
Práctica-4: Herramientas ETL SSIS (SQL Server Integration Services).



**UNIVERSIDAD
DE GRANADA**

Sistemas Multidimensionales (2019-2020)

Daniel Bolaños Martínez
danibolanos@correo.ugr.es
Grupo 2 - Viernes 15:30h

Índice

1. Crea una BD SQL Server cuyo nombre sea prueba y tu nombre de usuario de correo UGR. Importa las hojas Cuándo, Dónde y Padrón del archivo obtenido con Power Query. En los pasos de importación, selecciona la opción 'Save SSIS Package' y 'File system' para generar un paquete de SSIS en forma de archivo. 3
2. Crea un proyecto SSIS en SSDT y una solución cuyos nombres sean prueba y tu nombre de usuario de correo UGR, importa el paquete generado en el punto 1 y ábrelo. 16
3. Crea una BD SQL Server cuyo nombre sea el nombre de la provincia que tienes asignada y tu nombre de usuario de correo UGR. Crea las tablas cuando, donde y padron añadiendo al nombre el sufijo de tu nombre de usuario. La estructura de estas tablas ha de ser similar a la de las hojas correspondientes del archivo obtenido con Power Query, usa el criterio camel_case. Para cada tabla define un flujo de datos específico. 20
4. Bibliografía. 31

1. Crea una BD SQL Server cuyo nombre sea prueba y tu nombre de usuario de correo UGR. Importa las hojas Cuándo, Dónde y Padrón del archivo obtenido con Power Query. En los pasos de importación, selecciona la opción 'Save SSIS Package' y 'File system' para generar un paquete de SSIS en forma de archivo.

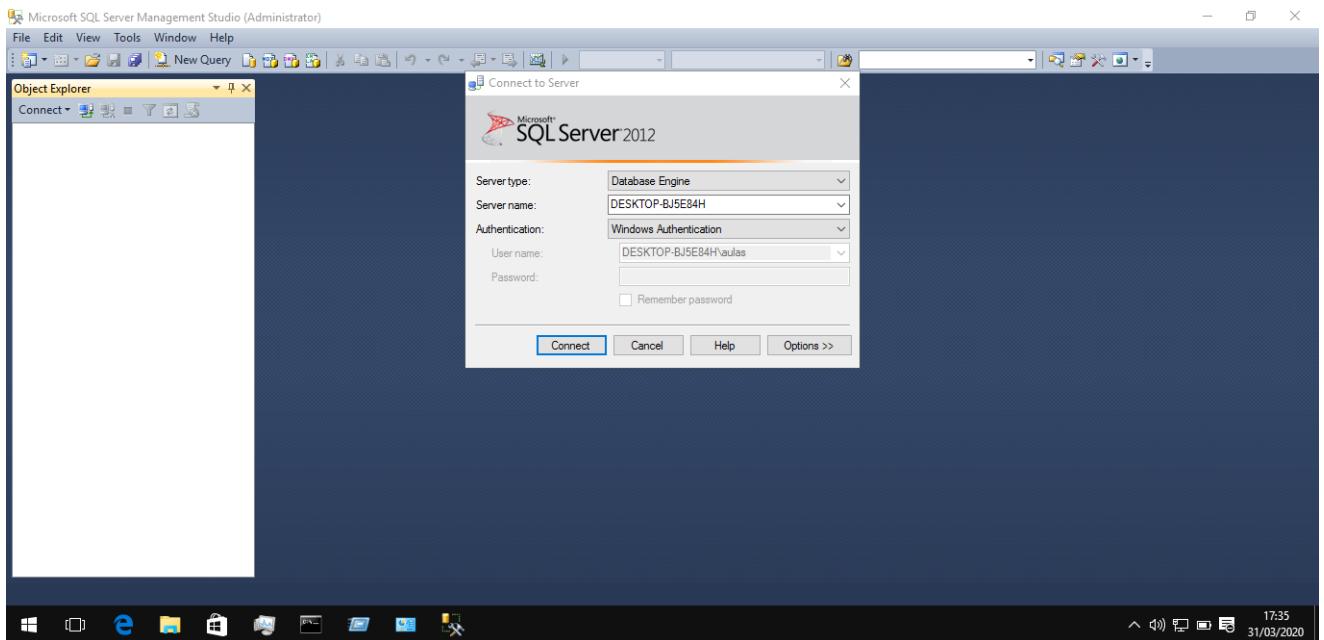


Figura 1: Conectamos con el servidor.

Creamos una nueva base de datos clickando en el apartado *Databases* en la columna *Object Explorer*. La base de datos se llamará **prueba_danibolanos**.

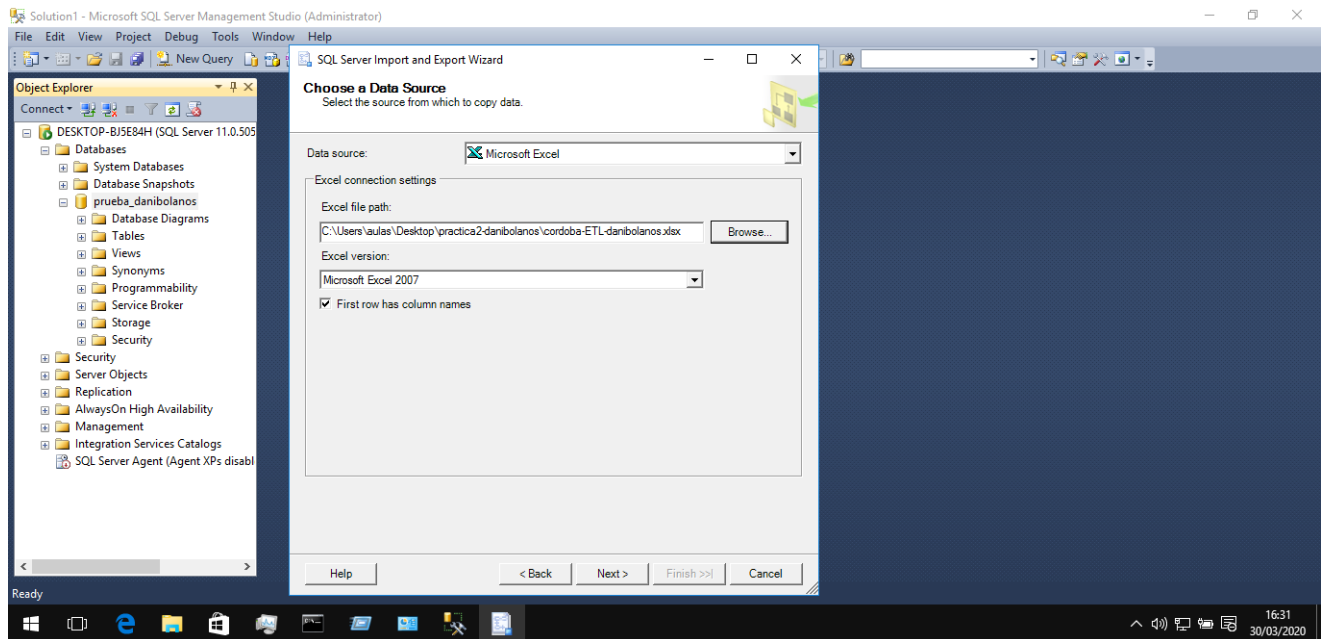


Figura 2: Definimos como fuente de datos el archivo *cordoba-ETL-danibolanos.xlsx*.

Accederemos a la ventana a partir del menú contextual de la base de datos **prueba_danibolanos** pulsando sobre la opción [«Tasks», «Import Data»].

