

A dark blue vertical bar on the left side of the page. A blue arrow points to the right from the bar, containing the date.

7-12-2023

ETL Proyecto B

Diseño y Explotación de Almacenes
de Datos

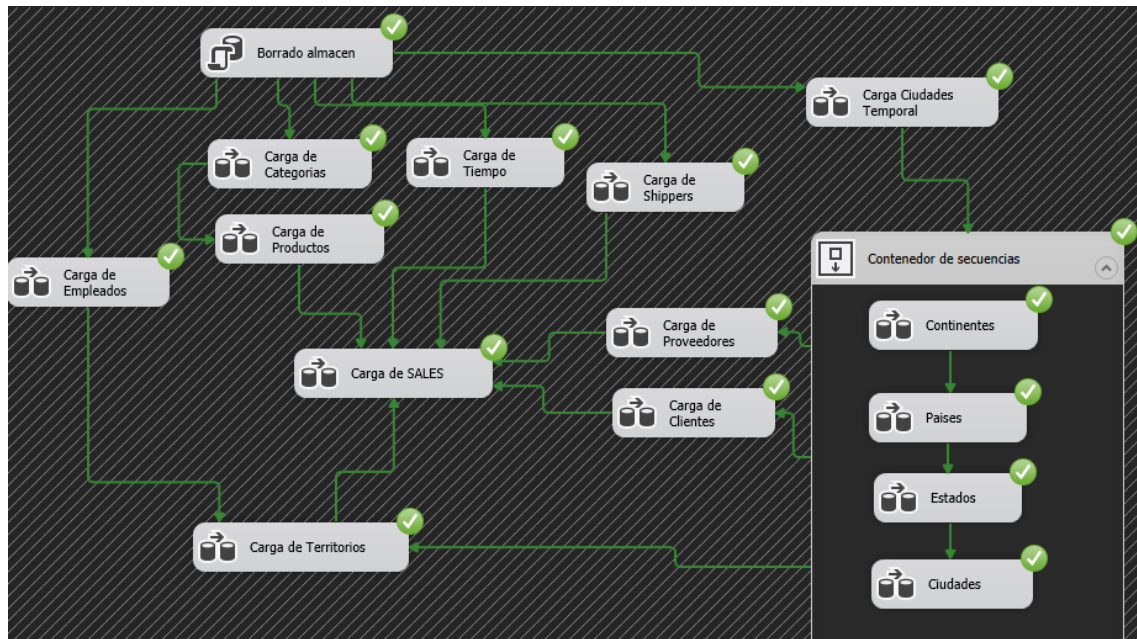
Several thin, curved lines in dark blue and light grey originate from the bottom left and curve upwards and to the right.

Roberto Navarro García
UMA

Contenido

CARGA ALMACÉN NORTHWIND	2
DIFICULTADES ENCONTRADAS:.....	2
Cambios en la Tarea 1	2
CARGA PROYECTO B. LIGA FÚTBOL	3
Tarea Borrado Almacén	3
Tarea Cargar Ligas	4
Tarea Cargar Temporadas	5
Tarea Cargar Evento	6
Tarea Cargar Ronda.....	7
Carga de Territorios	8
Tarea Cargar Continente	8
Tarea Cargar País.....	9
Tarea Cargar Ciudad.....	11
Tarea Cargar Campo.....	13
Tarea Cargar Equipo.....	15
Tarea Cargar Tiempo.....	17
Tarea Cargar Partido (Hecho).....	21
Carga Completa.....	22
Dificultades encontradas:	23
BIBLIOGRAFÍA	23

CARGA ALMACÉN NORTHWIND



DIFICULTADES ENCONTRADAS:

Para la realización de esta práctica guiada no he encontrado complicaciones a destacables.

Cambios en la Tarea 1

Cambios tabla tiempo: El ID de la tabla tiempo se ha cambiado a tipo Entero y ahora es autogenerado.

Los IDs de todas las tablas pertenecientes a los territorios ("Continente", "País", "Ciudad" y "Campo") ahora son autogenerados.

CARGA PROYECTO B. LIGA FÚTBOL

Tarea Borrado Almacén

Lo primero será añadir una tarea del tipo “Tarea Ejecutar SQL” para que lo primero que se realice en cada ejecución sea vaciar el almacén y de esta forma evitar errores por duplicidad de los datos. Esta tarea hay que conectarla al almacén (llamado ProyectoB en mi caso) y escribir las sentencias SQL que tiene que realizar.

La tarea de SQL queda de la siguiente 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

ResultSet	Ninguno
-----------	---------

General

Name	Borrado Almacén
Description	Tarea Ejecutar SQL

Instrucción SQL

ConnectionType	ADO.NET
Connection	LocalHost.ProyectoB
SQLSourceType	Entrada directa
SQLStatement	delete from Liga;delete from Temporada;
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

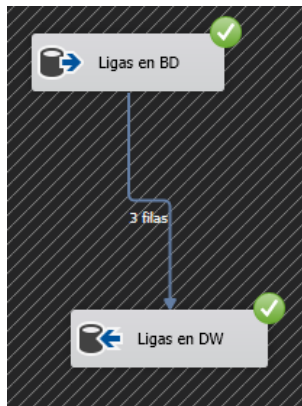
Escribir consulta SQL

```
delete from Partido;
delete from Ronda;
delete from Evento;
delete from Liga;
delete from Temporada;
delete from Equipo;
delete from Campo;
DBCC CHECKIDENT (Campo, RESEED, 0);
delete from Ciudad;
DBCC CHECKIDENT (Ciudad, RESEED, 0);
delete from Pais;
DBCC CHECKIDENT (Pais, RESEED, 0);
delete from Continente;
DBCC CHECKIDENT (Continente, RESEED, 0);
delete from Tiempo;
DBCC CHECKIDENT (Tiempo, RESEED, 0);
```

Aceptar Cancelar

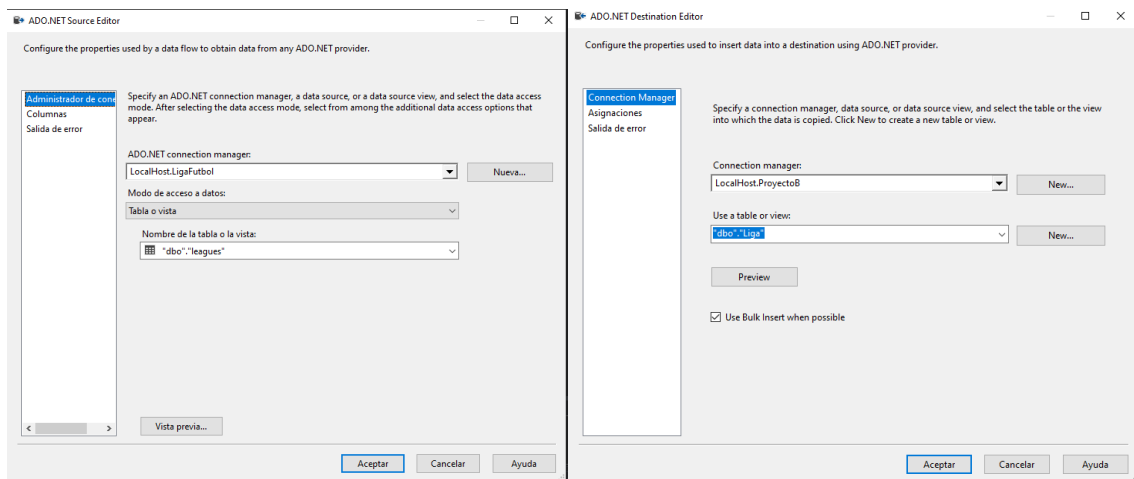
Las líneas “DBCC CHECKIDENT (‘Tabla’, RESEED, 0);” sirven para que se reinicien los ID autogenerados.

Tarea Cargar Ligas

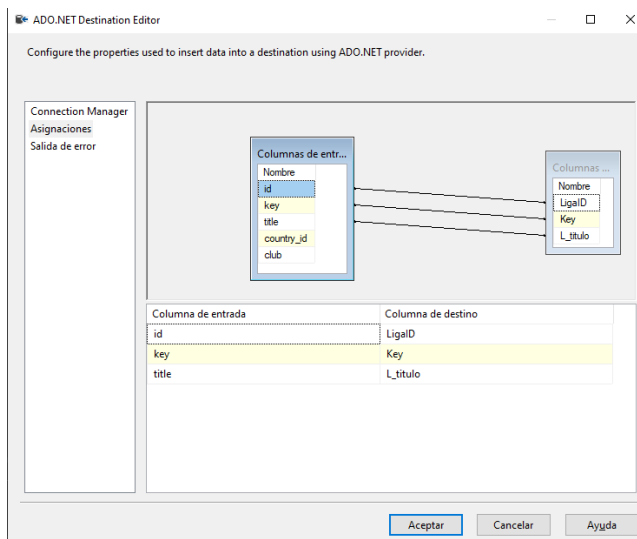


Se selecciona un origen de ADO.NET. En el origen se conecta con la BBDD (en mi caso “LigaFutbol”) y en el modo de acceso a los datos la tabla “Leagues”.

