


PASO 1: DESCARGAR

Visual Studio | Marketplace

Visual Studio > Tools > SQL Server Integration Services Projects



SQL Server Integration Services Projects

Microsoft | 890,290 installs | 4.5 (127) | Free

This project may be used for building high performance data integration and workflow solutions, including extraction, transformation, and loading (ETL) operations for data warehousing.

Download

Overview | Q & A | Rating & Review

This project may be used for building high performance data integration and workflow solutions, including extraction, transformation, and loading (ETL) operations for data warehousing. Currently the supported target server version starts from SQL Server 2012 up to 2019.

Visit <https://techcommunity.microsoft.com/t5/SQL-Server-Integration-Services/bg-p/SSIS> for the latest information, tips, news, and announcements about SSIS directly from the product team.

IMPORTANT: since version 3.3, Power Query Source for SQL Server 2017 and Microsoft Oracle Connector for SQL Server 2019 have been excluded from the installation of this product. To continue using these two components, please manually download and install them by yourselves. Here are the download links: [Power Query Source for SQL Server 2017 and 2019](#), [Microsoft Oracle Connector for SQL Server 2019](#)

Common Issues

Categories

Tools | Data | Modeling | Other

Tags

ADF | Azure | Azure-SSIS | Data Factory | ETL | Integration Runtime | Integration Services | IR | SQL Server | SSOT | SSOTIS | SSIS

Works with

Visual Studio 2019

Resources

Learn SSIS

PASO 2: INSTALAR

Seleccione el idioma del instalador:

English (United States)

Aceptar | Cancelar

VERSION 3.14

SQL Server Integration Services Projects

Welcome. Click "Next" to begin.

By clicking the "Next" button, I acknowledge that I accept the [License Terms](#) and [Privacy Statement](#).

This product will transmit information about your installation experience, as well as other usage and performance data, to Microsoft to help improve the product. To learn more about data processing and privacy controls, and to turn off the collection of this information after installation, see the [documentation](#).

Next | Close

VERSION 3.14

SQL Server Integration Services Projects

Install this product to the selected Visual Studio instance(s):

☒ Visual Studio Community 2019

Install | Close

VERSION 3.14

SQL Server Integration Services Projects

Loading packages. Please wait...

Cancel

VERSION 3.14

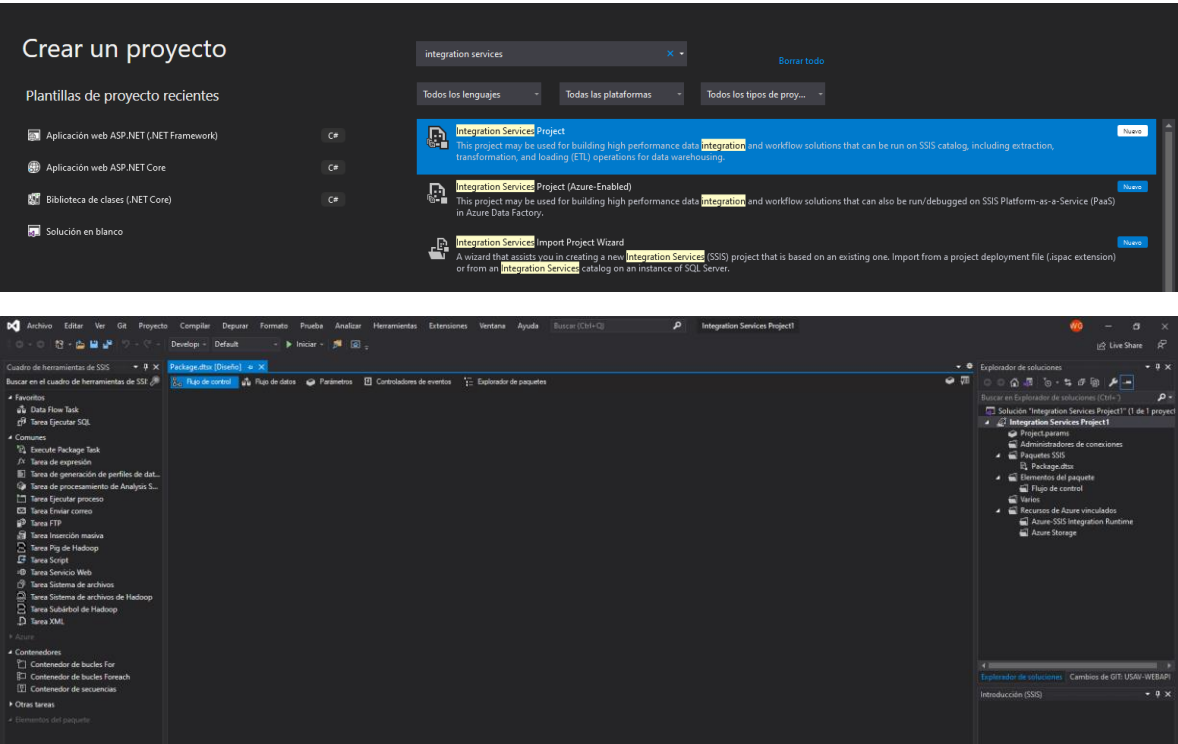
SQL Server Integration Services Projects

Setup Completed

All specified components have been installed successfully.

