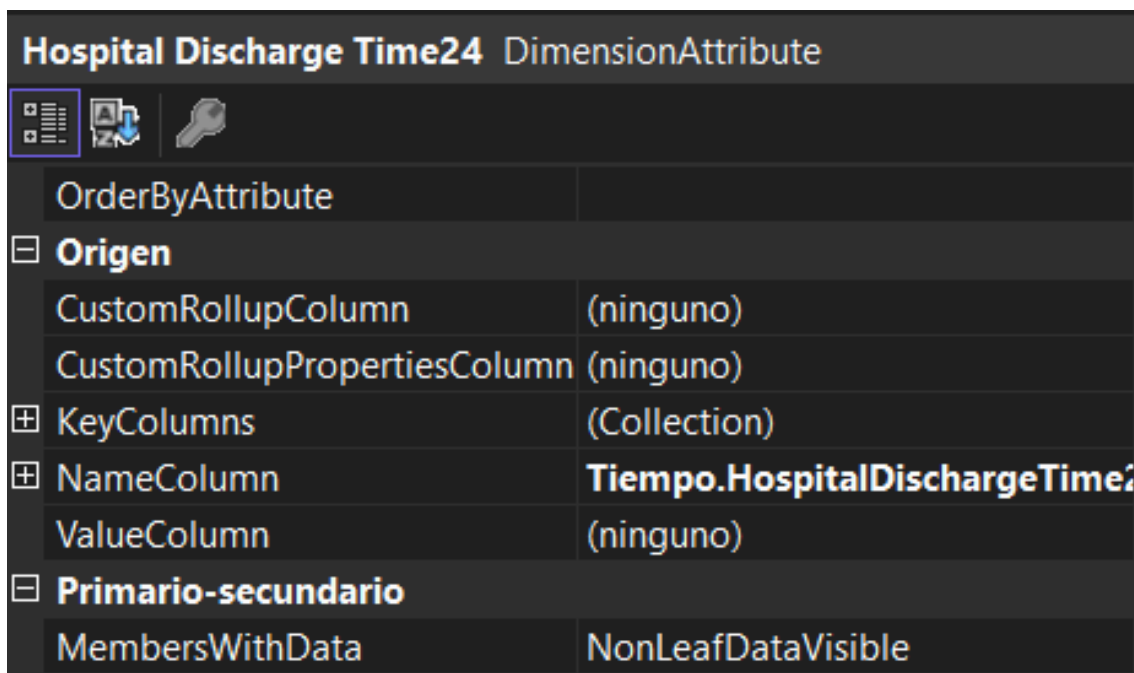


Figure 25: Propiedades de Tiempo

Seleccionaremos las columnas `HospitalDischargeYear` y `HospitalDischargeTime24` para crear juntas una clave única. Con esto, aunque se repitan las horas, estarán vinculadas a un año, impidiendo el error de clave duplicada.

En **NameColumn**, seleccionaremos `HospitalDischargeTime24`. Esto es obligatorio, ya que en **keyColumn** tenemos dos claves y hay que especificar el nombre de la columna sobre la que trabajamos.

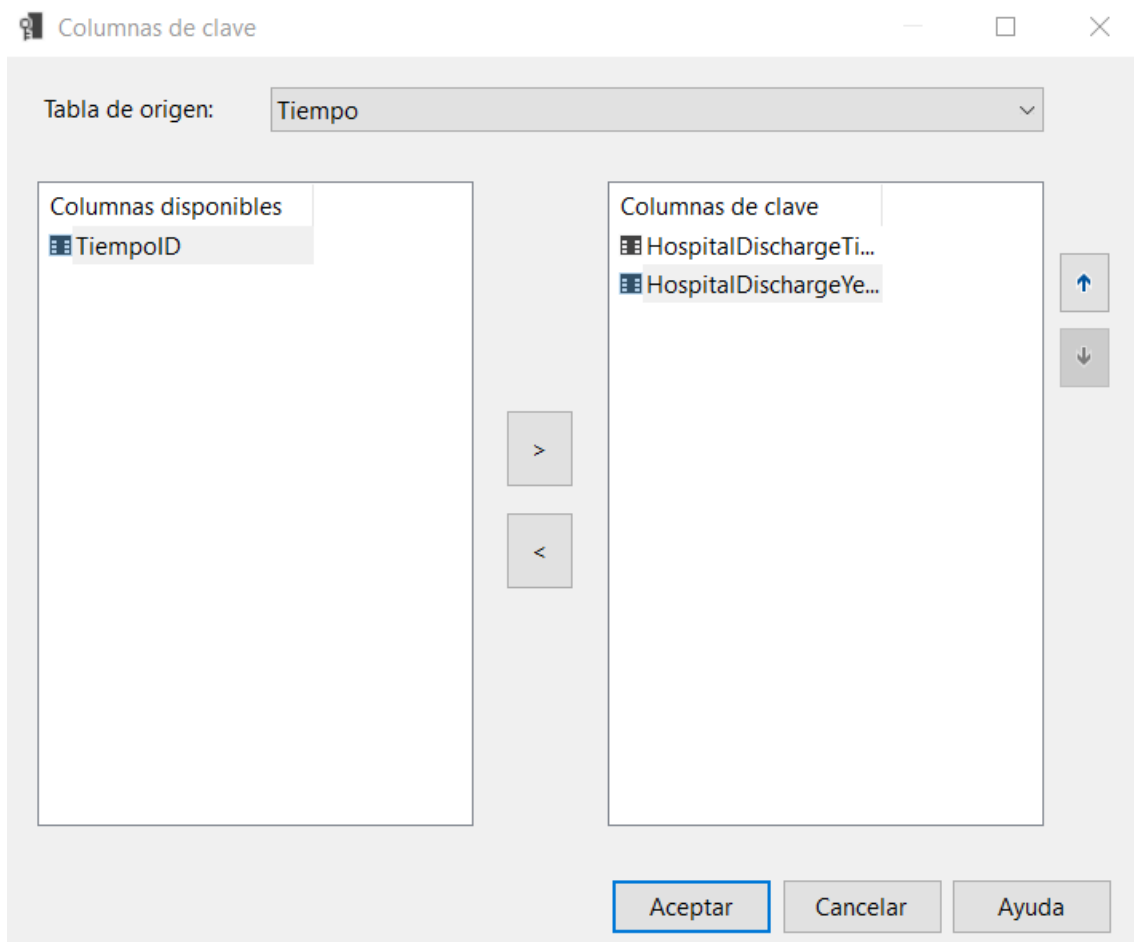


Figure 26: Propiedades de Tiempo

Finalmente, en el apartado de **relaciones de atributo** dentro de la dimensión Tiempo, eliminaremos la relación entre **TiempoID** y **HospitalDischargeYear**, dando clic derecho y seleccionando "Eliminar".