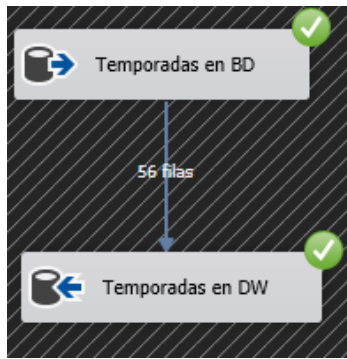
El origen se conecta a un destino de ADO.NET. Este se conecta al Data Warehouse (“ProyectoB” en mi caso) y la tabla “Liga”.



En el destino se realizan las siguientes asignaciones.



Tarea Cargar Temporadas



Igual que en Ligas, se selecciona un origen y un destino de ADO.NET. En el origen se selecciona la tabla “seasons” y en el destino la tabla “Temporada”.

Two screenshots of SQL Server Data Tools configuration windows. The left window is the 'ADO.NET Source Editor' for 'Administrador de con...' with 'Columnas' selected. It shows 'LocalHost.LigaFutbol' as the connection manager, 'Modo de acceso a datos' as the data access mode, and 'dbo.seasons' as the table/view. The right window is the 'ADO.NET Destination Editor' for 'Asignaciones' with 'Salida de error' selected. It shows 'LocalHost.ProyectoB' as the connection manager and 'dbo.Temporada' as the table/view. Both windows have 'Aceptar', 'Cancelar', and 'Ayuda' buttons at the bottom.

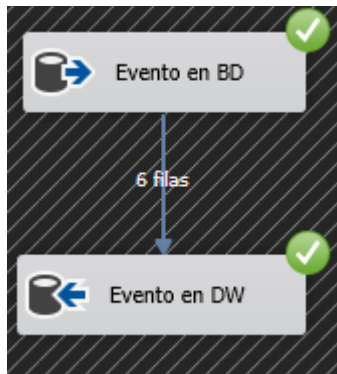
En el destino se realizan las siguientes asignaciones:

A screenshot of the 'ADO.NET Destination Editor' showing the 'Asignaciones' (Mappings) tab. It displays two column lists: 'Columnas de entrada' (Input Columns) and 'Columnas de destino' (Destination Columns). The input columns are 'id', 'key', and 'title'. The destination columns are 'temporadaID', 'Key', and 'Titulo'. Lines connect 'id' to 'temporadaID', 'key' to 'Key', and 'title' to 'Titulo'. Below the column lists, a table summarizes these mappings:

Columna de entrada	Columna de destino
id	temporadaID
key	Key
title	Titulo

The 'Aceptar', 'Cancelar', and 'Ayuda' buttons are at the bottom.

Tarea Cargar Evento



Evento está conectado con liga y temporada. Primero pensé en colocar un origen de datos para los datos del evento en la base de datos y otro para los IDs de liga y temporada en el DW pero pensé que los IDs de estos dos en el DW no son autogenerados y se alimentan de la BD. Por este motivo y de la misma forma que en los casos anteriores, solo necesito el origen de datos de la BD a la tabla “events” y el destino al DW en la tabla “Evento”.

The screenshot shows two side-by-side windows from SQL Server Data Tools. The left window is the 'ADO.NET Source Editor' and the right is the 'ADO.NET Destination Editor'. Both are configured for a data flow.

ADO.NET Source Editor Configuration:

- ADO.NET connection manager: LocalHost.LigaFutbol
- Modo de acceso a datos: Tabla o vista
- Nombre de la tabla o la vista: "dbo"."events"

ADO.NET Destination Editor Configuration:

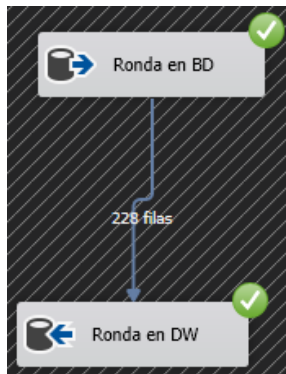
- Connection manager: LocalHost.Proyecto8
- Use a table or view: "dbo"."Evento"
- ☒ Use Bulk Insert when possible

Se realizan las siguientes asignaciones.

The screenshot shows the 'ADO.NET Destination Editor' with the 'Asignaciones' (Mappings) tab selected. It displays two columns lists: 'Columnas de entrada' (Input columns) and 'Columnas de destino dis...' (Destination columns). Lines connect the input columns to the destination columns.

Columna de entrada	Columna de destino
id	ID
key	Clave
league_id	Liga_LigaID
season_id	Temporada_temporadaID
start_at	Tiempo_Fecha

Tarea Cargar Ronda



Basta con un origen ADO.NET a la tabla “rounds” de la base de datos y un destino a la tabla “Ronda” en el DW.

Two screenshots of SQL Server Data Tools configuration windows. The left window is the 'ADO.NET Source Editor' and the right is the 'ADO.NET Destination Editor'. Both show the configuration for an ADO.NET provider.

ADO.NET Source Editor:

- ADO.NET connection manager: LocalHost.LigaFutbol
- Modo de acceso a datos: Tabla o vista
- Nombre de la tabla o la vista: "dbo"."rounds"

ADO.NET Destination Editor:

- Connection manager: LocalHost.Proyecto8
- Use a table or view: "dbo"."Ronda"
- ☒ Use Bulk Insert when possible

Con las siguientes asignaciones:

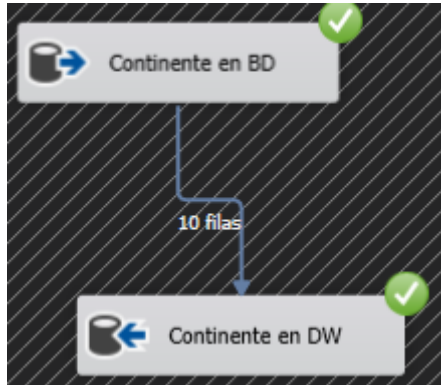
A screenshot of the 'ADO.NET Destination Editor' showing the column mapping between the source table 'rounds' and the destination table 'Ronda'.

Columna de entrada	Columna de destino
id	RondaID
event_id	Evento_ID
title	Titulo
pos	Pos
end_at	Tiempo_Fecha2
start_at	Tiempo_Fecha

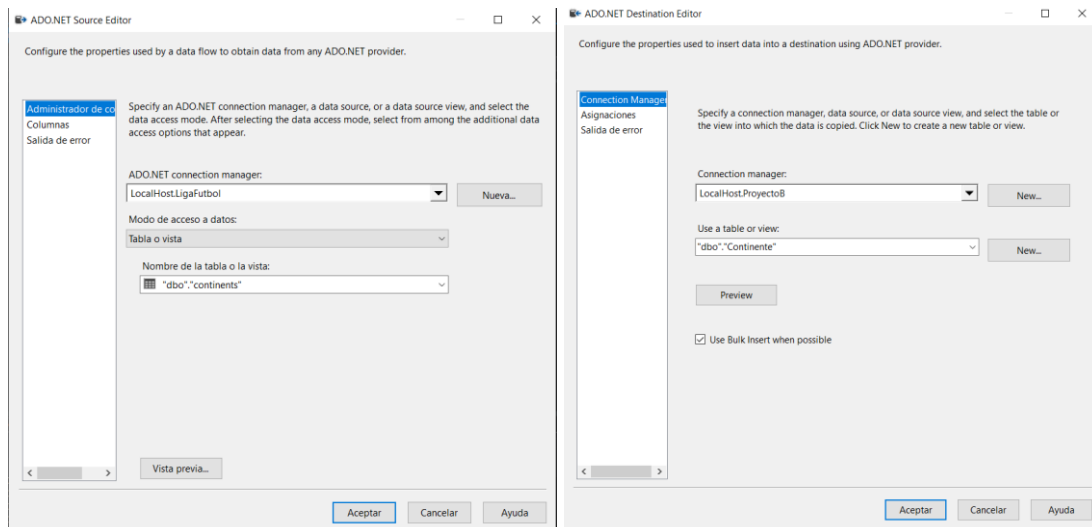
Carga de Territorios

La carga de todos los territorios irá incrustada en un contenedor de secuencias que contendrá, en concreto, la carga de Continente, País, Ciudad y Campo.