Close

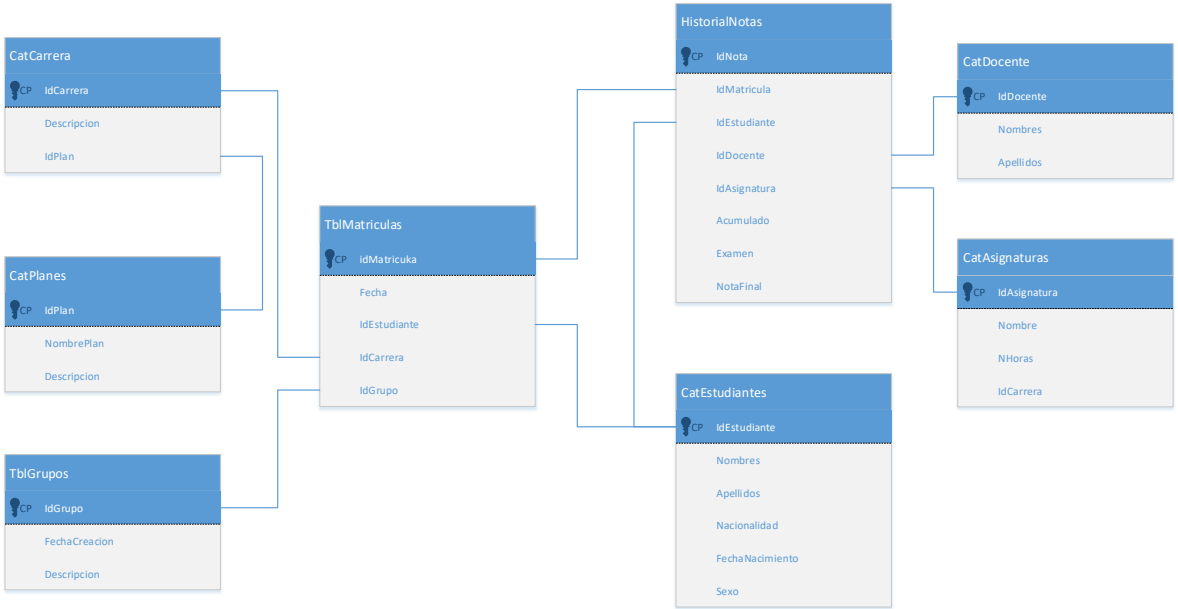
PASO 3: CREAR PROYECTO



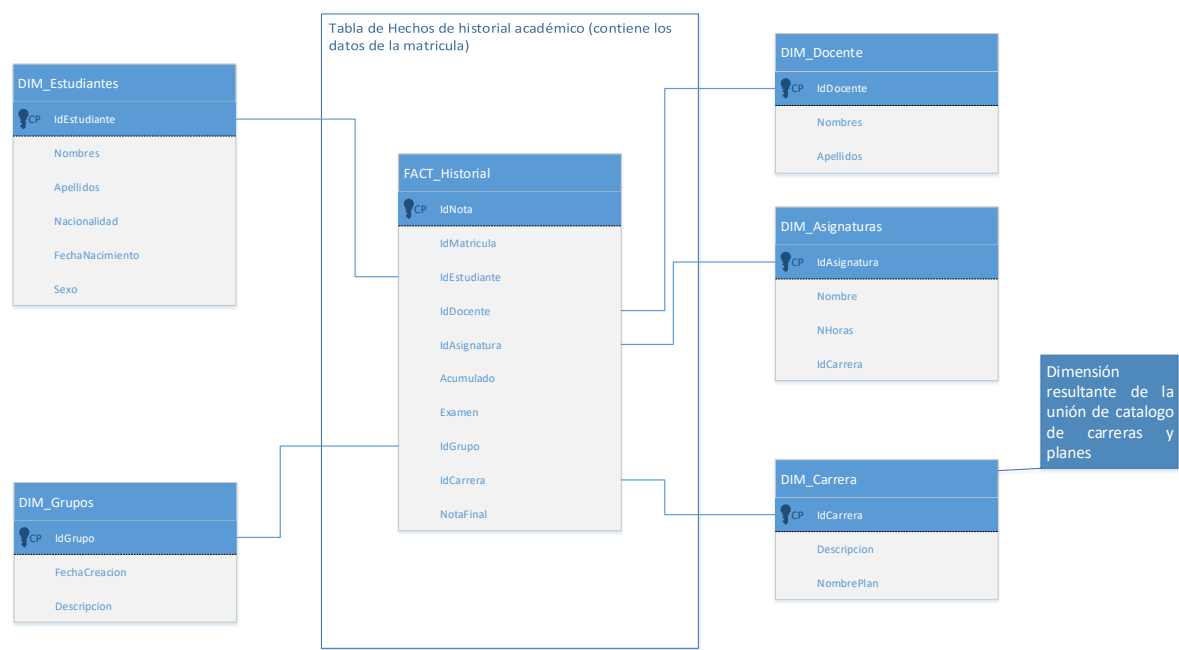
PASO 4: DEFINIR UN FLUJO DE TRABAJO DE DATOS