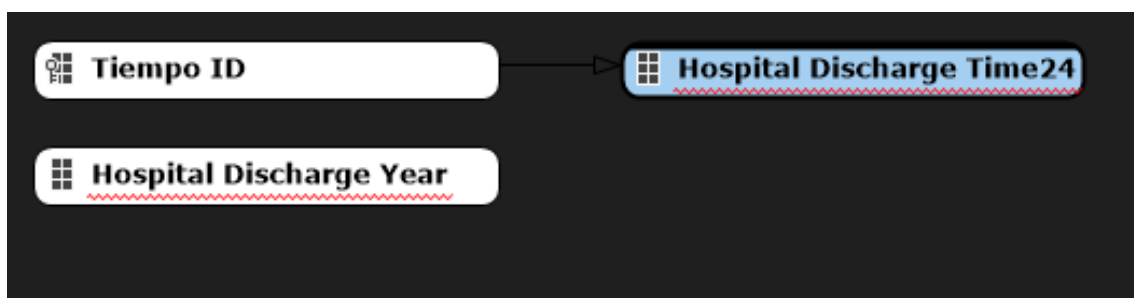


Figure 27: Relaciones de Tiempo Antes

Luego, crearemos una nueva relación de atributo para **HospitalDischargeTime24**. Esto generará una configuración como la Figura 28.

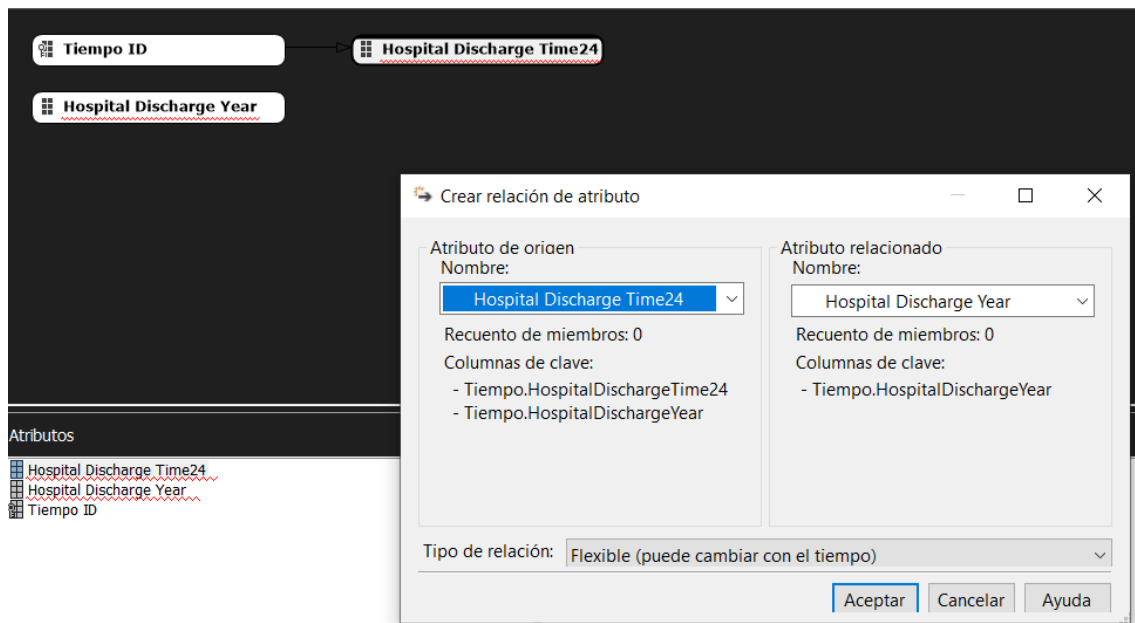


Figure 28: Relaciones de Tiempo Después

Procedemos a darle a aceptar, y con ello habremos construido una jerarquía correctamente definida para Tiempo.

#### 4.4 Modificaciones del cubo

Antes de proceder al despliegue del cubo, será necesario realizar ciertos ajustes en los grupos de medida para garantizar que las tablas de relación NxM funcionen correctamente.

El ajuste principal se ilustra en la Figura 29, en la cual será necesario asociar el hecho **Ingreso UCI** a la tablas las tablas **Allergy**, **Diagnosis** y **Medication** mediante una relación de tipo varios a varios utilizando sus tablas intermedias correspondientes.

Prueba1.dsv [Diseño] PruebaDefini...cubo [Diseño] Paciente 1.dim [Diseño] Hospital.dim [Diseño] Tiempo.dim [Diseño]					
Estructura de cu... Uso de dimensiones Cálculos KPI Acciones Particiones Agregaciones Perspectivas Traducciones Explorad...					
Grupos de medida					
Dimensiones	Ingreso UCI	Admission Dx	Apache Aps Var	Respiratory Care	
Ingreso UCI	Patient Unit Stay ID	Patient Unit Stay ID	Patient Unit Stay ID	Patient Unit Stay ID	Patient Unit Stay ID
Hospital	Hospital ID	Ingreso UCI	Ingreso UCI	Ingreso UCI	Ingreso UCI
Tiempo	Tiempo ID	Ingreso UCI	Ingreso UCI	Ingreso UCI	Ingreso UCI
Admission Dx		Admission Dx ID			
Allergy	UCI Alergia				Allergy ID
Diagnosis	UCI Diagnosis				
Medication	UCI Medicacion				
Paciente	Unique PID	Ingreso UCI	Ingreso UCI	Ingreso UCI	Ingreso UCI

Figure 29: Grupos de Medida



**Allergy**, **Diagnosis** y **Medication** no tienen relaciones con el resto de tablas, por lo que el resto de columnas que no sean IngresoUCI o las tablas intermedias, deberán quedar vacías.