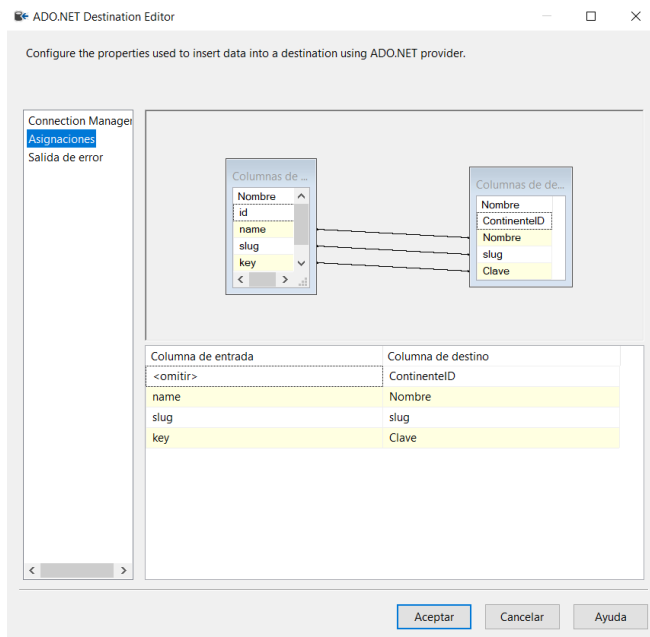
Tarea Cargar Continente



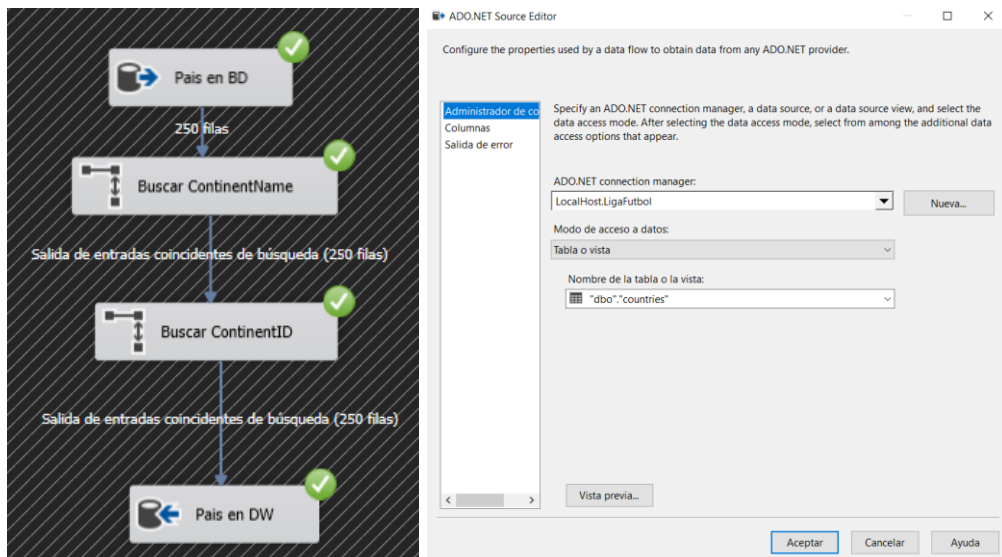
Basta con un origen ADO.NET a la tabla “continents” de la base de datos y un destino a la tabla “Continente” en el DW.



Tiene las siguientes asignaciones. El ContinenteID se omite ya que es autogenerado.

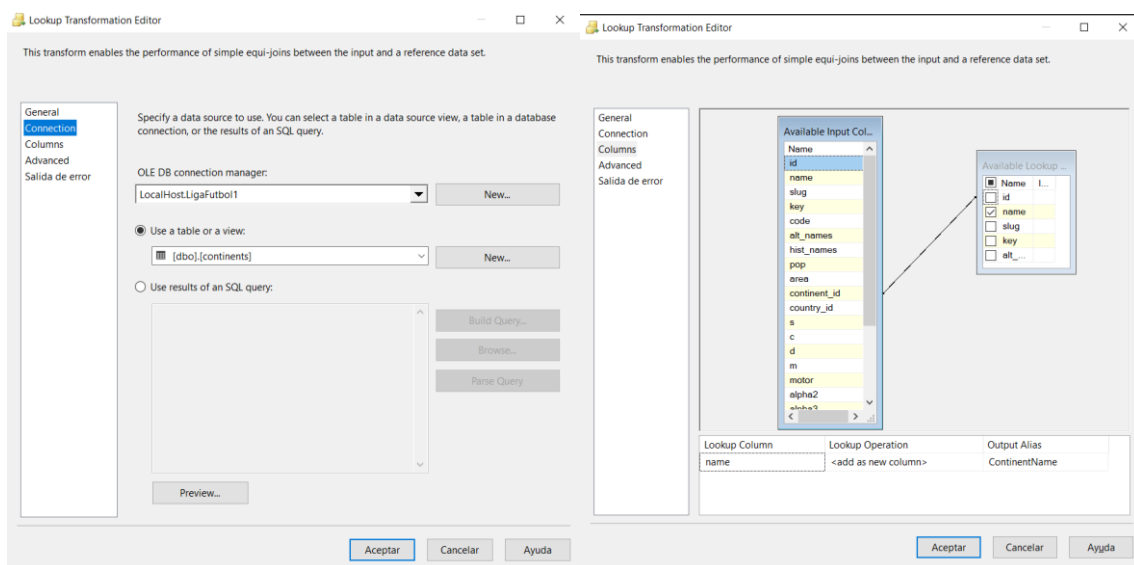
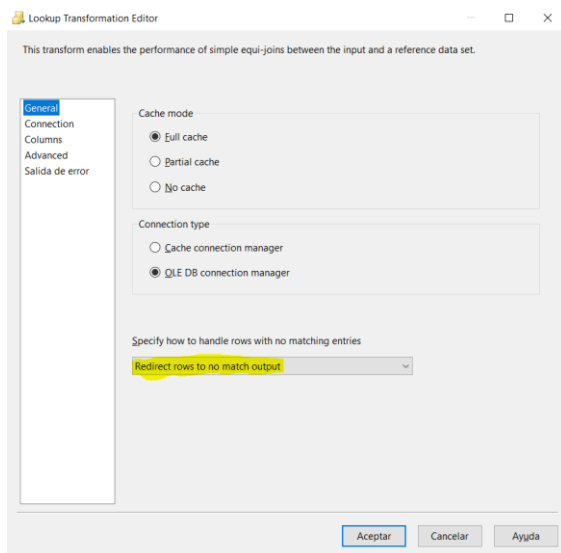


Tarea Cargar País

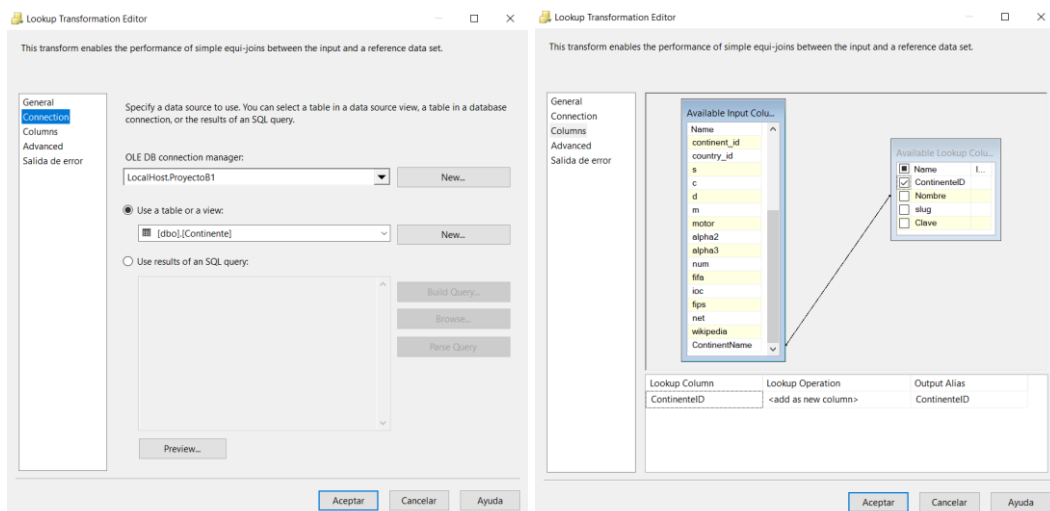


Se comienza a partir de un origen ADO.NET conectado a la tabla “country” de la base de datos.

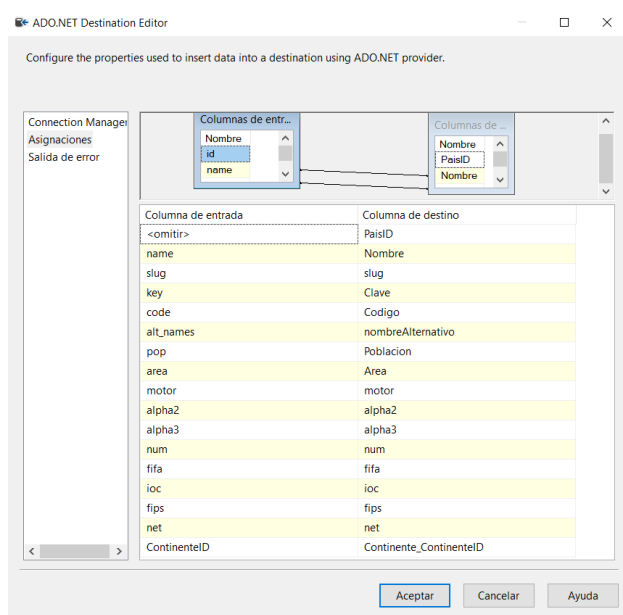
Como las claves de los territorios son autogeneradas, no nos sirve la clave del continente que tenemos en la base de datos y tenemos que buscar la correspondiente en el DW. Para ello necesitamos un bloque de búsqueda. Este bloque de búsqueda se conecta a la tabla “continents” de la base de datos y se utiliza para averiguar el nombre del continente asociado al ID del continente de la tabla “country”. Las salidas sin coincidencias se redirigen a la salida de las filas sin coincidencias.