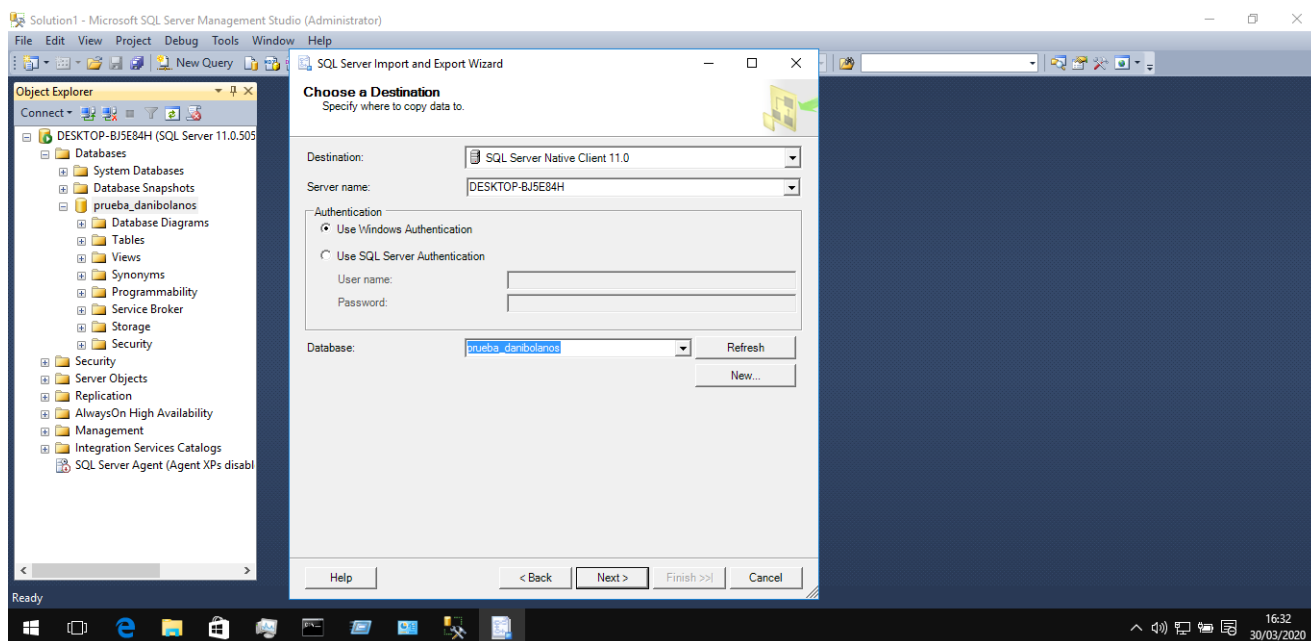


Figura 3: Definimos el destino de los datos.

Nos aseguramos que esté seleccionada la opción correspondiente a «SQL Server Native Client 11.0» y el nombre del servidor sea el correcto.

Indicamos la forma de autenticación para nuestra instalación («Use Windows Authentication») y seleccionamos la base de datos **prueba_danibolanos**.

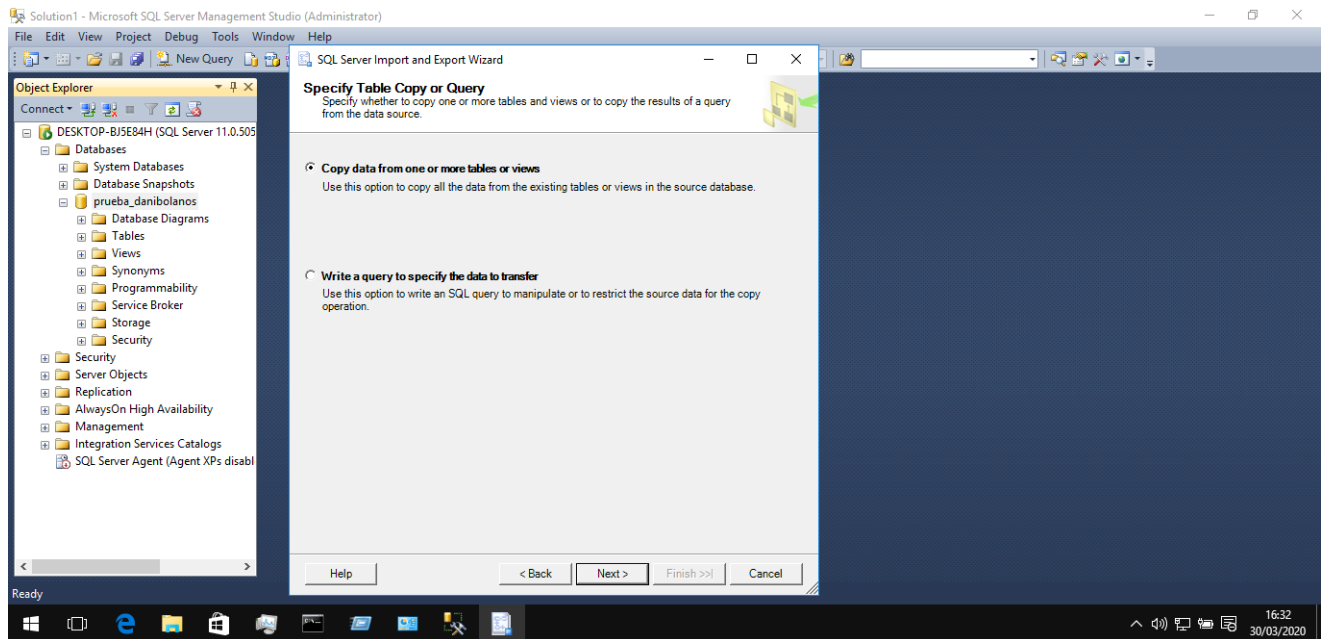


Figura 4: Copiamos todos los datos disponibles en las hojas.

Queremos importar todos los datos disponibles, por lo que seleccionamos la opción «Copy data from one or more tables or views».

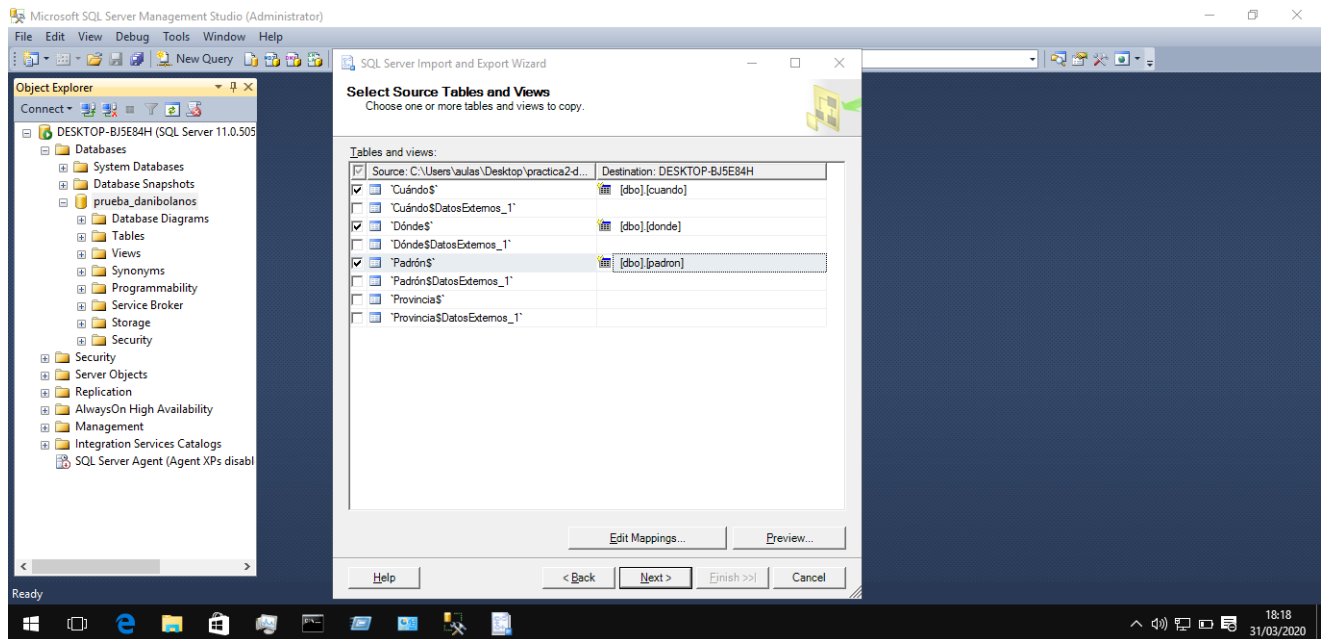


Figura 5: Correspondencia entre las hojas y las tablas.

Definimos la correspondencia entre las hojas y las tablas de la base de datos utilizando el criterio **camel_case** especificado en el enunciado para cada campo de las tablas de datos.