Adicionalmente, será necesario asignar **Allergy ID** como atributo de granularidad entre las tablas **Allergy** y **UCI\_Alergia**. Este mismo procedimiento deberá aplicarse a todas las demás tablas intermedias (UCI Diagnosis y UCI Medication en este ejemplo).

#### 4.4.1 Nuevas medidas

Finalmente, procederemos a la incorporación de nuevas medidas, específicamente **Máximo Age** y **Mínimo Age**, como se muestra en la Figura 30. Para realizar esta operación, basta con hacer clic derecho sobre la dimensión **Ingreso UCI**, seleccionar la opción **Crear nueva medida** y, en la ventana emergente, definir el uso como **Mínimo** o **Máximo**, según corresponda. La tabla origen para ambas medidas será **Ingreso UCI** y la columna origen será **age**.

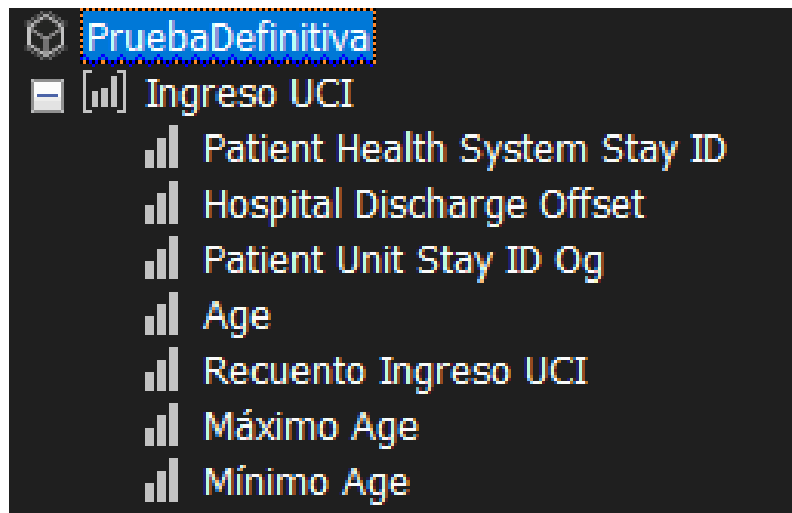


Figure 30: Nuevas medidas

En este apartado se optó por usar el valor edad para probar los distintos usos que se pueden llegar a hacer en las medidas, se invita al lector a usar más de estas pues muchos tutoriales se quedan solamente con el uso de recuento cuando son herramientas que bajo ciertos contextos pueden ser útiles y facilitar tanto la consulta o proporcionar nueva información que de otro modo llega a ser ignorada.

#### 4.4.2 Despliegue del cubo

Una vez realizados todos los ajustes necesarios, el cubo estará listo para ser desplegado. El proceso se inicia pulsando el botón correspondiente, y si todos los pasos han sido ejecutados correctamente, el sistema mostrará un **check verde**, tal como se ilustra en la Figura 31.

Esta medida puede proporcionar ciertos problemas al haber conexiones entre la base de datos y este archivo, un error muy común puede ser que dentro de las propiedades del proyecto en implementación este asignado el servidor localhost, esto se detallará más en el punto 4.5 a continuación.

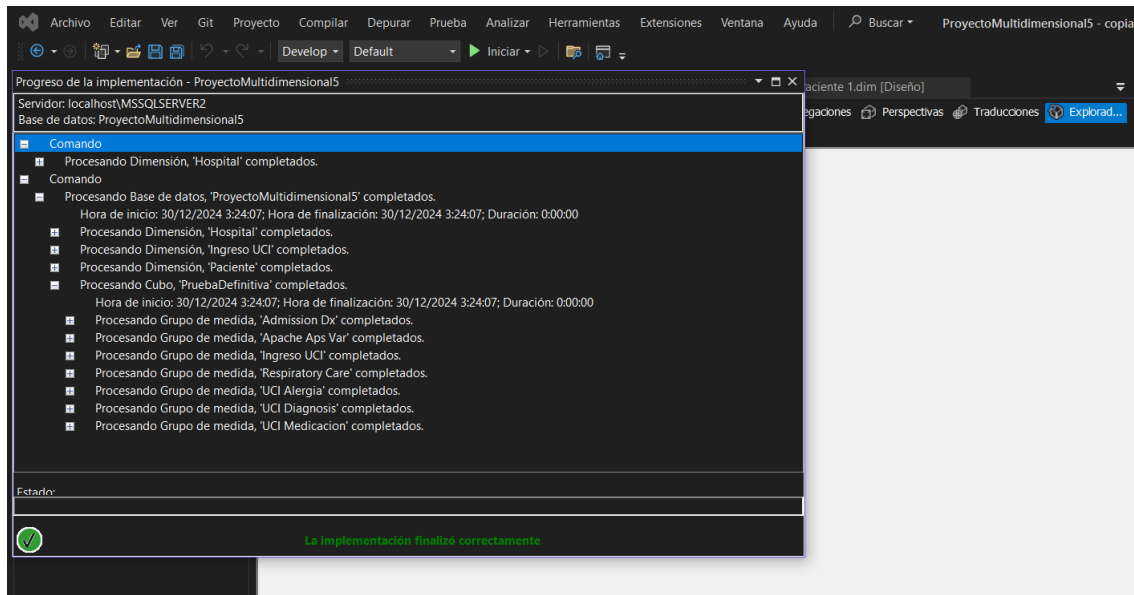


Figure 31: Check Verde

## 4.5 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 este caso 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.