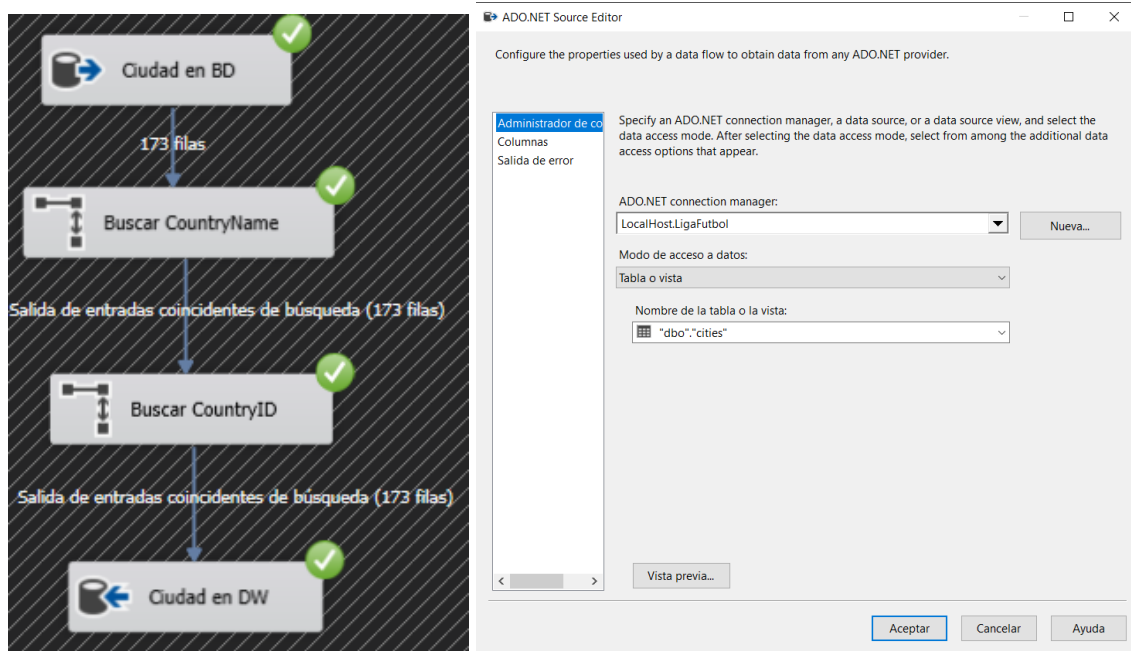
Después de este bloque, necesitamos otro como el anterior. En este caso el bloque se conectará a la tabla “Continente” del DW y buscará el ID del continente asociado al nombre que hemos buscado antes. Las salidas sin coincidencias se redirigen a la salida de las filas sin coincidencias.



Para finalizar, necesitamos un destino de ADO.Net que se conecte con la tabla “País” del DW. Estas son las asignaciones.

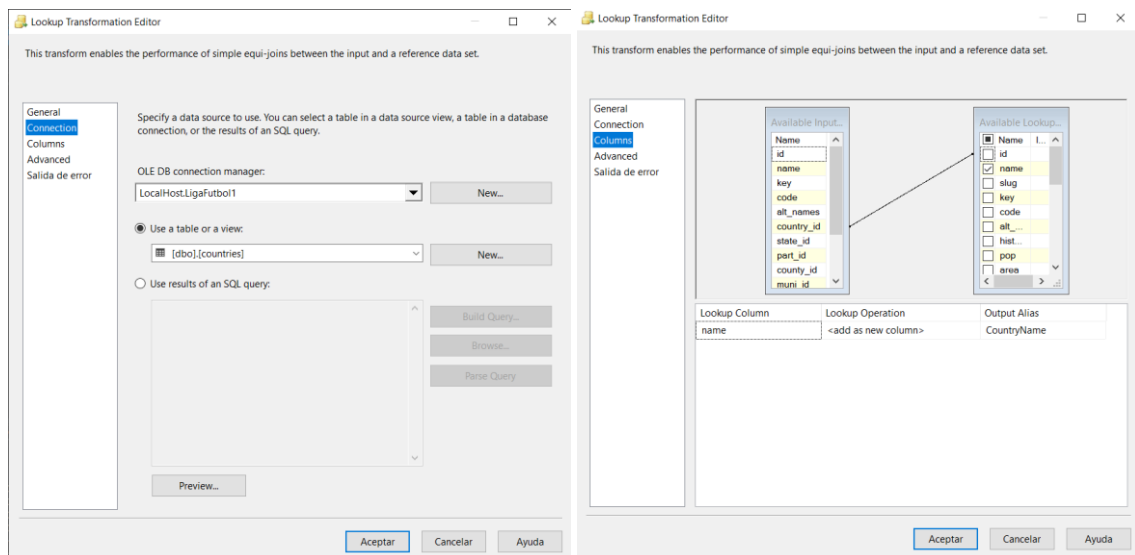


Tarea Cargar Ciudad

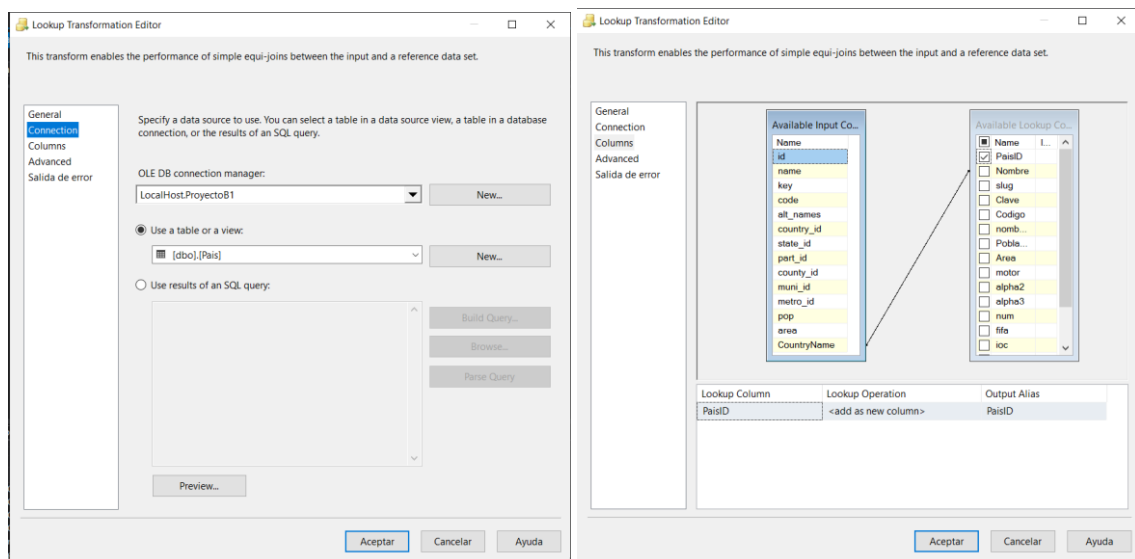


Se comienza a partir de un origen ADO.NET conectado a la tabla “cities” de la base de datos.

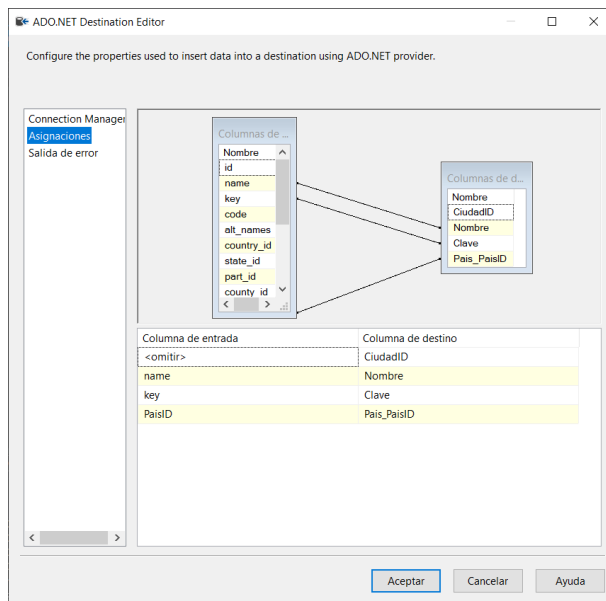
Como las claves de los territorios son autogeneradas, no nos sirve la clave del *país* que tenemos en la base de datos y tenemos que buscar la correspondiente en el DW. Para ello necesitamos un bloque de búsqueda. Este bloque de búsqueda se conecta a la tabla “countries” de la base de datos y se utiliza para averiguar el nombre del *país* asociado al ID del *país* de la tabla “cities”. Las salidas sin coincidencias se redirigen a la salida de las filas sin coincidencias.