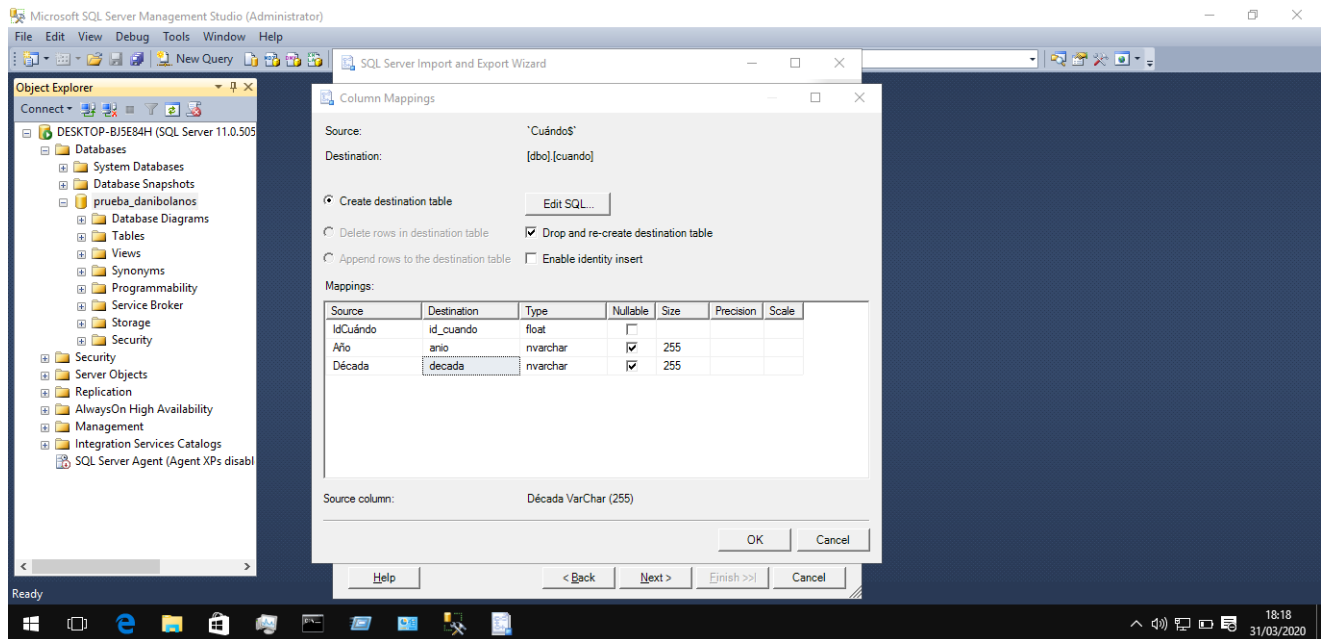


Figura 6: Correspondencia entre columnas y campos para **cuando**.

A continuación se muestran los ejemplos de como modificamos los campos para cada tabla de datos del archivo original.

Column Mappings

Source: 'Dónde\$'

Destination: [dbo].[donde]

☒ Create destination table

Edit SQL...

☐ Delete rows in destination table
☒ Drop and re-create destination table

☐ Append rows to the destination table
☐ Enable identity insert

Mappings:

Source	Destination	Type	Nullable	Size	Precision	Scale
IdDónde	id_donde	float	<input type="checkbox"/>			
CodMunicipio	cod_municipio	nvarchar	<input checked="" type="checkbox"/>	255		
Municipio	municipio	nvarchar	<input checked="" type="checkbox"/>	255		
CapitalMunicipio	capital_municipio	nvarchar	<input checked="" type="checkbox"/>	255		
CodComarca	cod_comarca	float	<input checked="" type="checkbox"/>			
Comarca	comarca	nvarchar	<input checked="" type="checkbox"/>	255		
CodProvincia	cod_provincia	nvarchar	<input checked="" type="checkbox"/>	255		
Provincia	provincia	nvarchar	<input checked="" type="checkbox"/>	255		
CapitalProvincia	capital_provincia	nvarchar	<input checked="" type="checkbox"/>	255		

Source column: Altitud_m VarChar (255)

OK

Cancel

Column Mappings

Source: 'Padrón\$'

Destination: [dbo].[padron]

☒ Create destination table

Edit SQL...

☐ Delete rows in destination table
☒ Drop and re-create destination table

☐ Append rows to the destination table
☐ Enable identity insert

Mappings:

Source	Destination	Type	Nullable	Size	Precision	Scale
IdCuándo	id_cuando	float	<input type="checkbox"/>			
IdDónde	id_donde	float	<input type="checkbox"/>			
Hombres	hombres	float	<input checked="" type="checkbox"/>			
Mujeres	mujeres	float	<input checked="" type="checkbox"/>			
Superficie	superficie	float	<input checked="" type="checkbox"/>			

Source column: Superficie Double (15)

OK

Cancel

Figura 7: Correspondencia entre columnas y campos para **donde** y **padron**.

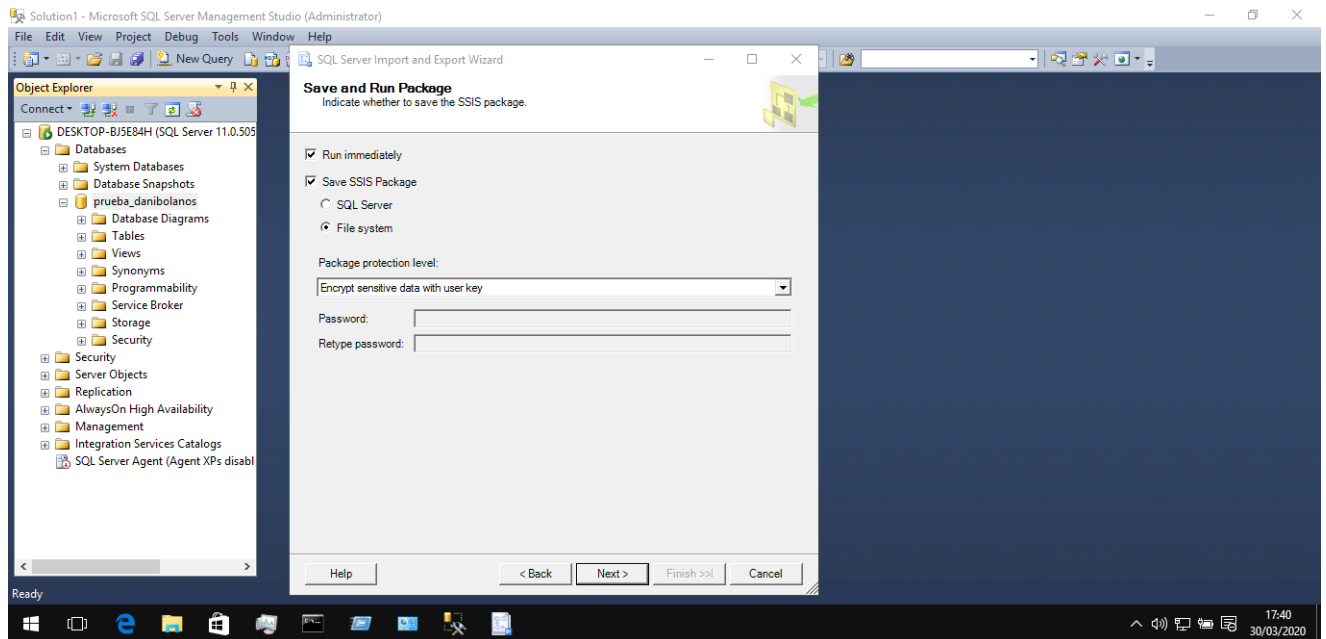


Figura 8: Ejecutamos y guardamos la transformación.

Almacenaremos el paquete SSIS como un archivo seleccionando la opción «File system». También permite encriptar las claves que se usen, en este caso, dejaremos la opción por defecto.

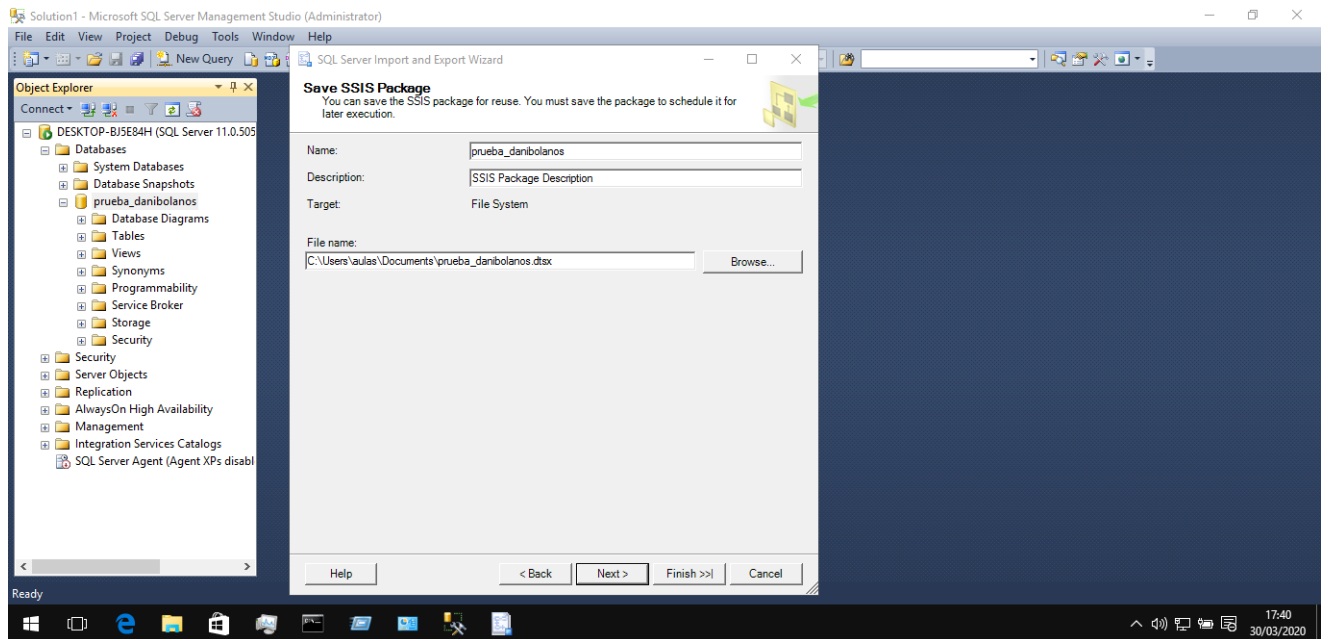


Figura 9: Guardamos la transformación como archivo *prueba_danibolanos.dtsx*.