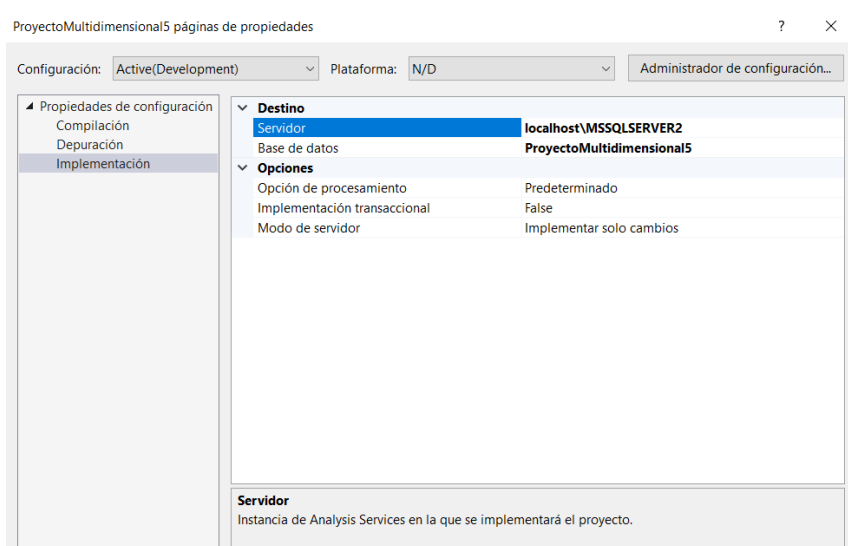


Figure 32: Configuración de despliegue

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

A continuación, se presentan varias consultas realizadas sobre el *cubo multidimensional* que contiene información hospitalaria. Se han desarrollado un **total de 8 consultas**, cuidadosamente seleccionadas para abarcar resultados de interés que respondan a preguntas clave del análisis de los datos. Estas consultas no solo utilizan *funciones básicas de MDX*, sino que también incorporan técnicas avanzadas, como medidas calculadas, filtros y agregaciones. Además, se ha hecho uso de funciones avanzadas de *navegación relativa*, como *PREVMEMBER*, *CURRENTMEMBER* y *PARENT*, que permiten recorrer jerarquías y explorar las relaciones entre diferentes niveles de los datos, logrando una mayor profundidad en el análisis.

Cabe destacar que el cubo cuenta con el nombre *PruebaDefinitiva* por practicidad, al ser un cubo desplegado a través de los datos suministrados en esta tarea comprobando su funcionamiento sin necesidad de contar con las partes anteriores, esto se resaltarán más detenidamente en la siguiente sección.

### 5.1 Consulta 1: Total de pacientes ingresados en hospital del oeste por género y edad

La siguiente consulta calcula el total de pacientes ingresados en hospitales de la región oeste, mostrando además información sobre la edad media, máxima y mínima de los pacientes, agrupados por su género. Se define una medida calculada para obtener la edad media de los pacientes ingresados en la UCI.

```
WITH
MEMBER [Measures].[MediaEdadUCI] AS
[Measures].[AgeTotal] / [Measures].[Recuento Ingreso UCI],
FORMAT_STRING = "##.00"
SELECT
NON EMPTY {[Measures].[Recuento Ingreso UCI], [Measures].[AgeMaximo],
[Measures].[AgeMenor], [Measures].[MediaEdadUCI]} ON COLUMNS,
NON EMPTY [Paciente].[Gender ID].MEMBERS ON ROWS
FROM
PruebaDefinitiva
WHERE
[Hospital].[Region].[Region ID].West
```

En la consulta anterior, se define la medida calculada `[Measures].[MediaEdadUCI]` para obtener el promedio de la edad de los pacientes ingresados en la UCI. Se realiza una agregación de las dimensiones de género, y se filtra por los hospitales ubicados en la región oeste (`[Hospital].[Region].[Region ID].West`). La salida de esta consulta puede apreciarse en la figura 33

	Recuento Ingreso UCI	AgeMaximo	AgeMenor	MediaEdadUCI
All	606	90	19	62.69
Male	438	90	19	60.65
Female	168	90	23	68.02

Figure 33: Resultado de la **consulta 1**: Total de pacientes ingresados en la UCI por género en hospitales de la región oeste.

Como se puede observar en esta región hay 606 ingresos, de los cuales podemos destacar que la mayoría son de personas masculinas, en el proceso de ETL se trato para que la edad fuera un valor *int* y así poder manejarlo como una medida, en este proceso la edad correspondiente a los valores mayor que 89 fue catalogado como 90 por efectos prácticos, teniendo bastante sentido que de esta consulta tan amplia coincidan las edades máximas como mínimas.

## 5.2 Consulta 2: Número de medicamentos usados por cada etnia y región de hospital.

La siguiente consulta calcula el número total de medicamentos administrados a los pacientes ingresados en la UCI, mostrando la distribución por etnia de los pacientes y la región del hospital. Se utiliza una medida calculada para nombrar con más claridad.

```
WITH
MEMBER [Measures].[TotalMedicamentos] AS
[Measures].[Recuento UCI Medicacion],
FORMAT_STRING = "##"
SELECT
NON EMPTY
{[Measures].[TotalMedicamentos]} ON COLUMNS,
NON EMPTY [Hospital].[Region ID].MEMBERS
* [Paciente].[Ethnicity ID].MEMBERS ON ROWS
FROM
[PruebaDefinitiva]
```

En esta consulta se define la medida calculada `[Measures].[TotalMedicamentos]` para obtener el total de medicamentos administrados en la UCI. La consulta genera una tabla cruzada que combina las dimensiones de región del hospital y etnia del paciente, proporcionando un desglose detallado del uso de medicamentos por cada combinación de región y etnia. La figura 34 muestra el resultado de esta consulta.

		Total Medicamentos
Midwest	African American	2637
Northeast	All	7287
Northeast	Other/Unknown	36
Northeast	Caucasian	7118
Northeast	African American	133
South	All	21980
South	Other/Unknown	407
South	Native American	48
South	Asian	671
South	Hispanic	2148
South		726
South	Caucasian	12609
South	African American	5371
West	All	10266

Figure 34: Resultado de la **consulta 2**: Número de medicamentos usados por cada etnia y región de hospital.