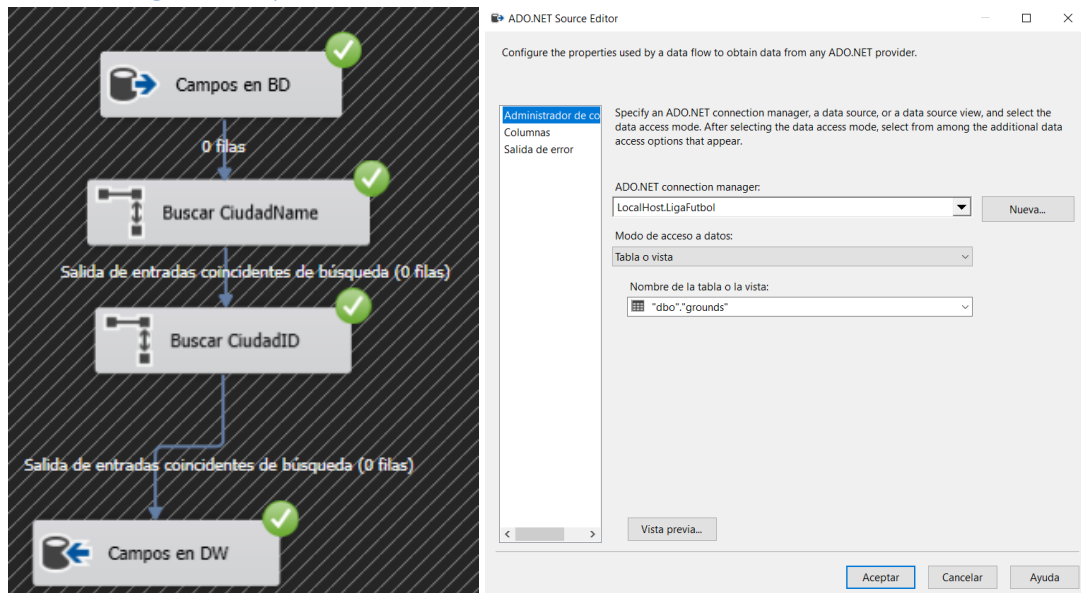
Después de este bloque, necesitamos otro como el anterior. En este caso, el bloque se conectará a la tabla “País” del DW y buscará el ID del país asociado al nombre que hemos buscado antes. Las salidas sin coincidencias se redirigen a la salida de las filas sin coincidencias.



Para finalizar, necesitamos un destino de ADO.Net que se conecte con la tabla “Ciudad” del DW. Estas son las asignaciones.

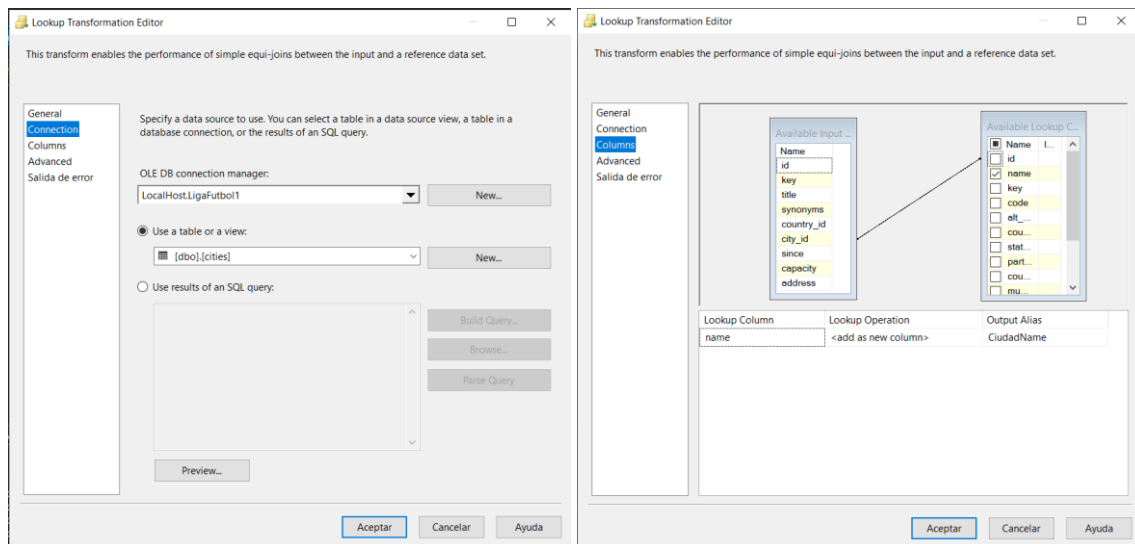


Tarea Cargar Campo

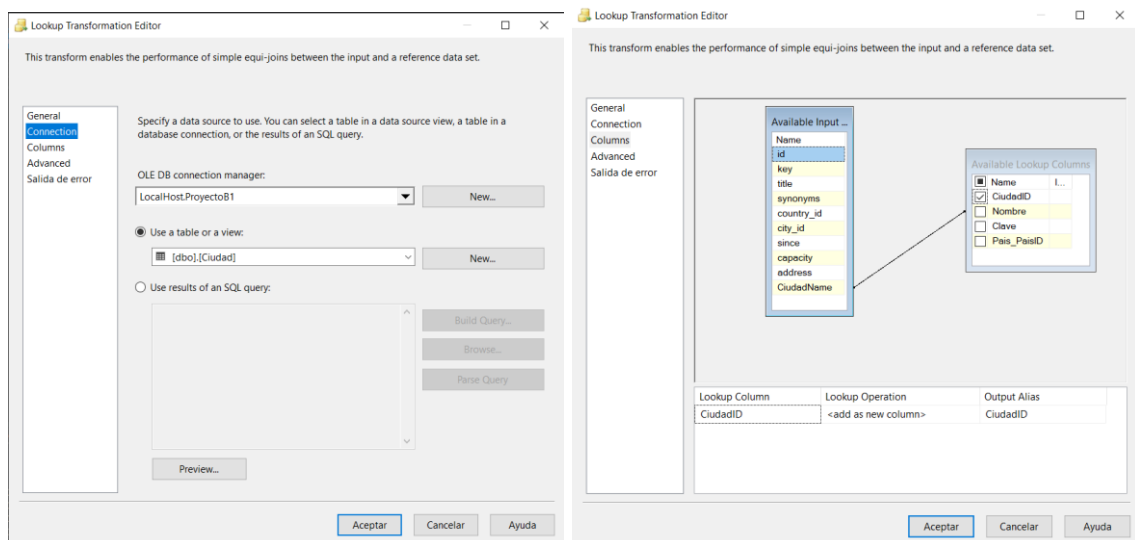


Se comienza a partir de un origen ADO.NET conectado a la tabla “grounds” de la base de datos.

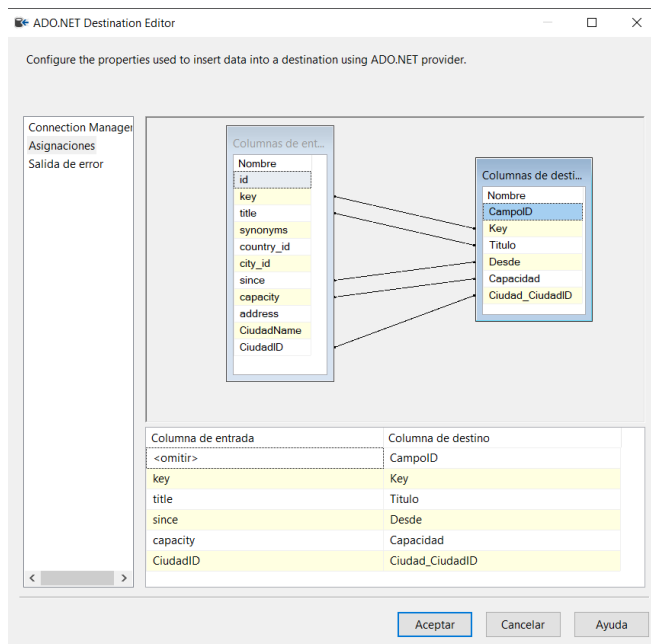
Como las claves de los territorios son autogeneradas, no nos sirve la clave de la *ciudad* que tenemos en la base de datos y tenemos que buscar la correspondiente en el DW. Para ello necesitamos un bloque de búsqueda. Este bloque de búsqueda se conecta a la tabla “cities” de la base de datos y se utiliza para averiguar el nombre de la *ciudad* asociado al ID de la *ciudad* de la tabla “grounds”. Las salidas sin coincidencias se redirigen a la salida de las filas sin coincidencias.



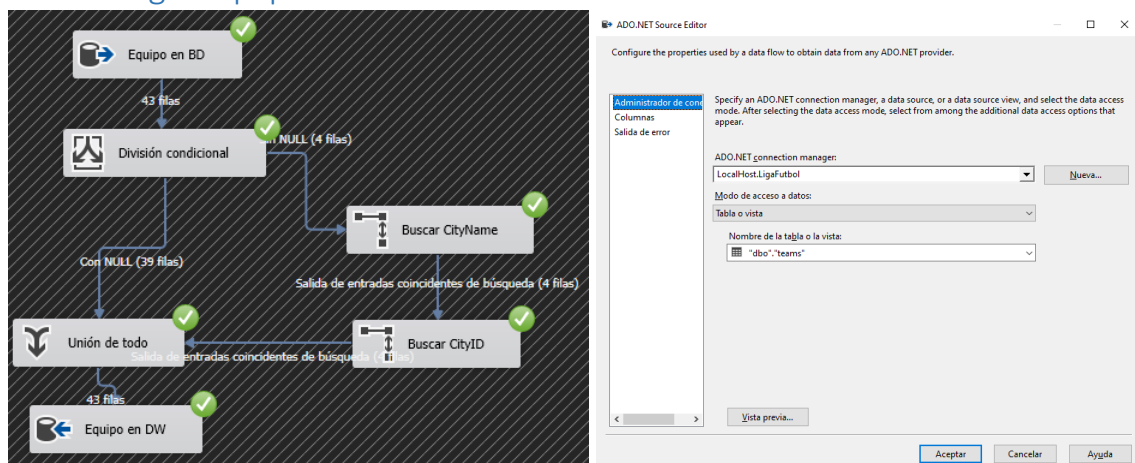
Después de este bloque, necesitamos otro como el anterior. En este caso, el bloque se conectará a la tabla “Ciudad” del DW y buscará el ID de la ciudad asociado al nombre que hemos buscado antes. Las salidas sin coincidencias se redirigen a la salida de las filas sin coincidencias.