Daremos un nombre al archivo donde se guardará el paquete e indicamos en qué carpeta se almacenará.

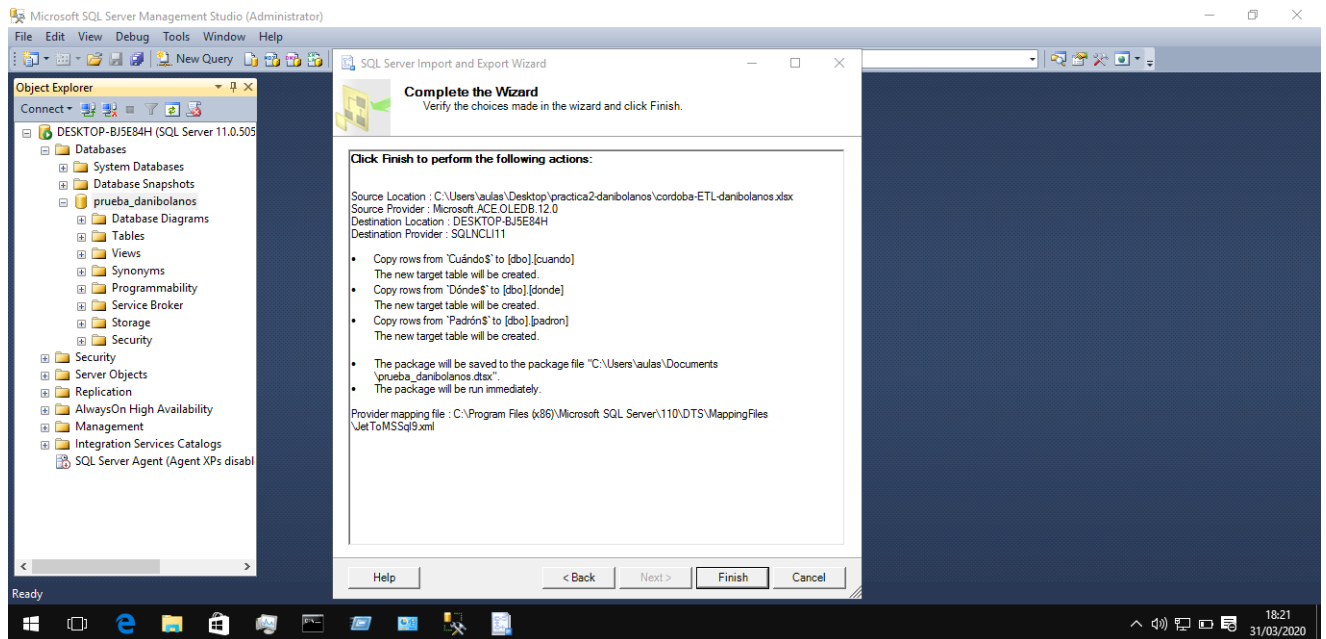


Figura 10: Operaciones a realizar para importar los datos.

Por último, nos informa de las operaciones que va a realizar para generar el paquete SSIS a partir de los datos importados.

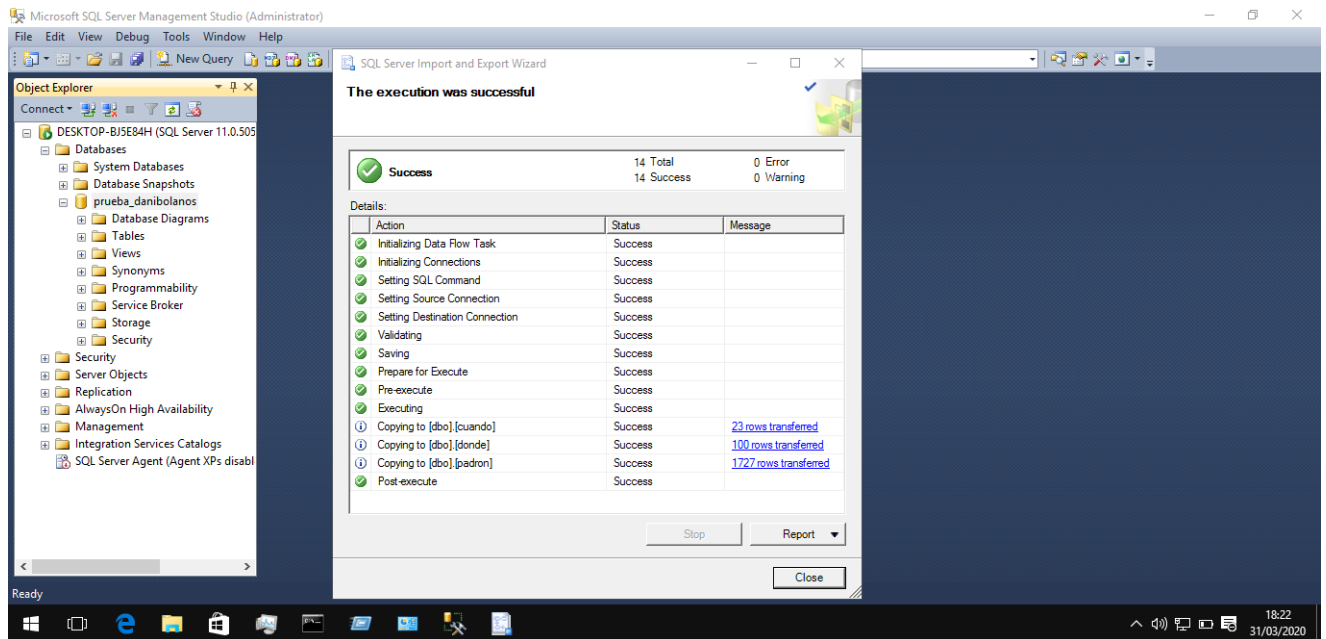


Figura 11: Resultado al ejecutar las operaciones anteriores.

Podemos ver como la ejecución ha tenido éxito. Esta ejecución dará como resultado el paquete SSIS como archivo *.dtsx*.

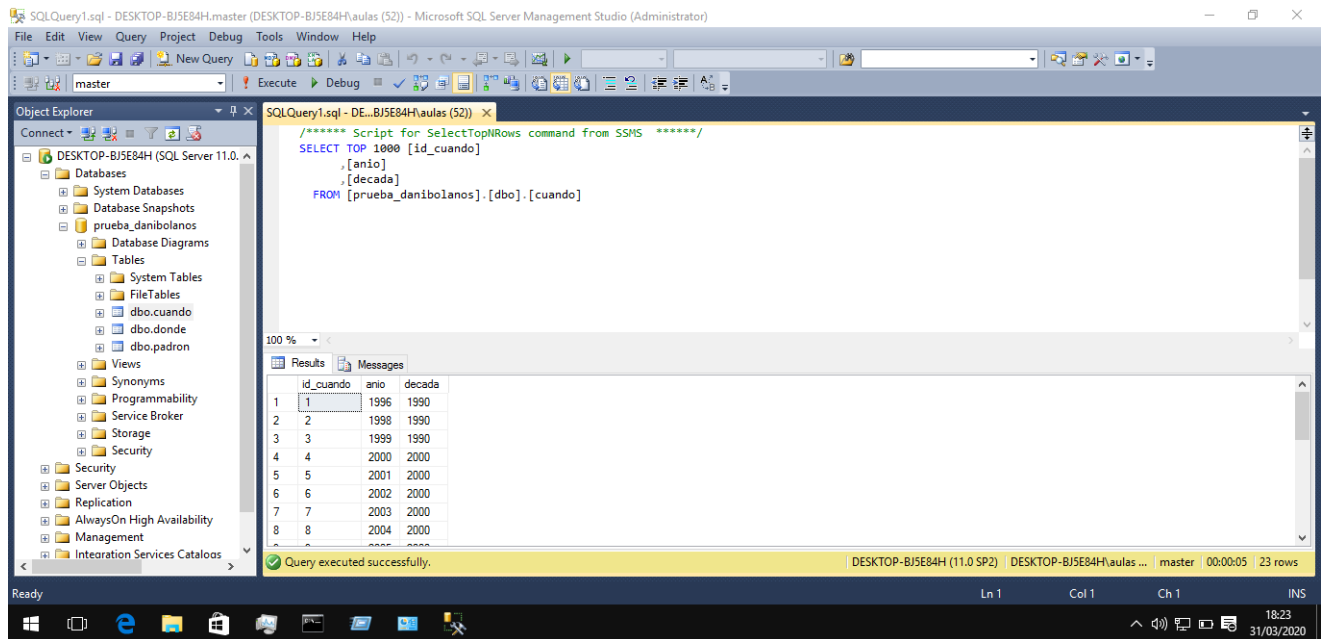


Figura 12: Datos **cuando** almacenados en la BD de *SQL Server*.