Es importante señalar que en esta versión de la consulta, se incluyen los grupos ALL, que representan todos los valores para las dimensiones de región y etnia. En el repositorio de GitHub se puede encontrar una versión alternativa de esta consulta, en la cual se excluyen los grupos ALL, proporcionando una versión donde se omiten las agregaciones globales.

### 5.3 Consulta 3: Promedio de PaO2, FiO2 y Frecuencia Respiratoria por Región Hospitalaria y Etnia para Pacientes de 20 a 30 Años en UCI

La siguiente consulta calcula el promedio de PaO2, FiO2 y frecuencia respiratoria de los pacientes ingresados en la UCI, filtrando por aquellos que tienen entre 20 y 30 años. Los resultados están desglosados por región del hospital y etnia. Además, se define una medida calculada para sumar el total de pacientes considerados.

El código de la consulta 3:

```
WITH
MEMBER [Measures].[Total Pacientes] AS
[Measures].[Recuento Apache Aps Var]
MEMBER [Measures].[MediaPaO2] AS
[Measures].[Pa O2] / [Measures].[Recuento Apache Aps Var],
FORMAT_STRING = "##.00"
MEMBER [Measures].[MediaFi O2] AS
```

```

[Measures].[Fi O2] / [Measures].[Recuento Apache Aps Var],
FORMAT_STRING = "##.00"
MEMBER [Measures].[MediaRespiratoryRate] AS
[Measures].[Respiratory Rate] / [Measures].[Recuento Apache Aps Var],
FORMAT_STRING = "##.00"
SELECT
NON EMPTY
{[Measures].[MediaPaO2],
[Measures].[MediaFi O2],
[Measures].[MediaRespiratoryRate],
[Measures].[Total Pacientes]} ON COLUMNS,
NON EMPTY
CROSSJOIN(
[Hospital].[Region].[Region ID],
[Paciente].[Etnia].[Ethnicity ID]
) ON ROWS
FROM
[PruebaDefinitiva]
WHERE
([Ingreso UCI].[agePatient].[Age].&[20] :
[Ingreso UCI].[agePatient].[Age].&[30])

```

En esta consulta se definen varias medidas calculadas, incluyendo [Measures].[MediaPaO2], [Measures].[MediaFi O2] y [Measures].[MediaRespiratoryRate], que calculan los promedios respectivos de PaO2, FiO2 y la frecuencia respiratoria. Los resultados están agrupados por región del hospital y etnia, y solo se consideran pacientes en el rango de edad de 20 a 30 años (Mediante el where). La figura 35 muestra los resultados obtenidos.

		MediaPaO2	MediaFi O2	MediaRespiratoryRate	Total Pacientes
	Native American	100.00	45.00	7.00	1
	Caucasian	86.29	24.29	17.29	7
	African American	-1.00	-1.00	6.00	1
Midwest	Other/Unknown	98.50	29.50	37.00	2
Midwest	Hispanic	-1.00	-1.00	32.00	1
Midwest	Caucasian	2.39	.13	29.78	23
Northeast	Caucasian	-1.00	-1.00	31.50	6
South	Other/Unknown	48.35	10.00	22.00	2
South	Hispanic	-1.00	-1.00	16.33	3
South		-1.00	-1.00	21.00	1
South	Caucasian	24.64	8.54	18.75	28
South	African American	-1.00	-1.00	19.73	22
West	Other/Unknown	-1.00	-1.00	11.00	1
West	Hispanic	-1.00	-1.00	25.00	4
West	Caucasian	10.20	2.07	18.40	30
West	African American	-1.00	-1.00	11.00	1

Figure 35: Resultado de la **consulta 3**: Promedio de PaO2, FiO2 y frecuencia respiratoria por región hospitalaria y etnia para pacientes de 20 a 30 años en UCI.

Es importante destacar que en la base de datos existen valores de -1 que podrían indicar la ausencia de información sobre PaO2. Sin embargo, al haber valores nulos en algunas tuplas, se decidió mantener la base de datos lo más fiel posible al sistema original, ElCU, considerando que algunos médicos podrían haber registrado estos valores intencionalmente.

Además, se ha observado que en algunas regiones, todos los pacientes de origen hispano y afroamericano tienen valores de PaO2 y FiO2 de -1, lo que no ocurre con los pacientes de origen nativo americano y la mayoría de los pacientes caucásicos. Esto podría reflejar diferencias en la calidad o disponibilidad de los registros médicos entre distintos grupos étnicos.

## 5.4 Consulta 4: Cálculo de Porcentaje Acumulado de Ingresos UCI por Región Hospitalaria

La siguiente consulta calcula el porcentaje acumulado de ingresos en UCI por cada región hospitalaria, excluyendo la etiqueta "Unknown" de las regiones que no contienen valores de ingresos (Esto se logra mediante Filter). Se utilizan medidas calculadas para sumar los ingresos acumulados por región y el total global de ingresos.

```
WITH
MEMBER [Measures].[Total Acumulado] AS
SUM(
{NULL:[Hospital].[Region ID].CURRENTMEMBER},
[Measures].[Recuento Ingreso UCI]
)
MEMBER [Measures].[Total Global] AS
SUM(
[Hospital].[Region ID].[All],
[Measures].[Recuento Ingreso UCI]
)
MEMBER [Measures].[Porcentaje Acumulado] AS
[Measures].[Total Acumulado] / [Measures].[Total Global],
FORMAT_STRING = "0.00"
SELECT
NON EMPTY {[Measures].[Recuento Ingreso UCI],
[Measures].[Total Acumulado],
[Measures].[Porcentaje Acumulado]} ON COLUMNS,
FILTER(
[Hospital].[Region ID].CHILDREN,
[Hospital].[Region ID].CURRENTMEMBER.NAME <> "Unknown" AND
NOT ISNULL([Measures].[Recuento Ingreso UCI])
) ON ROWS
FROM [PruebaDefinitiva]
```

En esta consulta se definen varias medidas calculadas, incluyendo [Measures].[Total Acumulado] para sumar los ingresos en UCI por región hospitalaria, y [Measures].[Porcentaje Acumulado], que calcula el porcentaje acumulado respecto al total global de ingresos en todas las regiones. Además, como se comentó se usó un filtro para excluir la

región "Unknown" y aquellas filas donde no existen ingresos registrados. La figura 36 muestra el resultado de la consulta.

