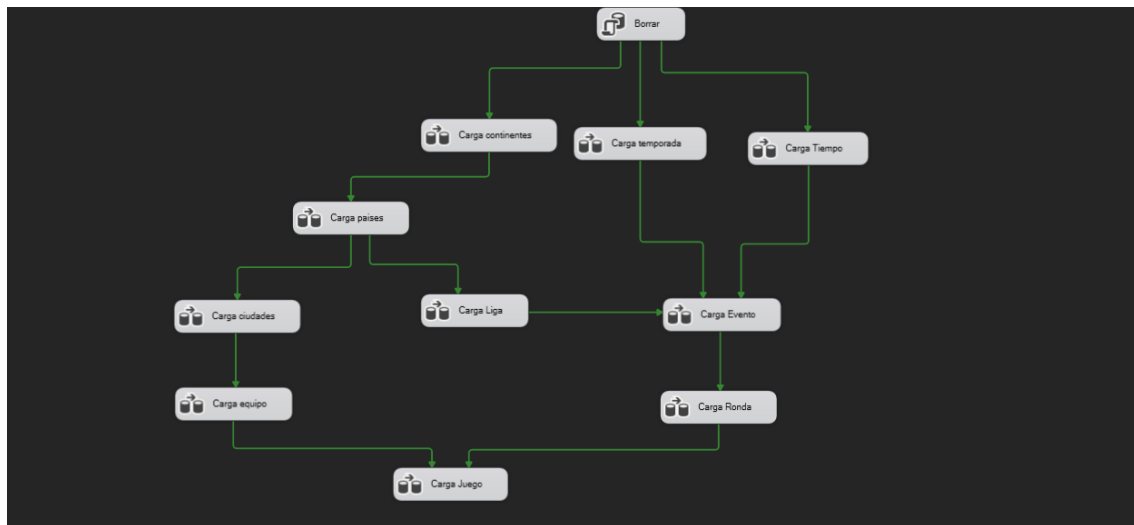
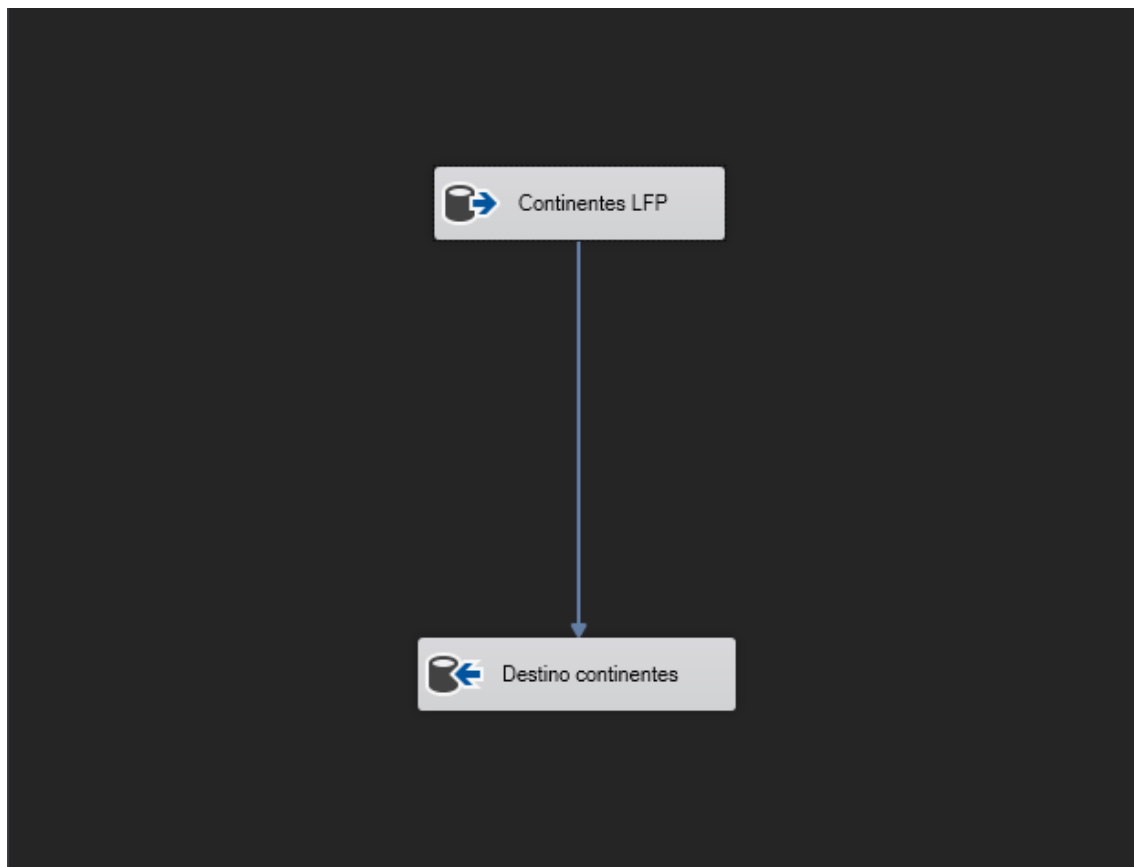