Podemos observar los datos del archivo *cordoba-ETL-danibolanos.xlsx* almacenados correctamente en la base de datos que hemos creado en *SQL Server*.

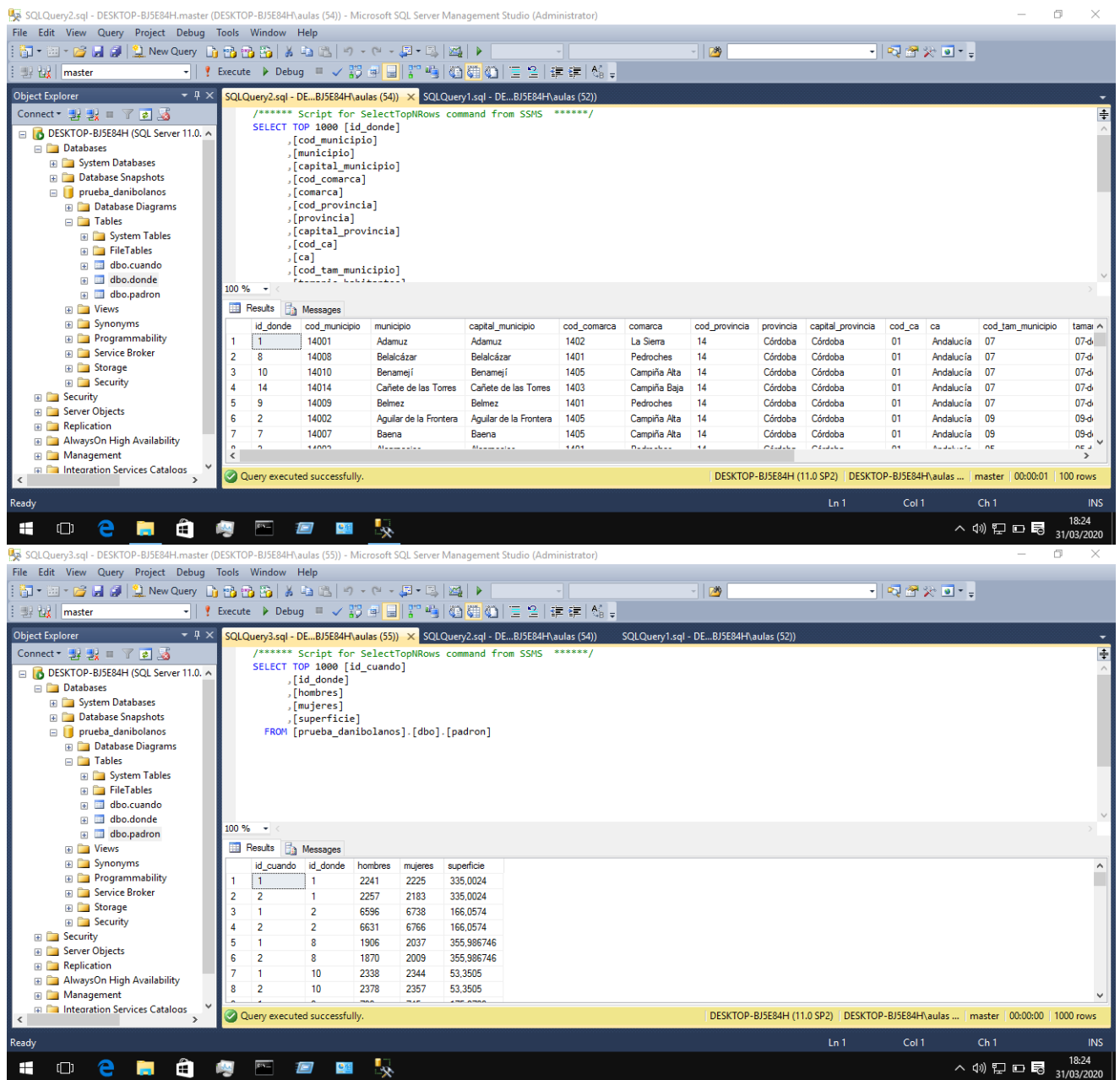


Figura 13: Datos **donde** y **padron** almacenados en la BD de *SQL Server*.

2. Crea un proyecto SSIS en SSDT y una solución cuyos nombres sean prueba y tu nombre de usuario de correo UGR, importa el paquete generado en el punto 1 y ábrelo.

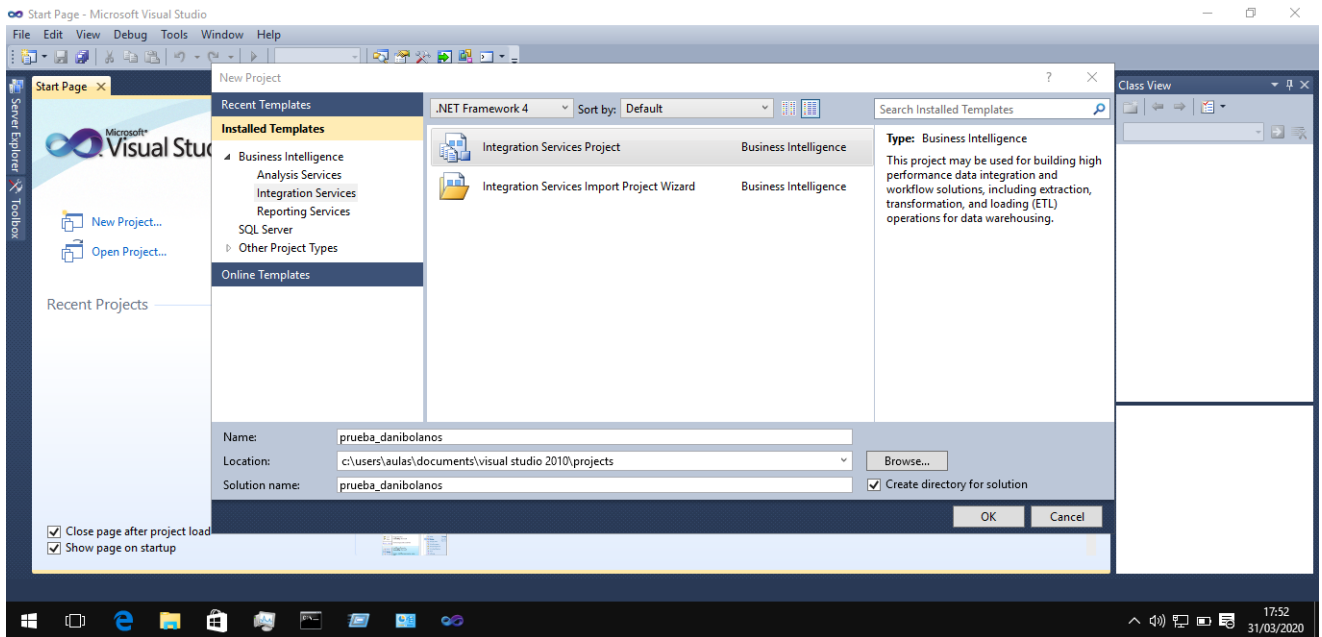


Figura 14: Creación de un proyecto en SSDT.

Pulsamos sobre sobre [«File», «New», «Project»] y a continuación, pulsamos directamente sobre «Integration Services Project».

Llamaremos a este proyecto **prueba_danibolanos** al igual que el paquete SSIS que hemos creado antes.