Previamente se debe tener construido el modelo de datos

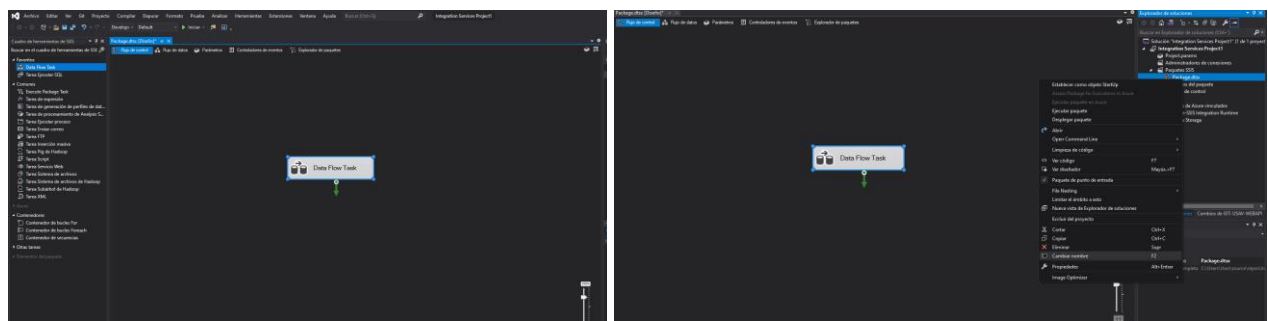
Fuente de datos relacional:



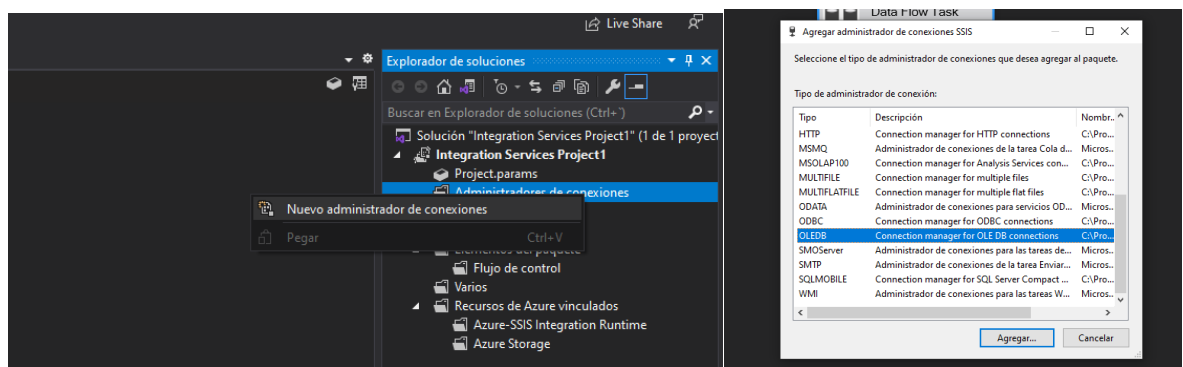
Almacén de datos (DataMart historial académico)

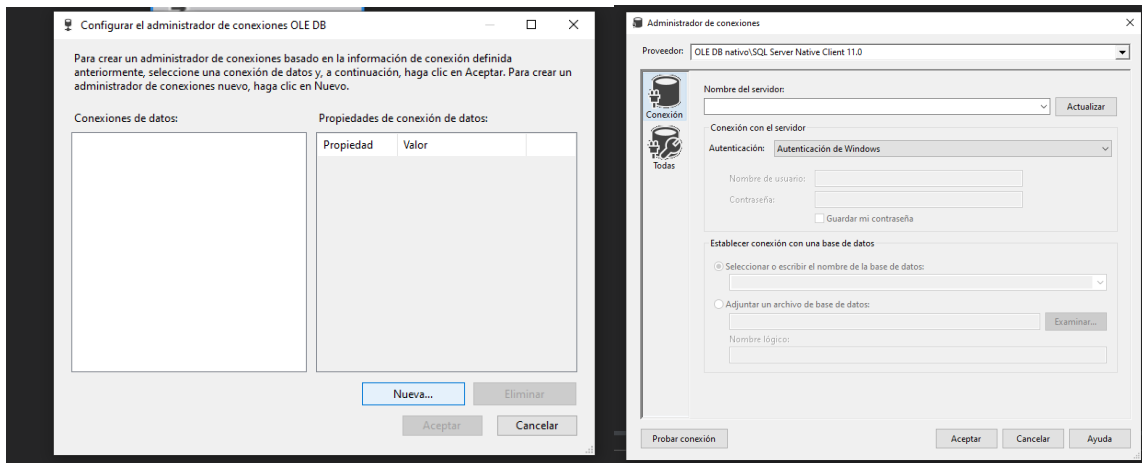


Se define lo que sería nuestro flujo de trabajo, el cual puede ser renombrado



Realizar la conexión:

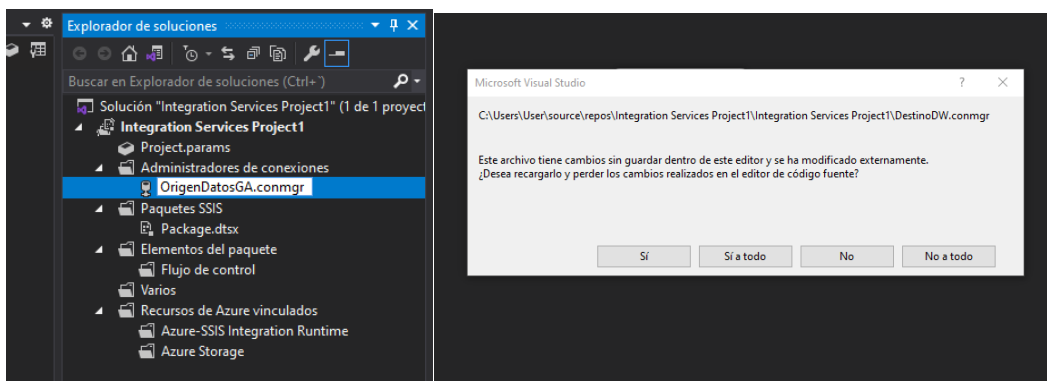
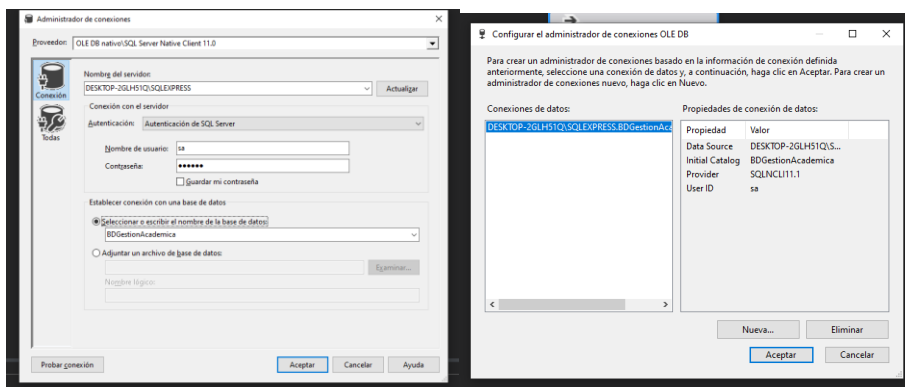




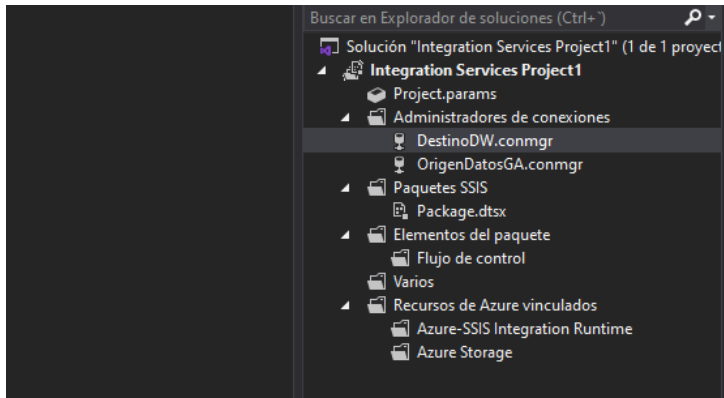
El nombre de su servidor variara en dependencia de su instancia de SQL, localhost (".") es para conexiones locales o se pueden utilizar instancias varias de SQL o nombres de dominio.

El tipo de autenticación será en dependencia del nivel de acceso que se tenga en la aplicación, para procesos ETL, dado que estos se ejecutan a nivel de CORE de los servidores se recomienda usar Windows Authenticate o nombres de dominio de active directory, siempre se recomienda guardar las credenciales si se usa SQL Authenticate.