## Diseño y Explotación de Almacenes de Datos

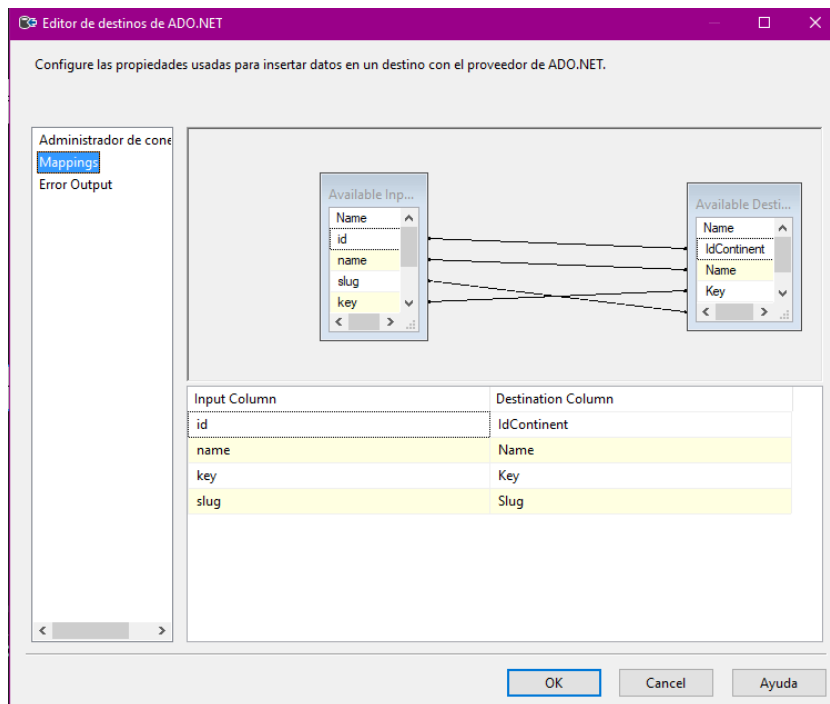
### Práctica 2: ETL



Para esta práctica se nos pedía realizar una integración de datos desde una base de datos a otra vacía. Lo primero que observábamos a añadir al SSMS las dos bases de datos es que las tablas no son al totalmente idénticas.



Como muestro en este ejemplo las tablas que más coinciden no necesitan un conversor de datos ni nada, simplemente crear un origen para ADO NET que es la tabla en la base de datos rellena, y un destino ADO NET y luego hacer el mapeo de los atributos.



En algunos casos si que es necesario añadir un conversor de datos si un atributo de la base de datos origen es diferente que destino, pero es tan sencillo como añadir el conversor seleccionar el atributo que deseamos cambiar y seleccionar el tipo de datos que lo queremos.

La mayor complejidad de esta práctica está relacionada con la tabla tiempo, ya que las otras que dependen de ella, hay añadir nuevos métodos para poder completarla.



Data Conversion Transformation Editor

Configure the properties used to convert the data type of an input column to a different data type. Depending on the data type to which the column is converted, set the length, precision, scale, and code page of the column.

Available Input C...