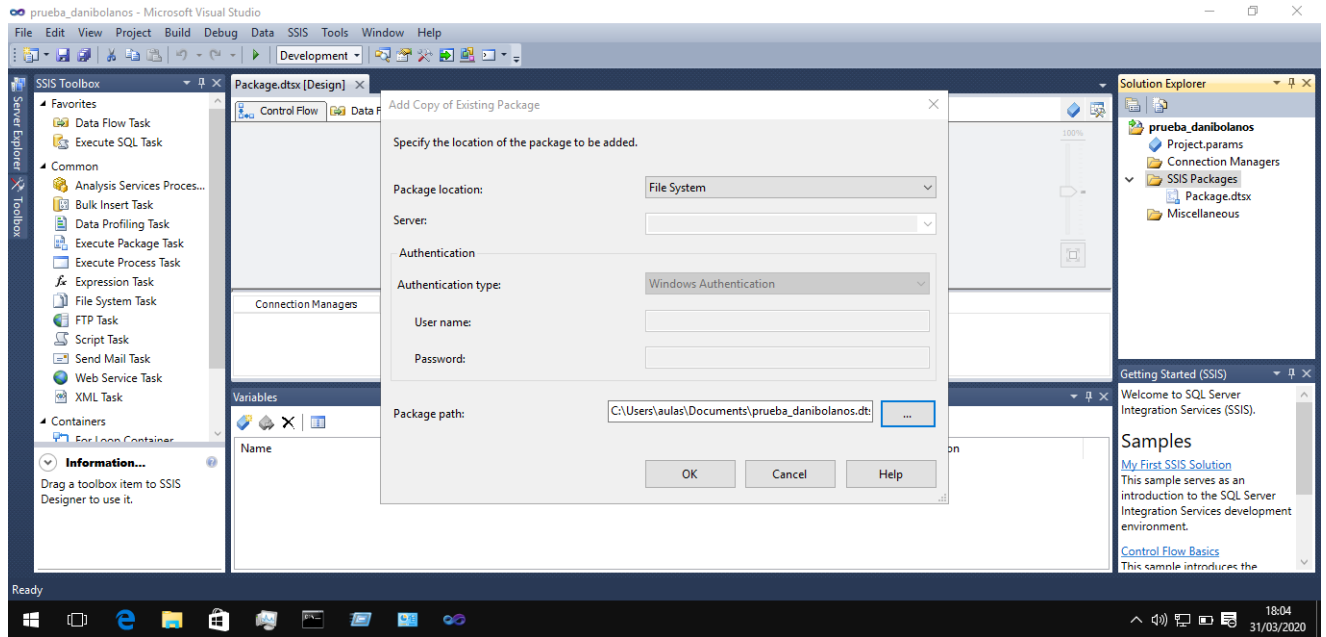


Figura 15: Seleccionamos del paquete *prueba_danibolanos.dtsx*.

Seleccionamos el archivo que contiene el paquete SSIS generado en el apartado anterior para añadirlo al entorno de trabajo de SSDT.

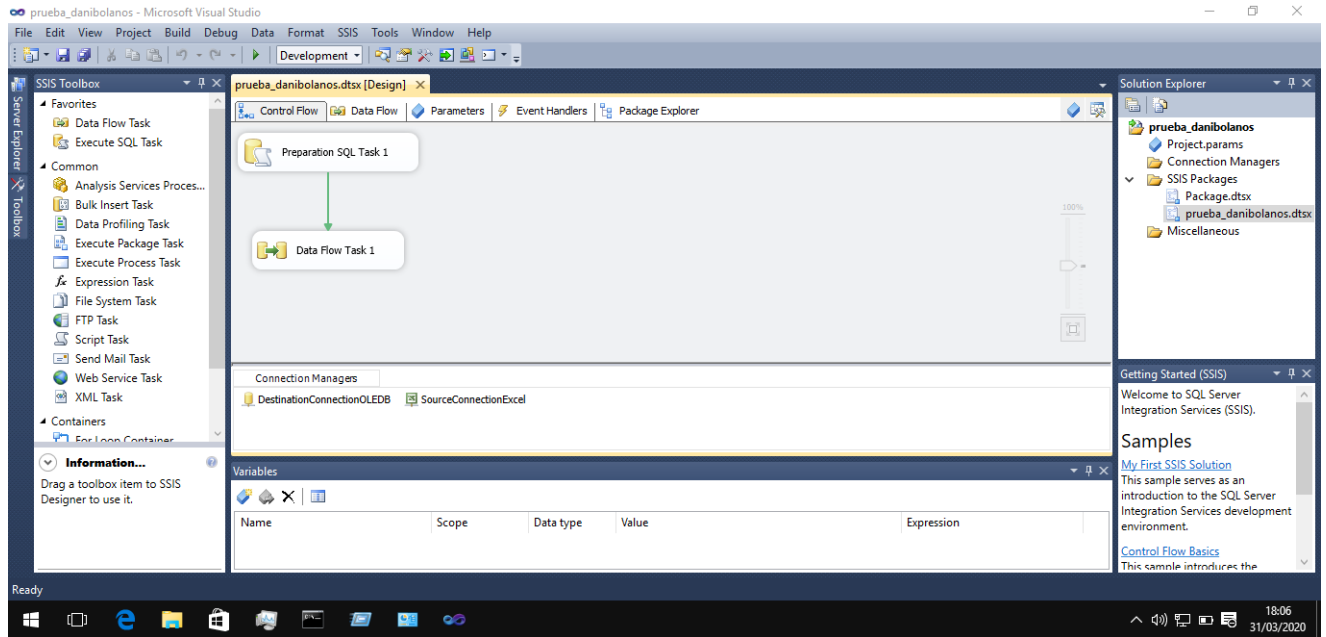


Figura 16: Flujo de control del paquete *prueba_danibolanos.dtsx*.

Finalmente, una vez abierto, podemos explorar y diseñar el paquete *prueba_danibolanos.dtsx*.

La pestaña «Control Flow» nos permite observar el flujo de control del paquete cargado.

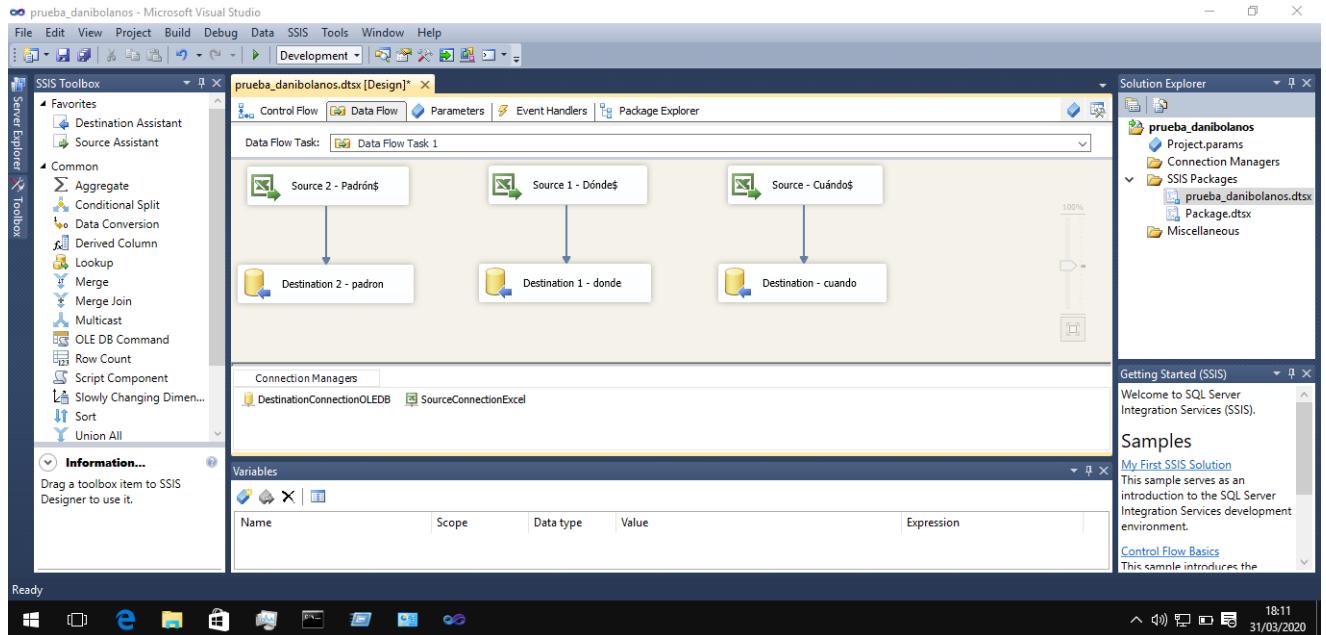


Figura 17: Flujo de datos del paquete *prueba_danibolanos.dtsx*.

La pestaña «Data Flow» nos permite observar el flujo de datos del paquete cargado.

3. Crea una BD SQL Server cuyo nombre sea el nombre de la provincia que tienes asignada y tu nombre de usuario de correo UGR. Crea las tablas cuando, donde y padron añadiendo al nombre el sufijo de tu nombre de usuario. La estructura de estas tablas ha de ser similar a la de las hojas correspondientes del archivo obtenido con Power Query, usa el criterio camel_case. Para cada tabla define un flujo de datos específico.

Crearemos una nueva BD en *SQL Server* con nombre **cordoba_danibolanos** e importaremos los datos del archivo *cordoba-ETL-danibolanos.xlsx* tal y como hemos hecho en el ejercicio 1. Mostraremos los pasos clave, ya que salvo el cambio de los nombres de las tablas (**cuando_danibolanos**, **donde_danibolanos** y **padron_danibolanos**) los pasos son equivalentes.