Para finalizar, necesitamos un destino de ADO.Net que se conecte con la tabla “Campo” del DW. Estas son las asignaciones.

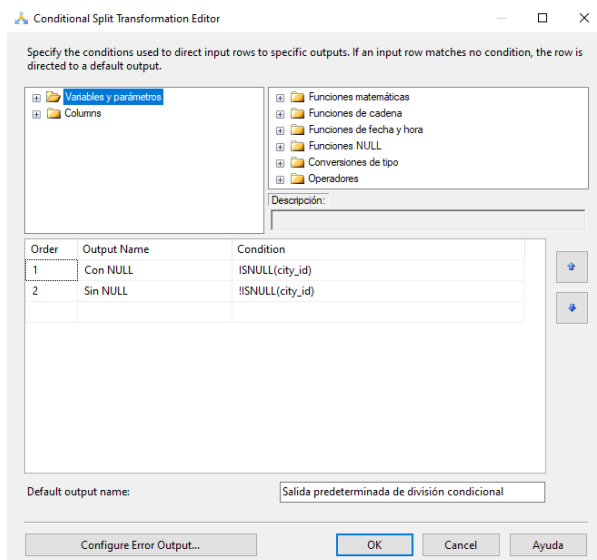


Tarea Cargar Equipo



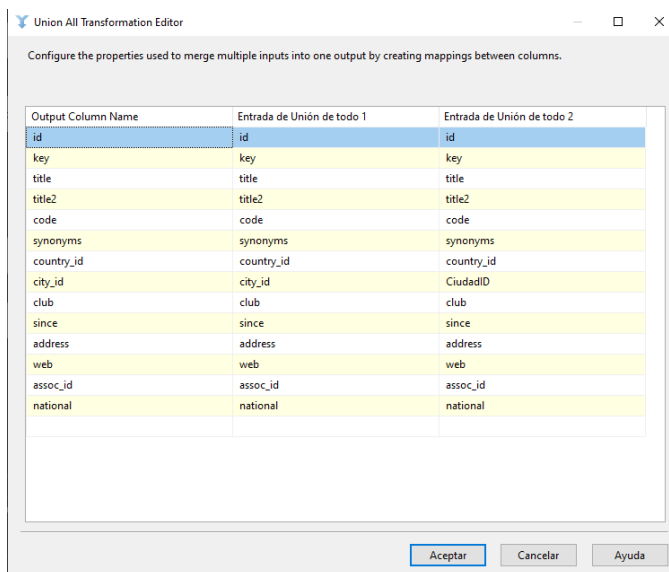
Se comienza a partir de un origen ADO.NET conectado a la tabla “teams” de la base de datos.

Lo primero es poner un bloque de división condicional para comprobar si las filas tienen el cityID a null o no.

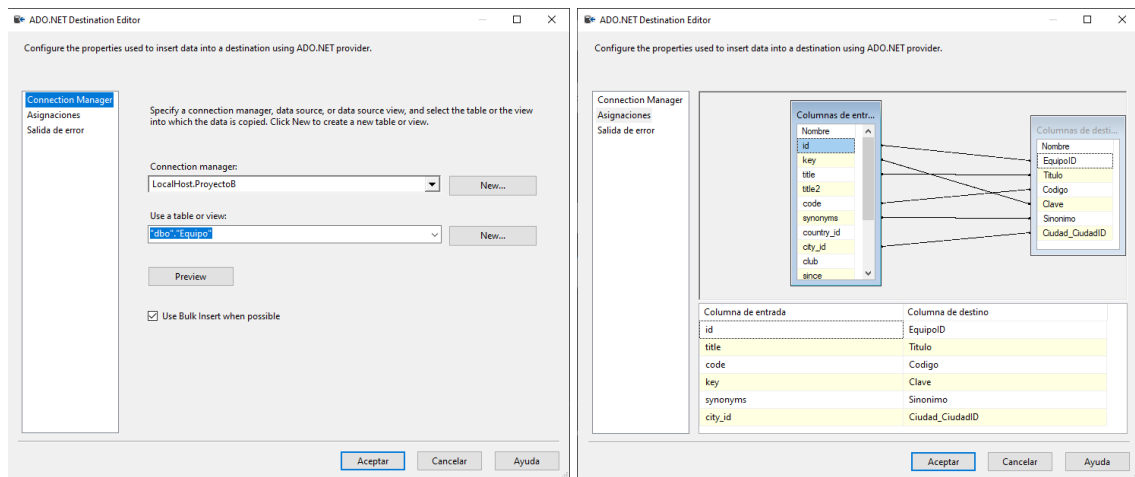


Esto es necesario porque la clave primaria de la tabla Ciudad es autogenerada, por lo que, si tiene valor distinto de null, hay que buscarla en el DW. Esto se hace con búsquedas de la misma manera que se hizo en 'Tarea cargar Campo', primero buscar el nombre de la ciudad en la base de datos y después el ID correspondiente en el DW.

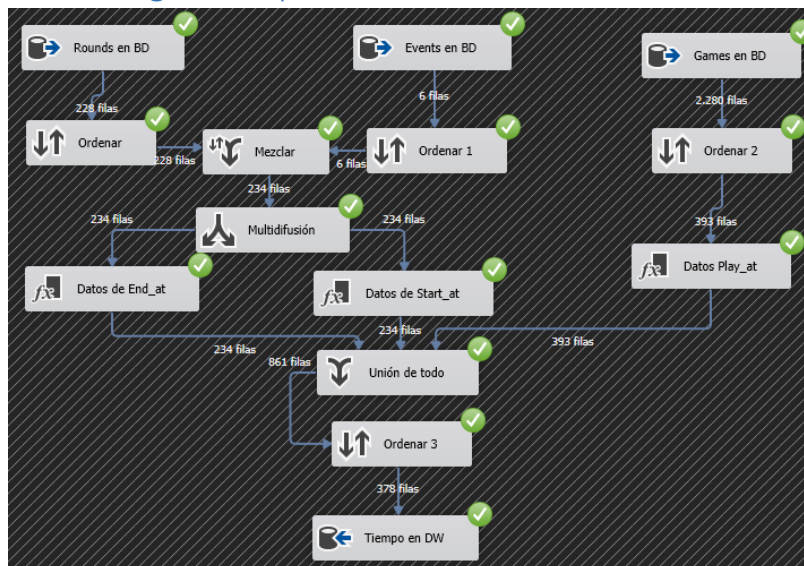
Las salidas nulas de la división y las salidas después del proceso de búsqueda se unen en un bloque de Unión de todo.



Finalmente se conecta con un destino de ADO.Net conectado a la tabla Equipo del DW con las siguientes asignaciones.



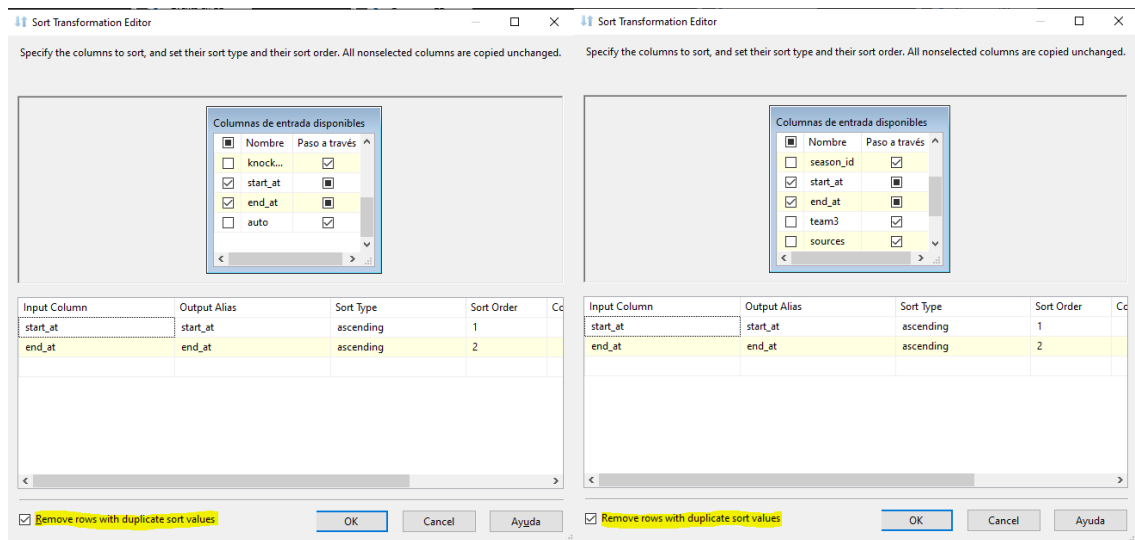
Tarea Cargar Tiempo



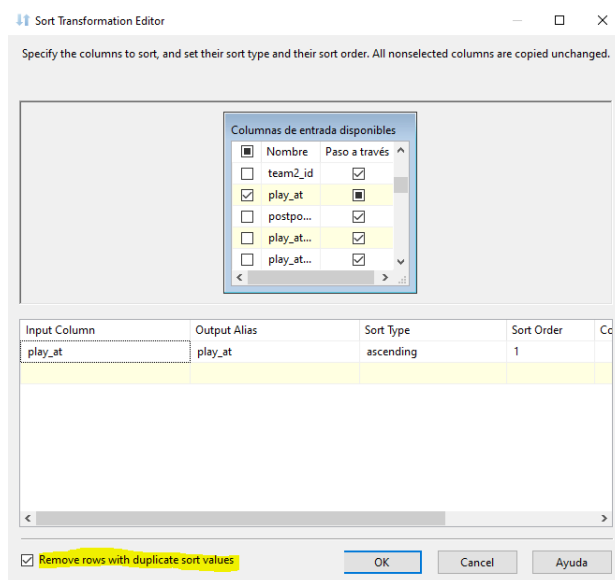
Los datos temporales los sacamos de tres fuentes dentro de la base de datos: las tablas “rounds”, “events” y “games” conectadas a un Origen ADO.Net.