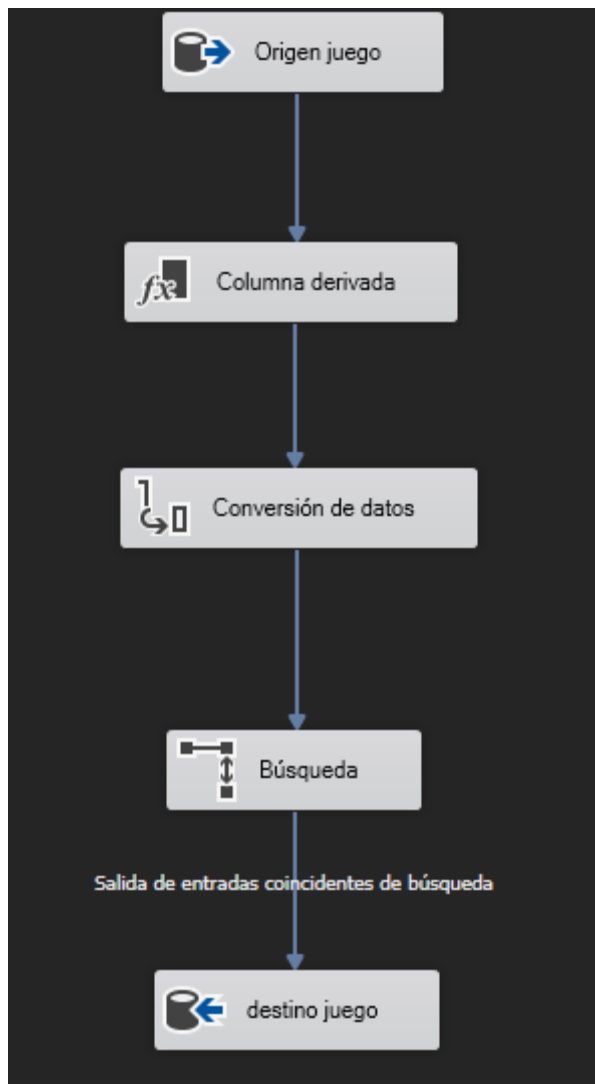
- ☒ Name
- ☒ Year
- ☒ Month

Input Column	Output Alias	Data Type	Length	Precision	Scale	Code Page
Year	Copy of Year	decimal [DT_DECIMAL]			0	
Month	Copy of Month	decimal [DT_DECIMAL]			0	
Day	Copy of Day	decimal [DT_DECIMAL]			0	