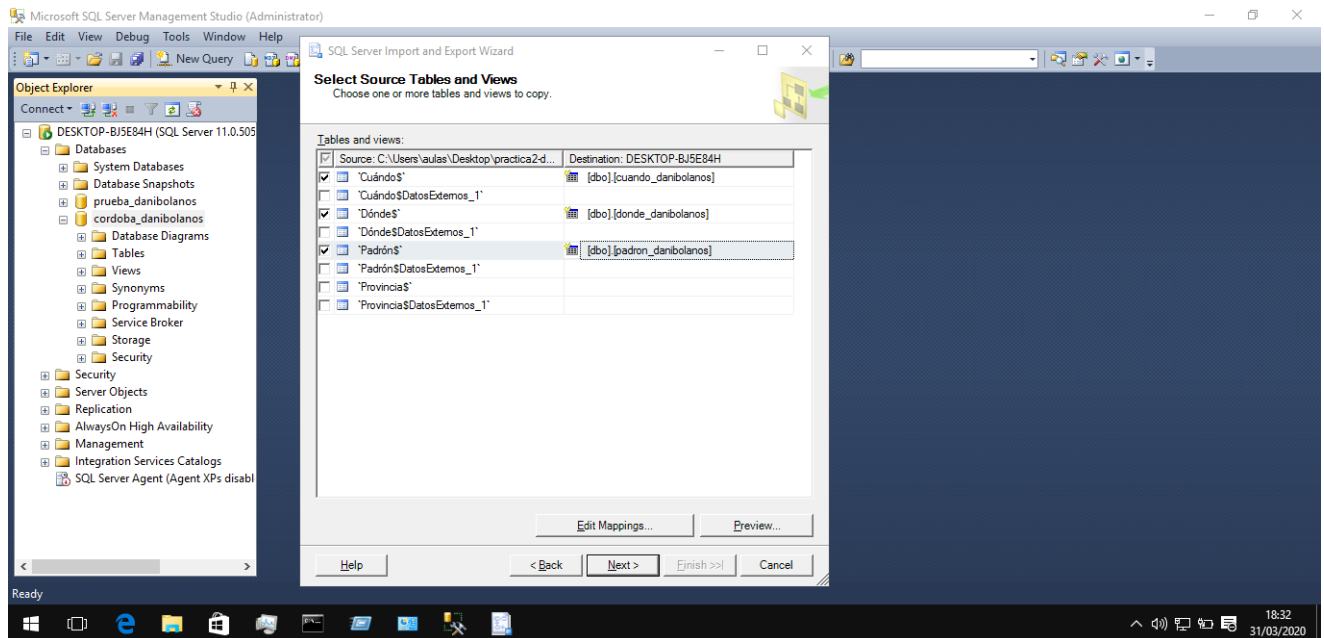


Figura 18: Correspondencia entre las hojas y las tablas.

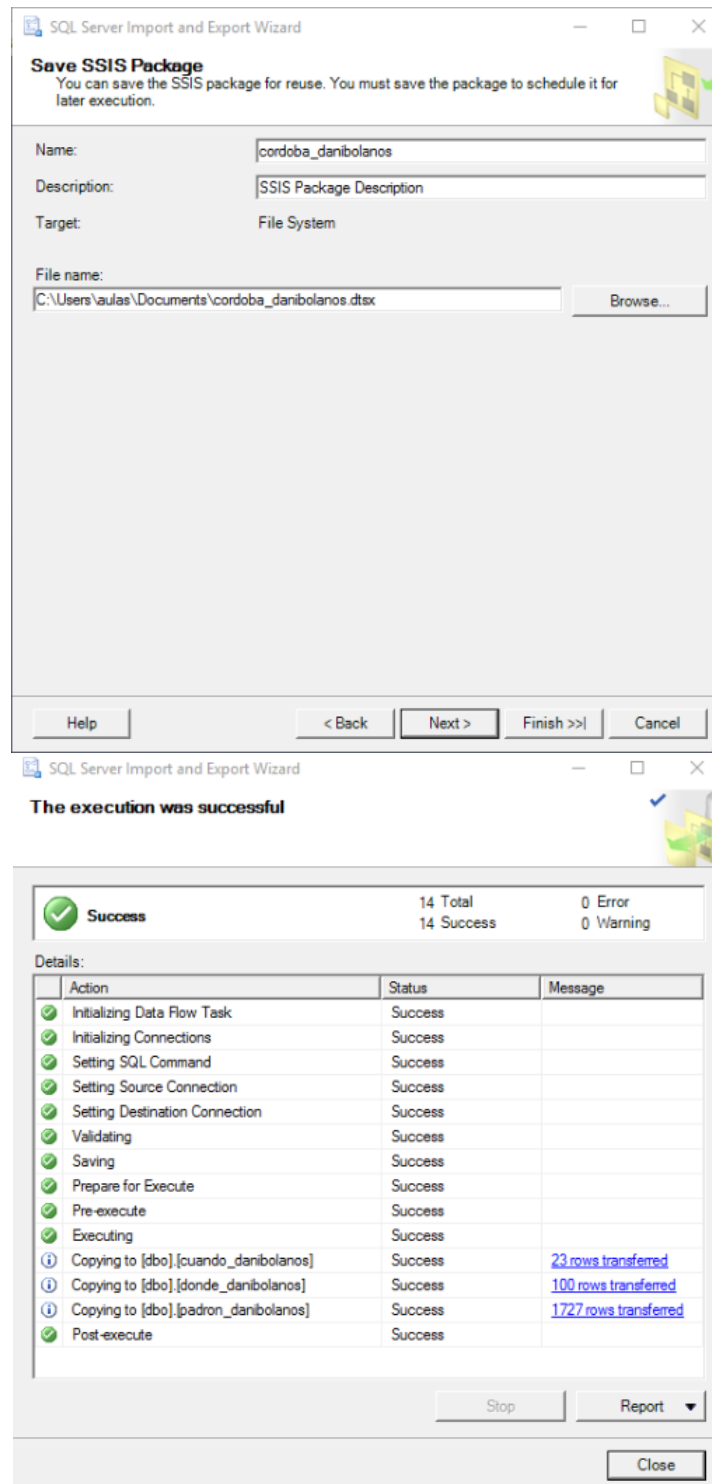


Figura 19: Ejecutamos las operaciones del paquete SSIS.

Una vez que hemos creados la BD en *SQL Server* con el nombre de **cordoba_danibolanos** y hemos guardado el paquete SSIS como archivo *cordoba_danibolanos.dtsx*, nos dispon-dremos a abrir el archivo en el entorno de SSDT.

En esta ocasión llamaremos al proyecto **cordoba_danibolanos**.