	Recuento Ingreso UCI	Total Acumulado	Porcentaje Acumulado
	210	210	0.08
Midwest	807	1017	0.40
Northeast	159	1176	0.47
South	738	1914	0.76
West	606	2520	1.00

Figure 36: Resultado de la consulta 4: Cálculo de porcentaje acumulado de ingresos UCI por región hospitalaria (sin Unknown).

Como se puede observar estamos tratando con los 2520 ingresos de la base de datos ElCU, se puede ver como se va pasando desde aquellos hospitales que no tienen una región asignada, a *Midwest*, *Northeast*, *South* y finalmente *West*. Hasta conseguir el 1 que representa el 100% de los ingresos.

## 5.5 Consulta 5: Recuento de Ingresos a UCI por Hora y Total Anual

La siguiente consulta calcula el recuento de ingresos a la UCI por hora y el total anual utilizando navegación relativa dentro de la dimensión temporal. Se utiliza la función `.PARENT` para obtener los ingresos del nivel superior (año) desde el nivel más granular (hora de alta hospitalaria).

```
WITH
MEMBER [Measures].[Recuento Anual] AS
([Tiempo].[Tiempo].CURRENTMEMBER.PARENT,
[Measures].[Recuento Ingreso UCI])
SELECT
{[Measures].[Recuento Ingreso UCI],
[Measures].[Recuento Anual]} ON COLUMNS,
[Tiempo].[Tiempo].[Hospital Discharge Time24].MEMBERS ON ROWS
FROM [PruebaDefinitiva]
```

En esta consulta, la medida calculada `[Measures].[Recuento Anual]` emplea la función `.PARENT` para navegar desde el nivel de la hora de alta hospitalaria al nivel padre correspondiente, que en este caso es el año. La consulta proporciona los ingresos en UCI por hora junto con el total anual de ingresos. La figura 37 ilustra el resultado de esta consulta.



	Recuento Ingreso UCI	Recuento Anual
23:39:00	1	1204
23:40:00	4	1204
23:43:00	2	1204
23:44:00	2	1204
23:45:00	7	1204
23:47:00	1	1204
23:48:00	1	1204
23:50:00	5	1204
23:53:00	1	1204
23:56:00	3	1204
23:58:00	2	1204
00:00:00	4	1316
00:02:00	3	1316
00:03:00	1	1316
00:05:00	1	1316

Figure 37: Resultado de la **consulta 5**: Recuento de ingresos a UCI por hora y total anual.

El uso de la función `.PARENT` permite obtener fácilmente las medidas de un nivel superior de la jerarquía temporal, lo que es útil para calcular el total anual de ingresos en UCI mientras se observa el comportamiento a nivel de hora, esta función tiene más sentido en aquellos casos donde existen muchos niveles de granularidad como los vistos en clase.

## 5.6 Consulta 6: Recuento de Diagnósticos por Edad y prioridad en un Intervalo de Tiempo Específico (00:00:00 a 12:00:00 del 2014)

Esta consulta calcula el recuento de diagnósticos de la UCI, es decir la cantidad de diagnósticos en un intervalo de tiempo específico, entre las 00:00:00 y las 12:00:00 para el año 2014. Utiliza las dimensiones de edad y diagnóstico para obtener un desglose detallado por grupo de edad y prioridad de diagnóstico.

```
SELECT
{ [Measures].[Recuento UCI Diagnosis] } ON COLUMNS,
NON EMPTY
CROSSJOIN(
```

```

[Ingreso UCI].[Age].MEMBERS,
[Diagnosis].[PriorityDiag].[Priority ID].MEMBERS
) ON ROWS
FROM
[PruebaDefinitiva]
WHERE
([Tiempo].[Tiempo].[Hospital Discharge Time24].&[00:00:00]&[2014] :
[Tiempo].[Tiempo].[Hospital Discharge Time24].&[11:40:00]&[2014])

```

En esta consulta, se realiza un cruce entre las dimensiones de edad de los pacientes y las prioridades de diagnóstico, y se calcula el número de diagnósticos registrados en el intervalo de tiempo especificado. La figura 38 muestra el resultado de esta consulta.

		Recuento UCI Diagnosis
All	Major	1551
All	Primary	602
All	Other	1042
16	Primary	2
16	Other	1
19	Primary	4
19	Other	6
21	Major	10
21	Primary	4
21	Other	7
23	Major	6
23	Primary	5
23	Other	14

Figure 38: Resultado de la **consulta 6**: Recuento de diagnósticos por edad y diagnóstico en un intervalo de tiempo específico (00:00:00 a 12:00:00 del 2014).

Es importante destacar que aunque parezcan muchos diagnósticos, se están buscando todos los diagnósticos realizados desde las 00:00 hasta las 12:00 en todo el año 2014, dando 3195 de los 24978 diagnósticos de la base de datos EICU, donde hay ingresos sin diagnósticos como ingresos con más de un diagnóstico.