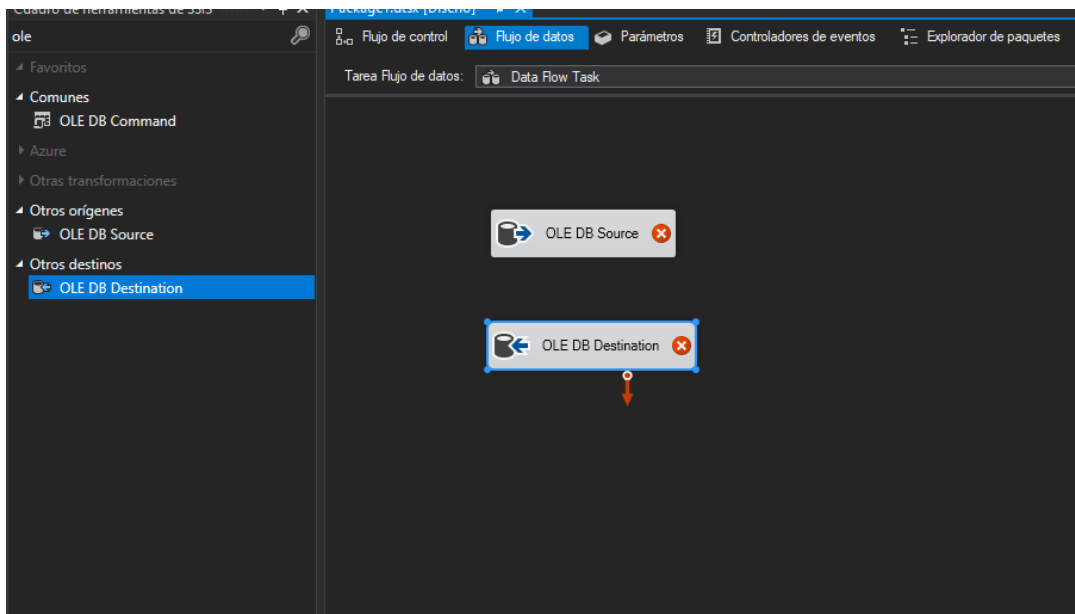
También es recomendable renombrar el nombre de la conexión para no tener confusiones, dar en la opción no a todo y luego cerrar el paquete y volver a abrirlo.



Así mismo se debe crear la conexión del destino de datos, y siempre actualizar cualquier instancia que se este utilizando después de cambiar el nombre.



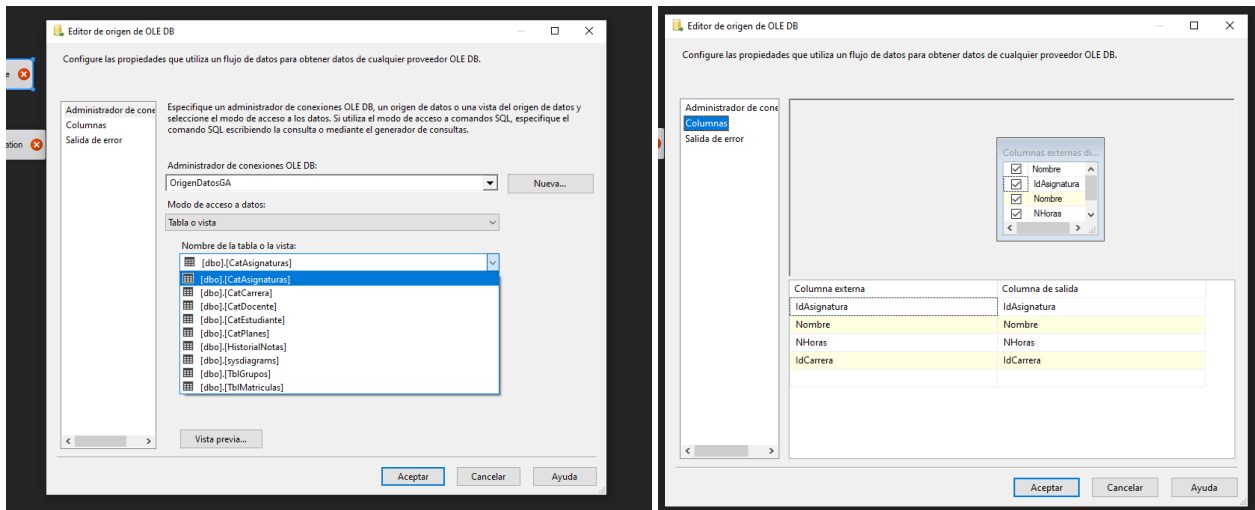
Luego se le da doble click al flujo de trabajo y esta nos mandara a la pestaña de flujo de datos donde tendremos herramientas específicas para el ETL y agregaremos un OLE DB source y un OLE DB Destination