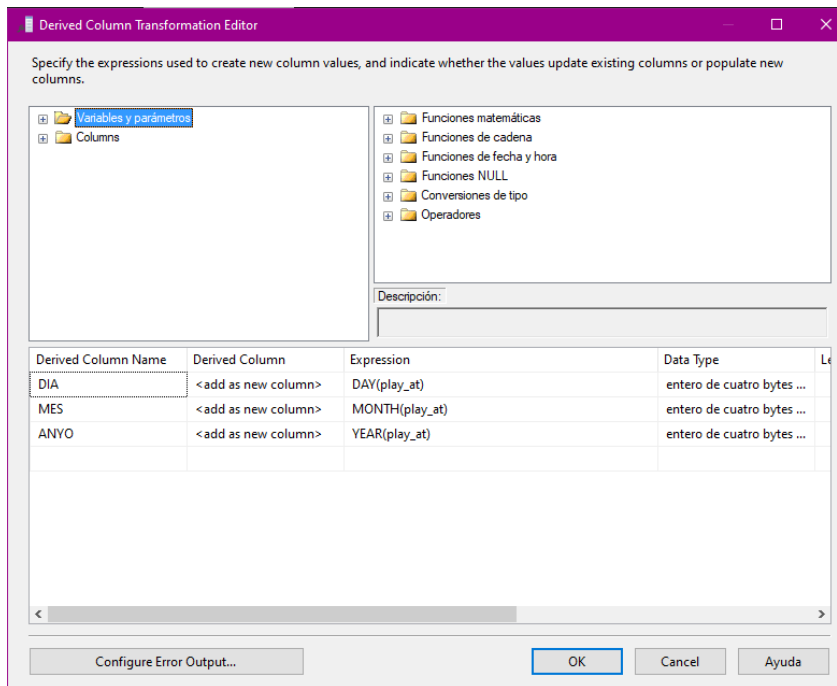
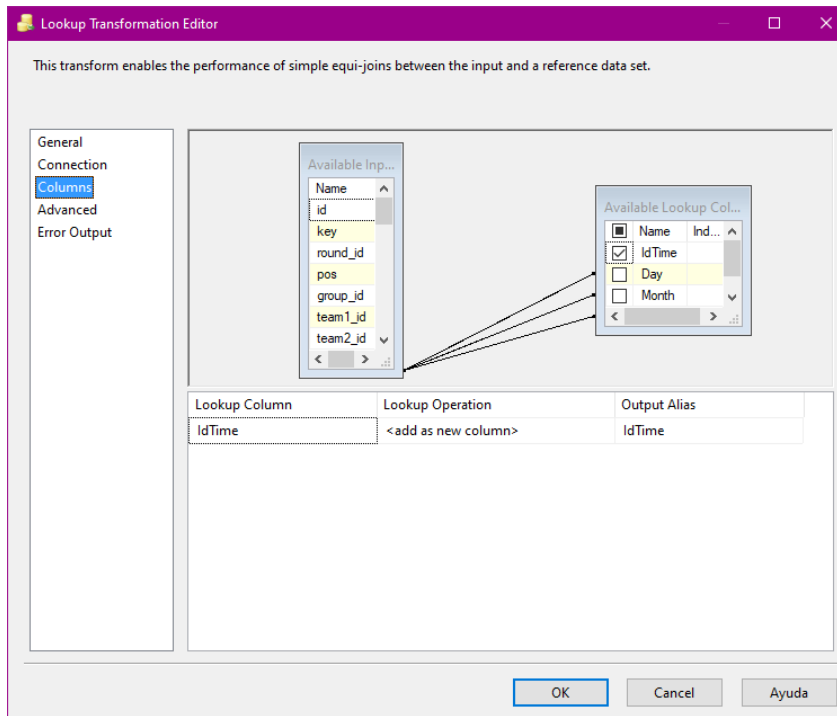
Configure Error Output... OK Cancel Ayuda

Como vemos en esta tabla los datos no derivan de la base de datos rellena, si no de un archivo Excel, el cual tiene los atributos día, mes, y año. Y para poder añadir estos valores a la tabla tiempo, necesitamos convertir el tipo de datos, para que no haya pérdidas a la hora de rellenar las columnas.

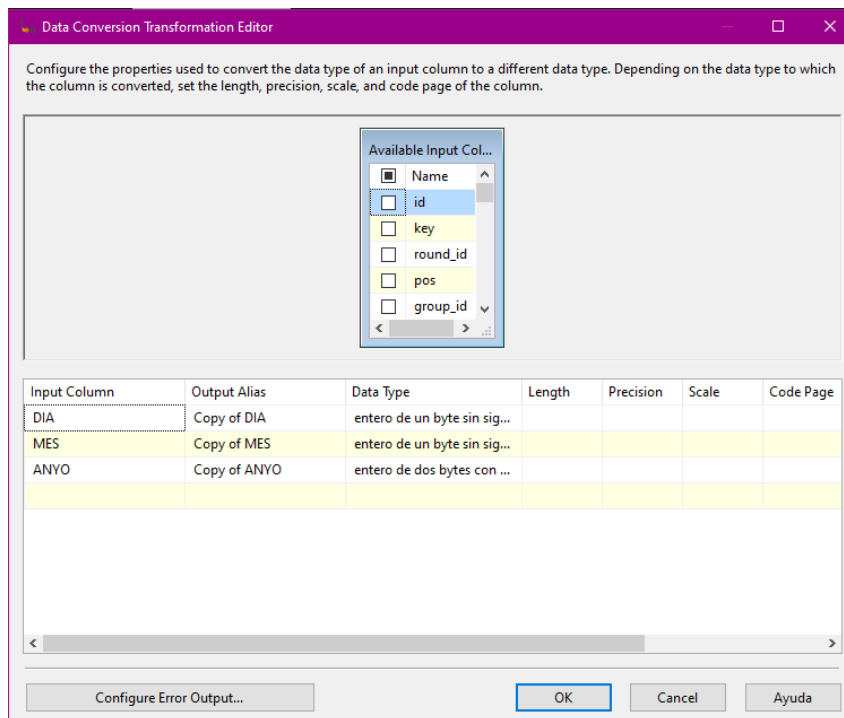
Un buen ejemplo de los procedimientos a añadir en las tablas que derivan de la tabla tiempo es el hecho Game.