## 5.7 Consulta 7: Cantidad de alergias, para los pacientes ingresados y Tasa Respiratoria Media por Edad, Tipo de Alergia y Diagnóstico para Pacientes Hispanos en 2015

Esta consulta calcula alergias de los pacientes ingresados junto a la tasa respiratoria media para pacientes hispanos en el año 2015, desglosado por edad, tipo de alergia y diagnóstico. Se utiliza una medida calculada para obtener la tasa respiratoria media, dividiendo la tasa respiratoria por el recuento de ingresos a la UCI.

```
WITH
MEMBER [Measures].[MediaRespiratoryRate] AS
[Measures].[Respiratory Rate] / [Measures].[Recuento Ingreso UCI],
FORMAT_STRING = "##.00"
SELECT
{
    [Measures].[Recuento UCI Alergia],
    [Measures].[Recuento Ingreso UCI],
    [Measures].[MediaRespiratoryRate]
} ON COLUMNS,

NON EMPTY
CROSSJOIN(
CROSSJOIN(
[Ingreso UCI].[agePatient].[Age].MEMBERS,
[Allergy].[TipoAlergia].[Tipo ID].MEMBERS
),
[Diagnosis].[PriorityDiag].[Priority ID].MEMBERS
) ON ROWS

FROM
[PruebaDefinitiva]

WHERE
-- Se filtra por el año 2015 y pacientes hispanos
([Tiempo].[Tiempo].[Hospital Discharge Year].[2015],
[Paciente].[Etnia].[Ethnicity ID].Hispanic)
```

En esta consulta se emplean los elementos vistos anteriormente intentando obtener mayor provecho de la multidimensionalidad de los cubos, accediendo a la mayoría de tablas en una sola consulta relativamente sencilla. La figura 39 muestra el resultado de esta consulta.

			Recuento UCI Alergia	Recuento Ingreso UCI	MediaRespiratoryRate
35	Drug	Major	3	1	39.00
35	Drug	Primary	3	1	39.00
35	Drug	Other	3	1	39.00
40	Drug	Major	1	1	10.00
40	Drug	Primary	1	1	10.00
40	Drug	Other	1	1	10.00
40	Non Drug	Major	3	1	10.00
40	Non Drug	Primary	3	1	10.00
40	Non Drug	Other	3	1	10.00
41	Drug	Major	1	1	44.00
41	Drug	Primary	1	1	44.00
41	Drug	Other	1	1	44.00
43	Drug	Major	9	1	31.00
43	Drug	Primary	9	1	31.00
43	Drug	Other	9	1	31.00
63	Drug	Major	1	1	12.00
63	Drug	Primary	1	1	12.00
63	Drug	Other	1	1	12.00
82	Drug	Major	5	1	28.00

Figure 39: Resultado de la **consulta 7**: Recuento UCI y tasa respiratoria media por edad, tipo de alergia y diagnóstico para pacientes hispanos en 2015.

La combinación de las dimensiones Edad, Tipo de Alergia y Diagnóstico permite un análisis exhaustivo de los ingresos.

## 5.8 Consulta 8: Análisis de Medicación y Parámetros Respiratorios por Género y Año, Excluyendo Pacientes de Etnia Hispánica y con Ventilación Distinta de Cero

Esta consulta realiza un análisis de medicación y parámetros respiratorios, como la media de PaO2 y FiO2, por género y año, excluyendo a los pacientes de etnia hispanica y aquellos con ventilación distinta de cero. Además, se usa la mediaPaO2 y MediaFiO2.

```

WITH
MEMBER [Measures].[MediaPaO2] AS
[Measures].[Pa O2] / [Measures].[Measures].[Recuento Ingreso UCI],
FORMAT_STRING = "##.00"

MEMBER [Measures].[MediaFiO2] AS
[Measures].[Fi O2] / [Measures].[Measures].[Recuento Ingreso UCI],
FORMAT_STRING = "##.00"

SELECT
{
[Measures].[Recuento UCI Medicacion],
[Measures].[MediaPaO2],
[Measures].[MediaFiO2]
} ON COLUMNS,

```

```

NON EMPTY
FILTER(
CROSSJOIN(
[Paciente].[Gender ID].MEMBERS,
[Tiempo].[Tiempo].[Hospital Discharge Year].MEMBERS
),
[Measures].[Vent] <> 0
) ON ROWS
FROM
[PruebaDefinitiva]

WHERE
EXCEPT(
[Paciente].[Etnia].[Ethnicity ID].MEMBERS,
{[Paciente].[Etnia].[Ethnicity ID].[Hispanic]}
)

```

Lo más destacable en esta función es el uso de EXCEPT que sirve para excluir a los hispanos de la búsqueda, el resto de herramientas reutilizan los visto en las demás consultas o en clase.

La figura 40 muestra el resultado de esta consulta.

		Recuento UCI Medicacion	MediaPaO2	MediaFiO2
All	2014	34486	21.87	9.63
All	2015	37834	16.99	7.29
Male	2014	19415	23.75	10.58
Male	2015	19722	18.28	8.29
Female	2014	15048	18.95	8.14
Female	2015	18107	15.20	5.90

Figure 40: Resultado de la **consulta 8**: Análisis de medicación y parámetros respiratorios por género y año, excluyendo pacientes de etnia hispanica y con ventilación distinta de cero.

Estas consultas resultan muy útiles en el ámbito médico, particularmente en el análisis de pacientes en unidades de cuidados intensivos (UCI) con patologías respiratorias. Ofrecen información detallada sobre parámetros clínicos como ingresos a la UCI, tasas respiratorias, medicación administrada y características demográficas, lo que permite identificar patrones y realizar comparaciones entre grupos de pacientes mediante filtros y cálculos específicos, como el porcentaje acumulado de ingresos por región o el análisis de parámetros respiratorios por género.

No obstante, una posible mejora sería la incorporación de datos más específicos sobre las patologías respiratorias. A pesar de que se ha realizado una transformación ETL de los datos, los valores extremos de las tasas respiratorias y de frecuencia cardíaca en la base EICU podrían no reflejar adecuadamente las condiciones

reales de los pacientes, ya que es poco probable que tantos pacientes presenten una tasa respiratoria de alrededor de 60 en situaciones de hiperventilación. Contrastar estos datos con fuentes adicionales podría proporcionar un análisis más preciso y representativo.

## 6 Instrucciones para ejecutar las consultas MDX

Para ejecutar las consultas MDX, asegúrese de que el fichero `.bak` con el almacén de datos (resultado de la tarea de ETL) y el proyecto de Analysis Services, que genera el cubo multidimensional, hayan sido desplegados correctamente siguiendo las instrucciones de los apartados anteriores.