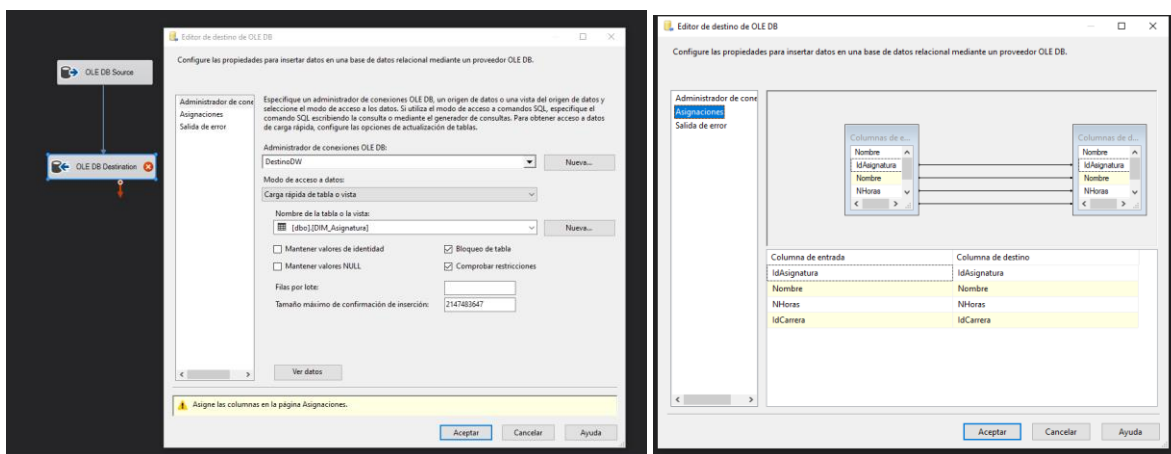
Luego al dar doble click sobre el control y aparecerá la configuración de este dónde definiremos que conexión utilizar, en este punto deberá determinar según el control, la conexión que utilizará además de seleccionar la tabla, vista o comando SQL que deberá ejecutar para la lectura o escritura de datos.

En el caso de el Source es de donde se extraerán los datos y en el caso del Destination es donde se guardarán los datos una vez procesados.

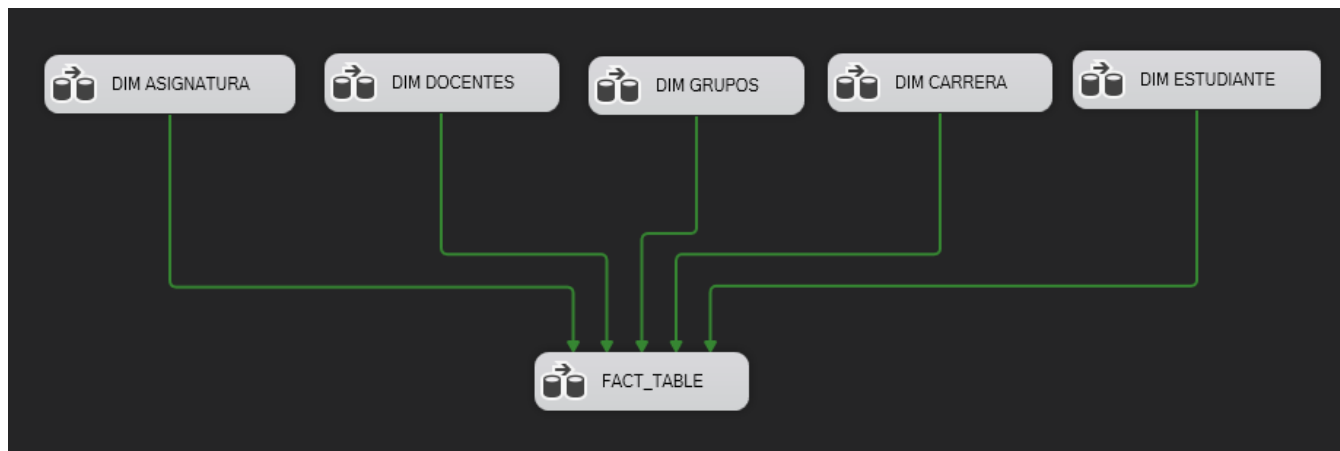
En los source además se podrán seleccionar del origen que columnas realmente me interesa cargar, importante sobre todo para depurar datos irrelevantes para el análisis y que no están incluidos en mi DataMart