Cada origen se conecta a un bloque Ordenar para seleccionar las filas con datos temporales y eliminar las filas repetidas.

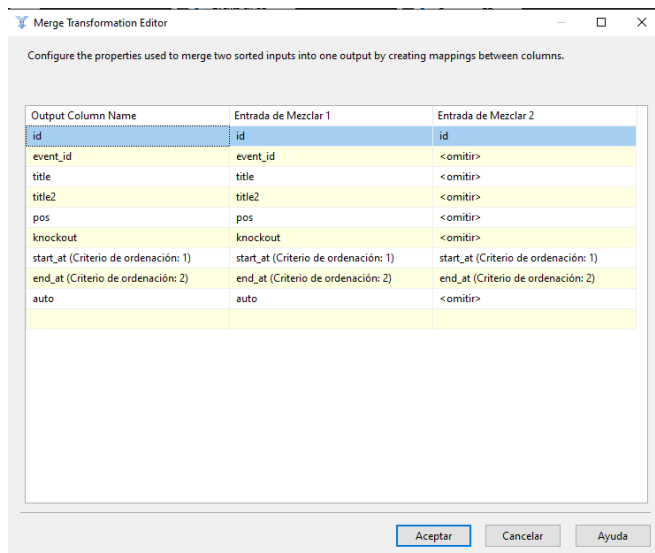
En los bloques Ordenar (rounds) y Ordenar 1 (events) se seleccionan los datos ‘start_at’ y ‘end_at’ y se selecciona la casilla de eliminar filas repetidas.



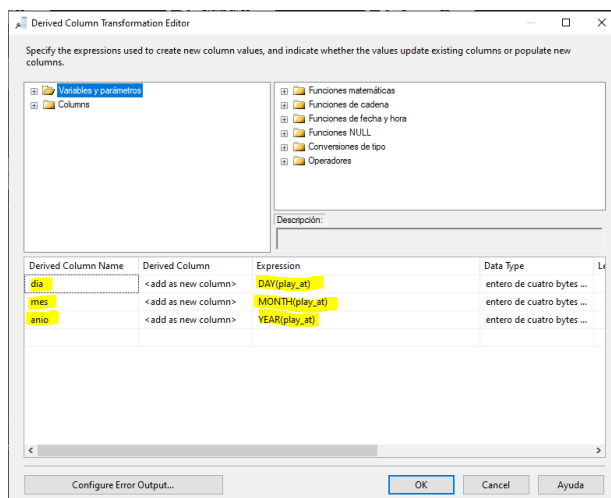
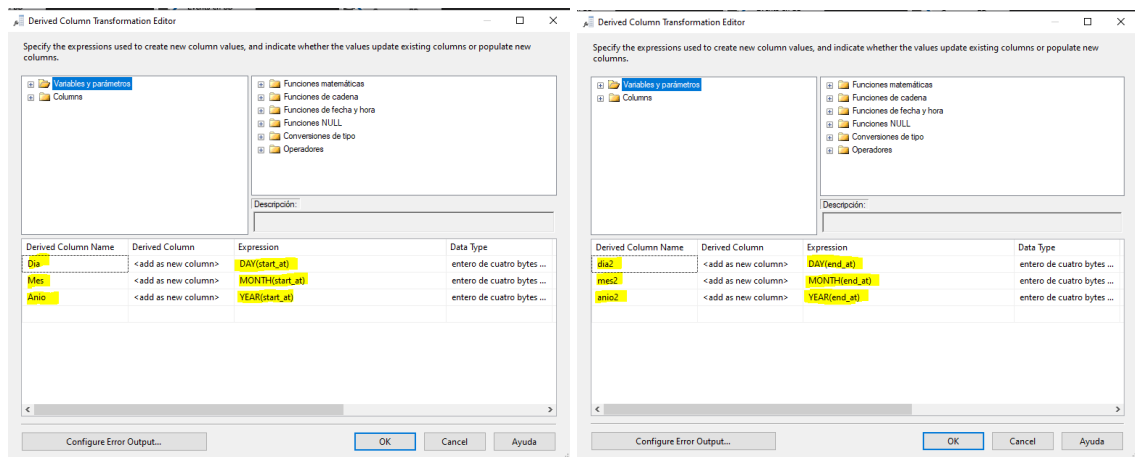
En el bloque Ordenar 2 (games) se seleccionan el atributo 'play_at'. Los atributos 'play_atv2' y 'play_atv3' no se seleccionan por estar vacíos.



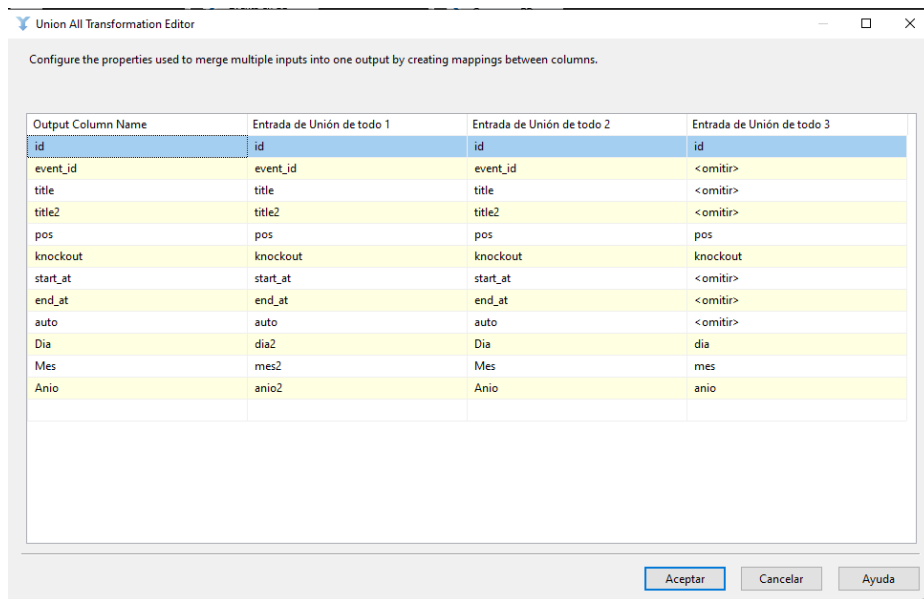
Los bloques Ordenar y Ordenar 1 se juntan en un bloque Mezclar para unir los dos pares de atributos que se llaman igual.



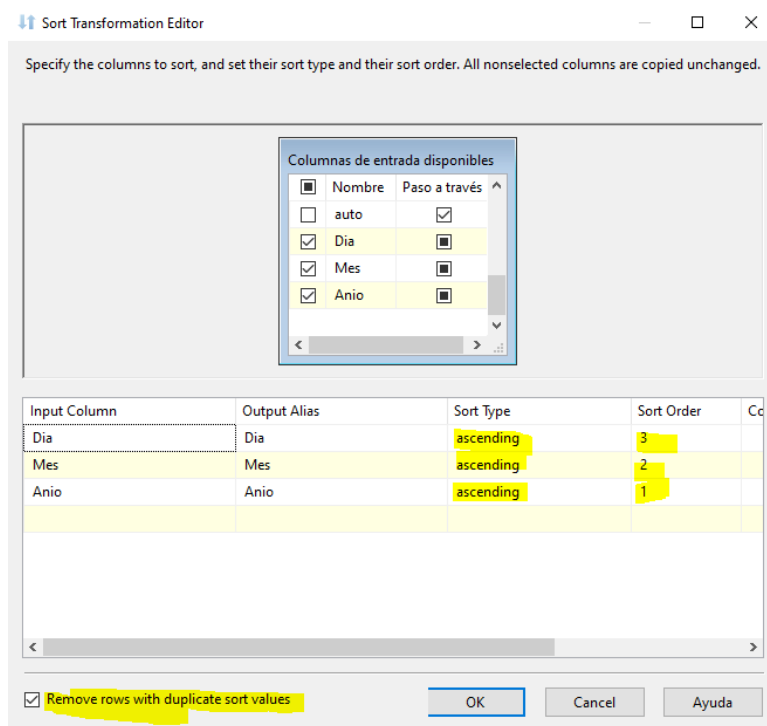
De la mezcla sale un bloque de Multidifusión para tratar por separado los datos de 'start_at' y 'end_at'. Estos datos los tratamos con un bloque de Columna derivada que utilizaremos para almacenar el día, mes y año que forman las fechas en atributos separados. Esto mismo se hace con Ordenar 2 (play_at).