### 6.1 Acceso al servidor de Analysis Services

Abra SQL Server Management Studio y conéctese nuevamente a la base de datos. Esta vez, seleccione el tipo de servidor como *Analysis Services*. Ingrese sus credenciales como lo hizo anteriormente. Una vez conectado, expanda las bases de datos disponibles y localice la base de datos llamada `ProyectoMultidimensional5`.

### 6.2 Ejecución de consultas MDX

Para ejecutar las consultas use el botón derecho sobre el proyecto, seleccione la opción *New Query* y elija *MDX*. Esto le permitirá acceder al cubo `PruebaDefinitiva`. En este entorno, similar al de consultas SQL, podrá escribir y ejecutar consultas en lenguaje MDX.

### 6.3 Recomendaciones para la ejecución de consultas

Debido a la naturaleza del entorno de ejecución, solo es posible realizar una consulta por vez. Por lo tanto, se recomienda ejecutar cada consulta por separado. Si desea ejecutar varias consultas de manera consecutiva, puede cargar el archivo que contenga todas las consultas MDX. Asegúrese de seleccionar el código correspondiente a la consulta que desea ejecutar y luego presione el botón *Execute* (puede ver el ejemplo en la figura 41).

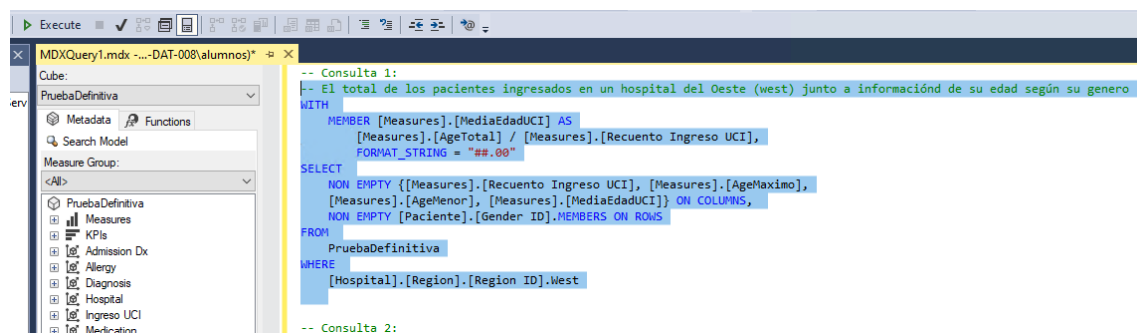


Figure 41: Captura de pantalla al seleccionar una consulta junto al botón *Execute* arriba a la izquierda

## 7 Problemas encontrados

Entre los principales problemas que encontramos, destaca la conexión con la base de datos, la cual generaba errores de manera constante debido a que el método de autenticación no estaba configurado para utilizar *SQL Authentication*.

Adicionalmente, surgieron problemas derivados de un proceso de ETL incorrectamente implementado, lo que resultó en la presencia de múltiples tuplas con valores duplicados, como ocurrió en la tabla *ingresoUCI* inicialmente.

Aunque no se considera un problema en sentido estricto, decidimos modificar el atributo *age*, cambiando su tipo de dato a *int* para facilitar su utilización en las consultas, dado que este atributo es de gran utilidad.

La protección de datos también llega a ser un obstáculo en la comprensión y corroboración de que todo esté en orden, ya que, al contar solamente con identificadores autogenerados en la granularidad más baja, muchas veces no se podía comprobar que todo estuviera en orden a simple vista.

Finalmente, destaca lo caótico que puede llegar a ser entender los errores de los programas usados, ya que contienen una gran cantidad de texto y no queda un mensaje claro sobre cuál puede ser el error. Además, se presentaron problemas con Visual Studio, que recomendaba desinstalar la extensión en casos de errores que bloqueaban la aplicación. Esto, sumado al hecho de que, en las máquinas virtuales, el servidor proxy rechazaba las conexiones, incrementó el tiempo requerido, ya que tuvimos que descargar y acceder a internet desde nuestros propios equipos.

## 8 Conclusiones

En esta fase del proyecto, se logró con éxito la creación de un cubo multidimensional adaptado para el análisis de pacientes con patologías respiratorias. Este cubo permite realizar consultas avanzadas que facilitan la extracción de información relevante para el ámbito hospitalario, mejorando así el análisis de datos clínicos complejos de manera eficiente.

A lo largo del desarrollo, se enfrentaron varios desafíos, entre ellos, se experimentó lo importante que es el proceso ETL, y como es un proceso vital que será muy útil considerar una segunda corrección una vez se empieza a desplegar el cubo y se llega a observar mejoras y necesidades en nuestro almacén, como fue el caso de la transformación del atributo *Age* y la eliminación de registros duplicados. Estos ajustes permitieron desarrollar nuevas consultas en *MDX* que no fueron tomadas en cuenta desde un inicio.

También destacamos como una base de datos transaccional y un almacén de datos se diferencian principalmente en su propósito y estructura. Mientras que las bases de datos están diseñadas para gestionar las operaciones diarias y registrar datos de forma eficiente, un almacén de datos está optimizado para el análisis histórico y la toma de decisiones estratégicas. Un cubo multidimensional, como el implementado en este proyecto, permite la organización de datos en múltiples dimensiones, facilitando el análisis desde diferentes perspectivas, lo cual sería complejo con estructuras relacionales tradicionales. Las consultas *MDX* ofrecen ventajas significativas frente a *SQL*, al estar diseñadas específicamente para navegar y extraer información de cubos a través de varias dimensiones, permitiendo cálculos complejos y agregaciones dinámicas, algo que *SQL* no puede realizar de manera eficiente ni intuitiva en este

tipo de estructuras, en este ejemplo de pequeña escala se puede llegar a ver, pero en caso de haber usado todo EICU, el uso del cubo multidimensional sería aún más importante.

Finalmente, este trabajo sienta lo aprendido y abre nuevos horizontes en cuanto a desafíos a la hora de implementar un almacén y como superarlos.

## 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