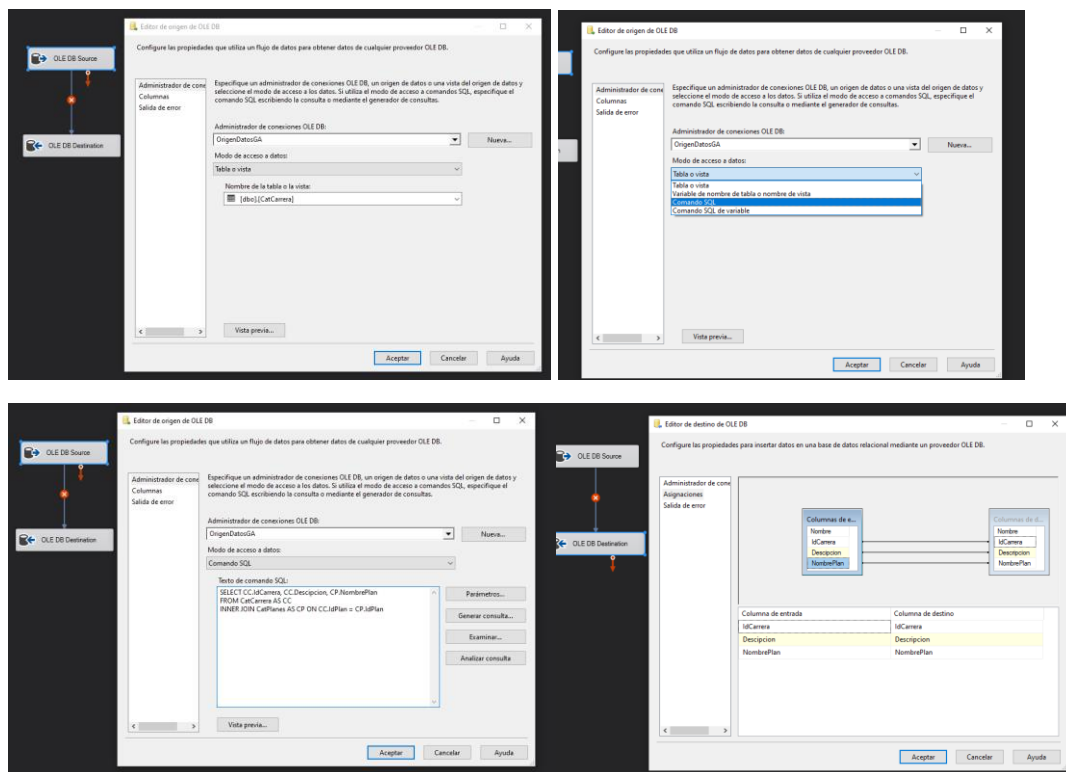
En el caso del destino priemro se debera unir con el source y posterior se definira a que dimencion esta dirigida, asi como definir en que orden estaran las columnas ordenadas, en este punto es de suma importancia que los tipos de datos esten correctamente definidos, para evitar complicaciones.



Dado que un DataWareHouse pose muchos Datamart y estos a su vez tienen muchas dimensiones podríamos dividirlos por flujos de trabajo (Data flow task) distinto cada uno con una tarea en específico dejando de ultimo la tabla de hechos para evitar problemas de identidad. Ordenándolos de esta manera todas las dimensiones se cargarán de forma paralela y una vez todas estén listas procedería a crear los hechos.