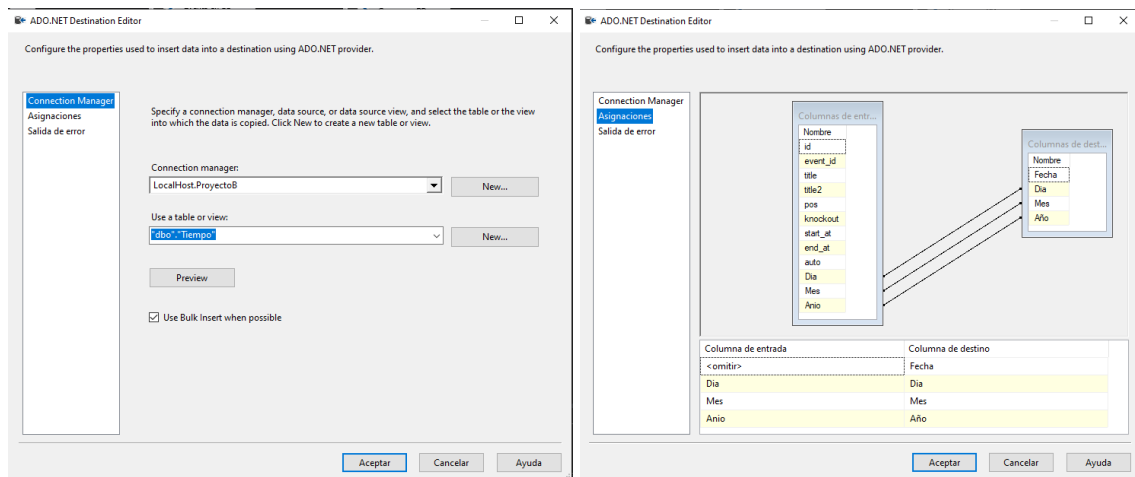
Todo esto se une en un bloque de Unión de Todo.



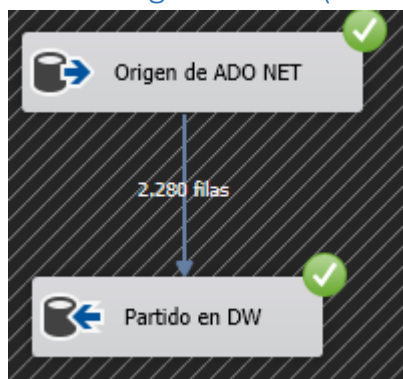
Después de unir todos los datos los pasamos por un bloque de Ordenar en el que seleccionamos los datos que nos interesan (Dia, Mes y Anio) y los ordenamos de forma ascendente con orden de prioridad Año (1), Mes (2) y Dia (3). También seleccionamos la casilla para eliminar las filas repetidas.



Finalmente se conecta a un Destino de ADO.Net con la tabla "Tiempo" en el DW y con las siguientes asignaciones. El atributo Fecha de la tabla "Tiempo" actúa de identificador (tipo entero) y es autogenerado.



Tarea Cargar Partido (Hecho)



Para la carga del 'Hecho' ("Partido") se han extraído los datos de un origen ADO.Net conectado al DW y realizando la siguiente consulta SQL:

```
Select e1.EquipoID as equipo1, e2.EquipoID as equipo2, RondaID, G.id as juegoID,
T.Fecha, G.knockout, G.score1, G.score2, G.winner, G.pos, G.home
from LigaFutbol.dbo.games G join dbo.Equipo e1 on (G.team1_id = e1.EquipoID) join
dbo.Equipo e2 on (G.team2_id = e2.EquipoID) join dbo.Ronda on (G.round_id =
RondaID), Tiempo T
where T.Dia = DAY(G.play_at) and T.Mes = MONTH(G.play_at) and T.Año =
YEAR(G.play_at)
```

El destino sería de tipo ADO.Net conectado a la tabla "Partido" del DW. El origen y el destino con sus asignaciones quedarían de la siguiente manera.

ADO.NET Source Editor

Configure the properties used by a data flow to obtain data from any ADO.NET provider.

Administrador de conexiones

Columnas

Salida de error

ADO.NET connection manager:

LocalHost.Proyecto8

Nueva...

Modo de acceso a datos:

Comando SQL

Texto de comando SQL:

Select e1.EquipoID as equipo1, e2.EquipoID as equipo2, RondaID, G.id as juegoID, T.Fecha, G.knockout, G.score1, G.score2, G.winner, G.pos, G.home from LigaFutbol.dbo.games G join dbo.Equipo e1 on (G.team1_id = e1.EquipoID) join dbo.Equipo e2 on (G.team2_id = e2.EquipoID) join dbo.Ronda on (G.round_id = RondaID), Tiempo T where T.Dia = DAY(G.play_at) and T.Mes = MONTH(G.play_at) and T.Año = YEAR(G.play_at)

Generar consulta...

Examinar...

Vista previa...

Aceptar

Cancelar

Ayuda

ADO.NET Destination Editor

Configure the properties used to insert data into a destination using ADO.NET provider.

Connection Manager

Asignaciones

Salida de error

Connection manager:

LocalHost.Proyecto8

Nueva...

Use a table or view:

"dbo"."Partido"

Nueva...

Preview

☒ Use Bulk Insert when possible

Aceptar

Cancelar

Ayuda

ADO.NET Destination Editor

Configure the properties used to insert data into a destination using ADO.NET provider.

Connection Manager

Asignaciones

Salida de error

Columnas de entr...

Nombre

equipo1

equipo2

RondaID

Columnas de dest...

Nombre

Knockout

Score_1

Score_2

Ganador

Equipo_EquipoID

Equipo_EquipoID2

Ronda_RondaID

JuegoID

pos

Local

Campo_CampoID

Tiempo_Fecha

Columna de entrada	Columna de destino
knockout	Knockout
score1	Score_1
score2	Score_2
winner	Ganador
equipo1	Equipo_EquipoID
equipo2	Equipo_EquipoID2
RondaID	Ronda_RondaID
juegoID	JuegoID
pos	pos
home	Local
<omitir>	Campo_CampoID
Fecha	Tiempo_Fecha

Aceptar

Cancelar

Ayuda

Carga Completa

```

graph TD
    BorradoAlmacen[Borrado Almacén] --> CargaLiga[Carga de Liga]
    BorradoAlmacen --> CargaTemporadas[Carga de Temporadas]
    BorradoAlmacen --> CargaEquipo[Carga de Equipo]
    BorradoAlmacen --> CargaTerritorios[Carga de Territorios]
    CargaLiga --> CargaEvento[Carga de Evento]
    CargaLiga --> CargaRonda[Carga de Ronda]
    CargaTemporadas --> CargaEvento
    CargaTemporadas --> CargaTiempo[Carga de Tiempo]
    CargaEquipo --> CargaTiempo
    CargaEquipo --> CargaPartido[Carga de Partido]
    CargaEvento --> CargaPartido
    CargaRonda --> CargaPartido
    CargaTiempo --> CargaPartido
    CargaTerritorios --> CargaPartido
    CargaTerritorios --> CargaContinente[Carga de Continente]
    CargaTerritorios --> CargaPais[Carga de País]
    CargaTerritorios --> CargaCiudad[Carga de Ciudad]
    CargaTerritorios --> CargaCampos[Carga de Campos]
  
```

The diagram illustrates a data loading process flow for a football league database. It starts with a 'Borrado Almacén' (Delete Warehouse) step, which triggers the loading of various data components. The process is organized into a hierarchical structure where 'Carga de Territorios' (Load Territories) is the root, leading to 'Carga de Continente', 'Carga de País', 'Carga de Ciudad', and 'Carga de Campos'. Other components like 'Carga de Liga', 'Carga de Temporadas', 'Carga de Equipo', 'Carga de Evento', 'Carga de Tiempo', 'Carga de Ronda', and 'Carga de Partido' are interconnected, showing a complex data dependency and loading sequence. Each step is marked with a green checkmark, indicating successful completion.

Dificultades encontradas:

La principal dificultad encontrada ha sido la carga de la tabla “Tiempo” ya que tiene que extraer los datos de tres fuentes distintas para estar lo más completa posible. Esto conlleva la dificultad de unificar de forma correcta todos esos datos. Como hemos podido observar, se ha conseguido solventar.

La segunda dificultad por destacar sería la carga del Hecho por la necesidad de la consulta SQL para alimentarse de los datos de la BD y del DW. Aquí me he encontrado también con que, al estar la tabla “Campo” vacía, si la incluía en la consulta no devolvía datos. Esto se solucionará añadiendo datos a la tabla.

BIBLIOGRAFÍA

- Todo el material almacenado en el campus virtual de la asignatura Diseño y Explotación de Almacenes de Datos.