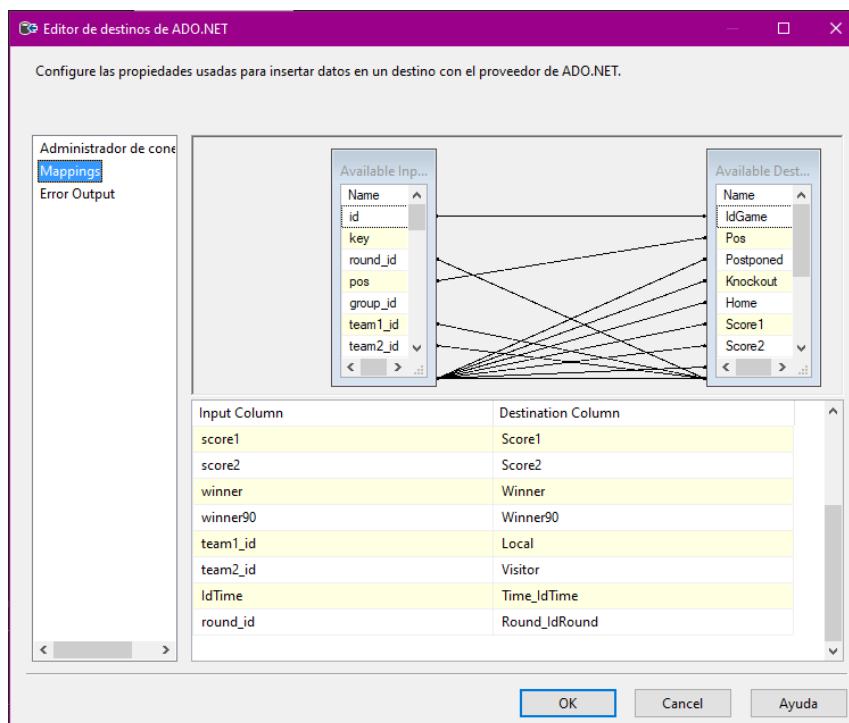
Como vemos, si que debemos añadir el origen ADO NET como en las tablas anteriores, pero aquí debemos añadir un procedimiento que nos haga modificar el atributo de tipo date de la tabla, y nos lo divida en tres columnas que serán en día, mes y año, que los necesitaremos para extraer de la tabla tiempo su PK mediante la búsqueda por valor de atributos, y así añadir el valor de la clave primaria a la columna correspondiente.



Pero primero una vez obtenido el día, mes y año, convertimos su tipo de datos para que coincida con el tipo de dato que tiene la tabla tiempo para sus atributos, tras esto, añadimos el procedimiento de búsqueda que nos extrae el idTime, a través del día, mes y año.



Para finalizar realizamos un mapping de las columnas que nos interesan añadir y ya tendría el ETL del almacén completamente funcional.



También al principio de la integración de datos he añadido un procedimiento SQL que nos sirva de borrado general de todas las tablas.

```
USE [LFP2]
GO
/***** Object: StoredProcedure [dbo].[borrar_lugares]    Script Date: 04/12/2022 21:51:50 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:      <Author,,Name>
-- Create date: <Create Date,,>
-- Description: <Description,,>
-- =====
ALTER PROCEDURE [dbo].[borrar_lugares]
AS
BEGIN
    delete from dbo.Game
    delete from dbo.Round
    delete from dbo.Event
    delete from dbo.League
    delete from dbo.Team
    DELETE FROM dbo.City
    DELETE FROM dbo.Country
    DELETE FROM dbo.Continent
    delete from dbo.Season
    delete from dbo.Time
END
```