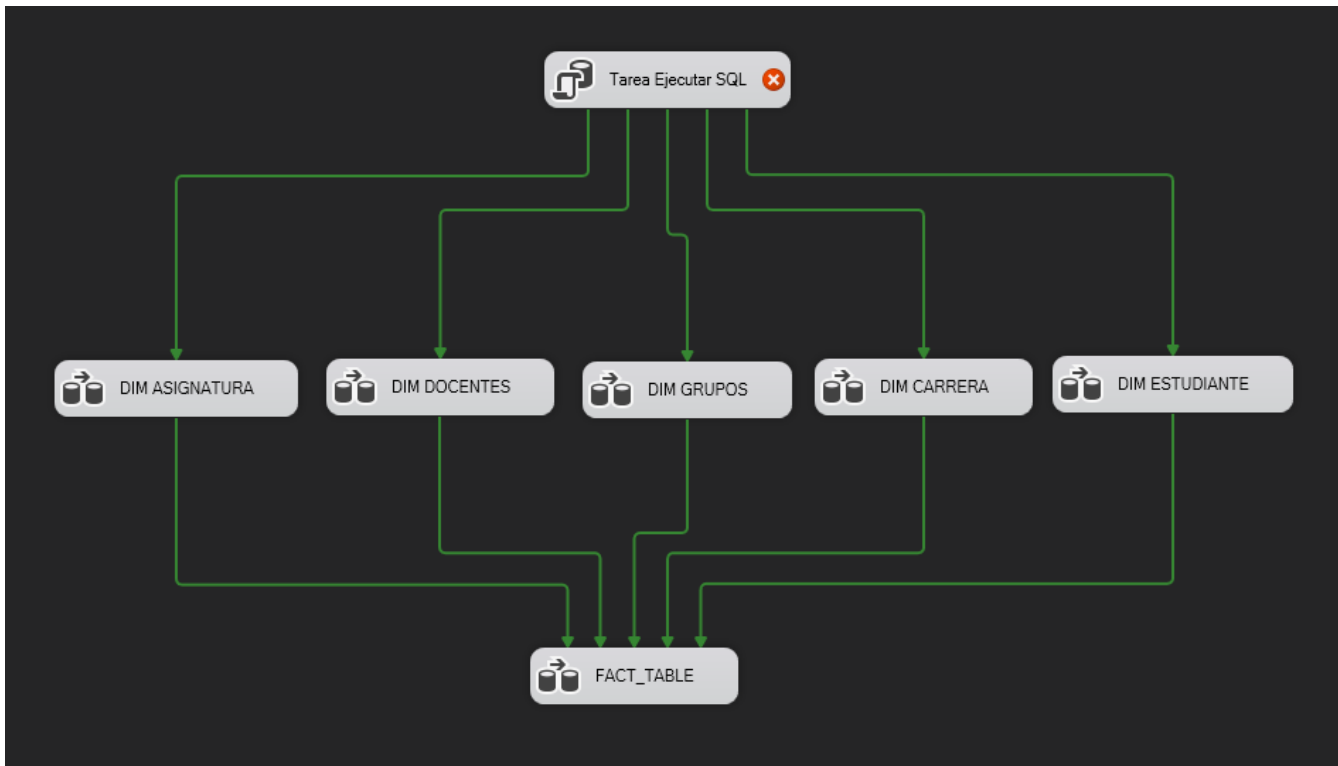


En el caso de la dimensión de carrera recordemos que, según este modelo, esta dimensión es resultante de la unión de los planes de estudio con la lista de carreras, por lo tanto, la tabla de carreras no coincide con su dimensión por lo tanto se debe hacer una correcta transformación, esto se debe lograr por medio de la creación de vistas, consultas SQL o procedimiento almacenado de la base de datos de destino. Una vez realizado solamente se debe de especificar la asignación de las columnas en el destino.



Así mismo se debe realizar un proceso similar para los Hechos dado que estos son resultantes de la unión de varias tablas.

Por ultimo siempre es recomendable ejecutar algunas tareas previas, como el borrado del Datamart para evitar redundancia de datos y errores de lectura, este método puede ser cuestionado o realizado de otra manera evitando que se carguen datos repetitivos. Para términos académicos se realizara de esta forma.



Editor de la tarea Ejecutar SQL

Configure las propiedades requeridas para ejecutar instrucciones SQL y procedimientos almacenados mediante la conexión seleccionada.

General

Asignación de parámetros
Conjunto de resultados
Expresiones

Conjunto de resultados: Ninguno

General

Name: Tarea Ejecutar SQL
Description: Tarea Ejecutar SQL

Instrucción SQL

ConnectionType: OLE DB
Connection: [seleccionada]
SQLSourceType: Entrada directa
SQLStatement: [vacío]
IsQueryStoredProcedure: False
BypassPrepare: True

Opciones

TimeOut: 0
CodePage: 1252
TypeConversionMode: Permitido

Name: Especifica el nombre de la tarea.

Examinar... Generar consulta... Analizar consulta

Aceptar Cancelar Ayuda

Editor de la tarea Ejecutar SQL

Configure las propiedades requeridas para ejecutar instrucciones SQL y procedimientos almacenados mediante la conexión seleccionada.

General

Asignación de parámetros
Conjunto de resultados
Expresiones

Conjunto de resultados: Ninguno

General

Name: Tarea Ejecutar SQL
Description: Tarea Ejecutar SQL

Instrucción SQL

ConnectionType: OLE DB
Connection: [seleccionada]
SQLSourceType: Nueva conexión...
SQLStatement: DestinoDWH
IsQueryStoredProcedure: OrigenDatosGA
BypassPrepare: true

Opciones

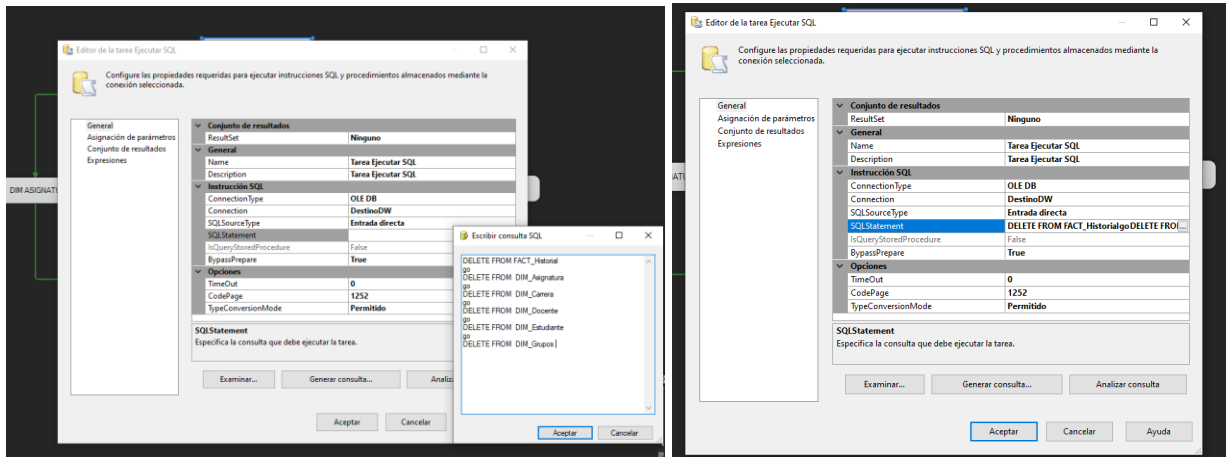
TimeOut: 0
CodePage: 1252
TypeConversionMode: Permitido

Connection

Especifica la conexión usada para obtener acceso a los datos.

Examinar... Generar consulta... Analizar consulta

Aceptar Cancelar Ayuda



Una vez realizada todas las configuraciones solamente procederíamos a depurar el proyecto y si todo esta bien hecho tendríamos este resultado