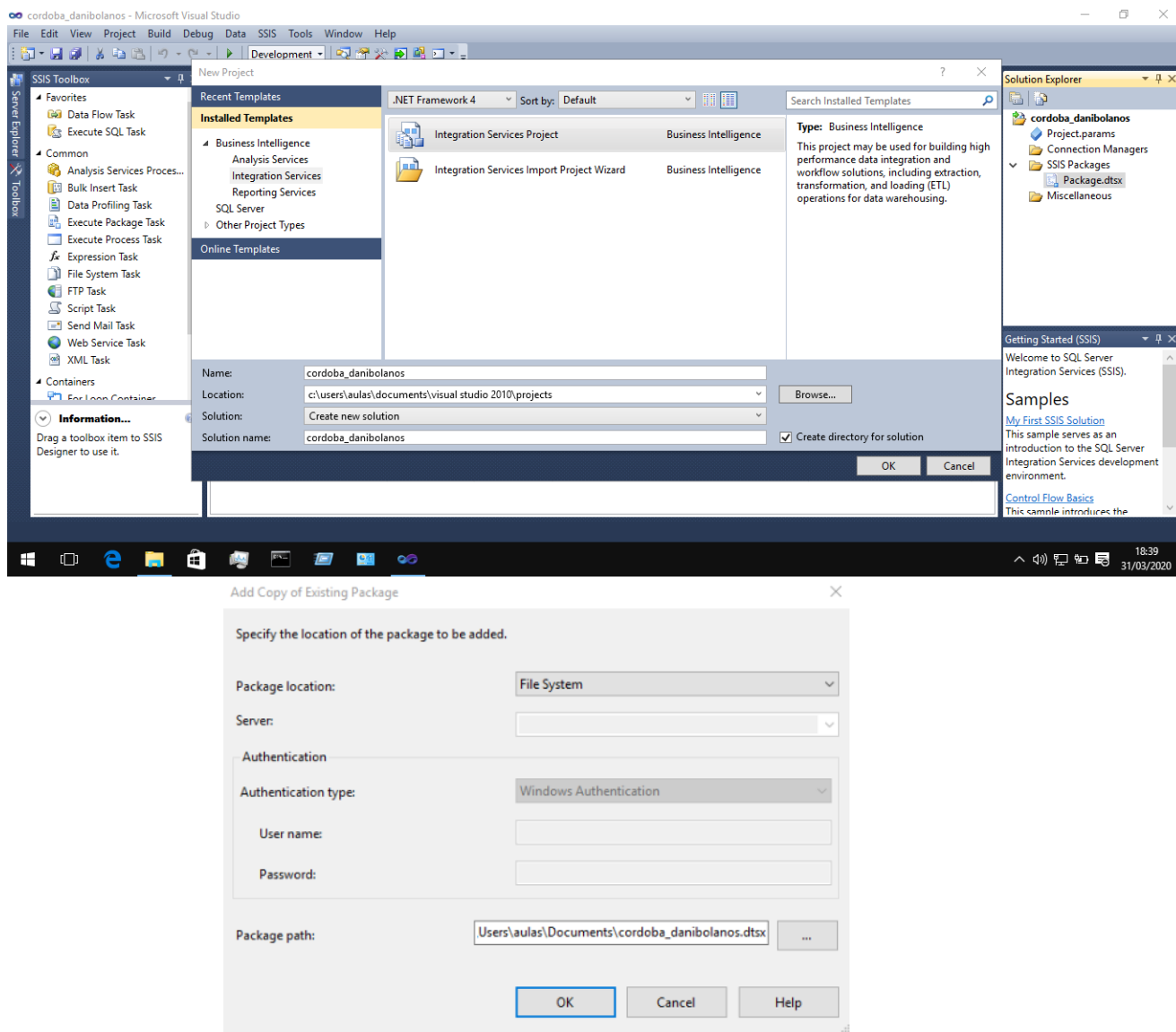


Figura 20: Creamos un proyecto SSDT e importamos el paquete SSIS.

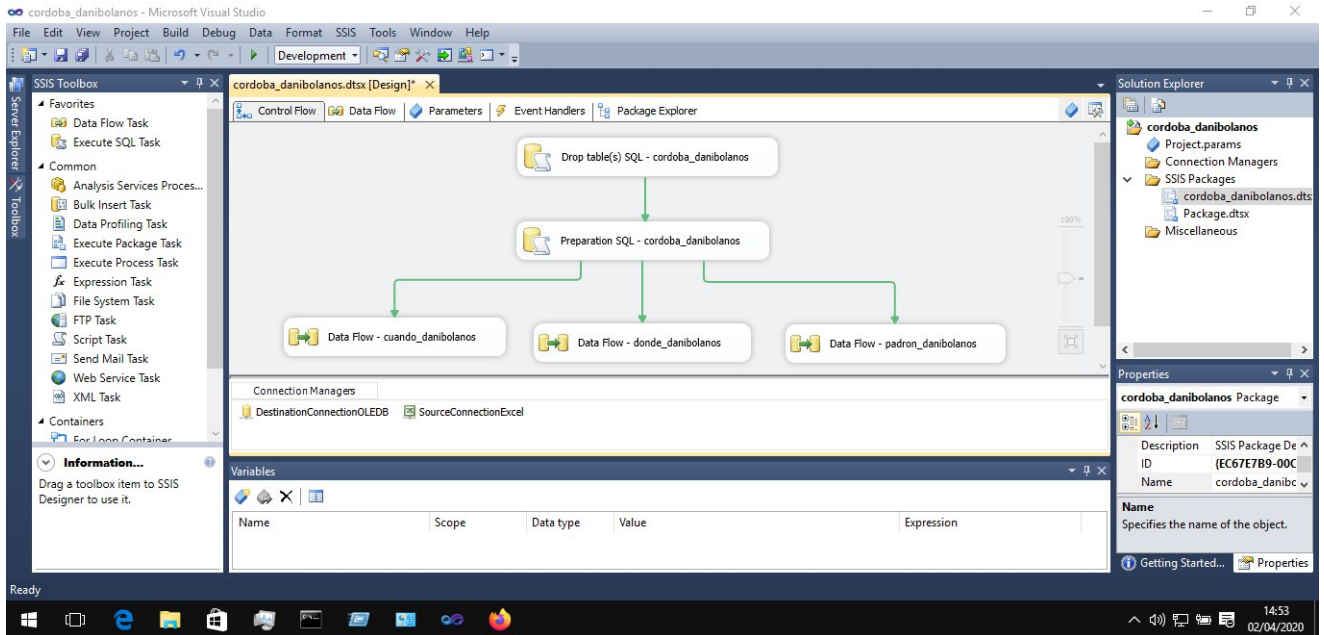


Figura 21: Flujo de control para *cordoba_danibolanos.dtsx*.

Tenemos un flujo de control para nuestro paquete *cordoba_danibolanos.dtsx* a partir del cual se integran tres componentes de flujo de datos (uno para cada tabla), cada uno de ellos se corresponderá con un elemento «Data Flow Task» de la pestaña «Control Flow».

Tal y como se indica en el enunciado renombramos cada tarea para que contenga el nombre de usuario.

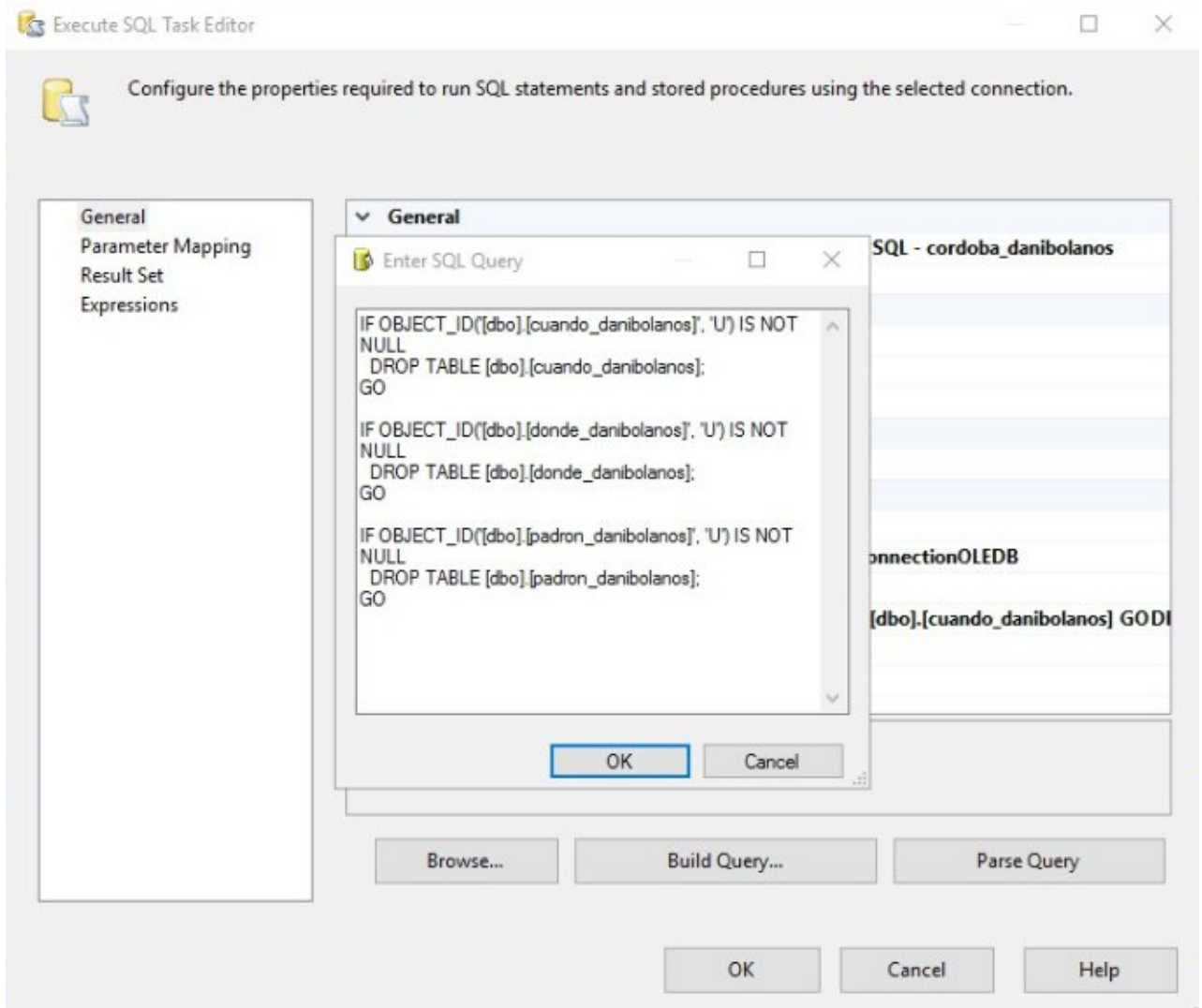


Figura 22: Sentencias *SQL* para borrar las tablas.

Hemos creado un paso previo para eliminar las tablas en el caso de que se hayan generado anteriormente evitando así posibles errores a la hora de ejecutar el programa.^[2]

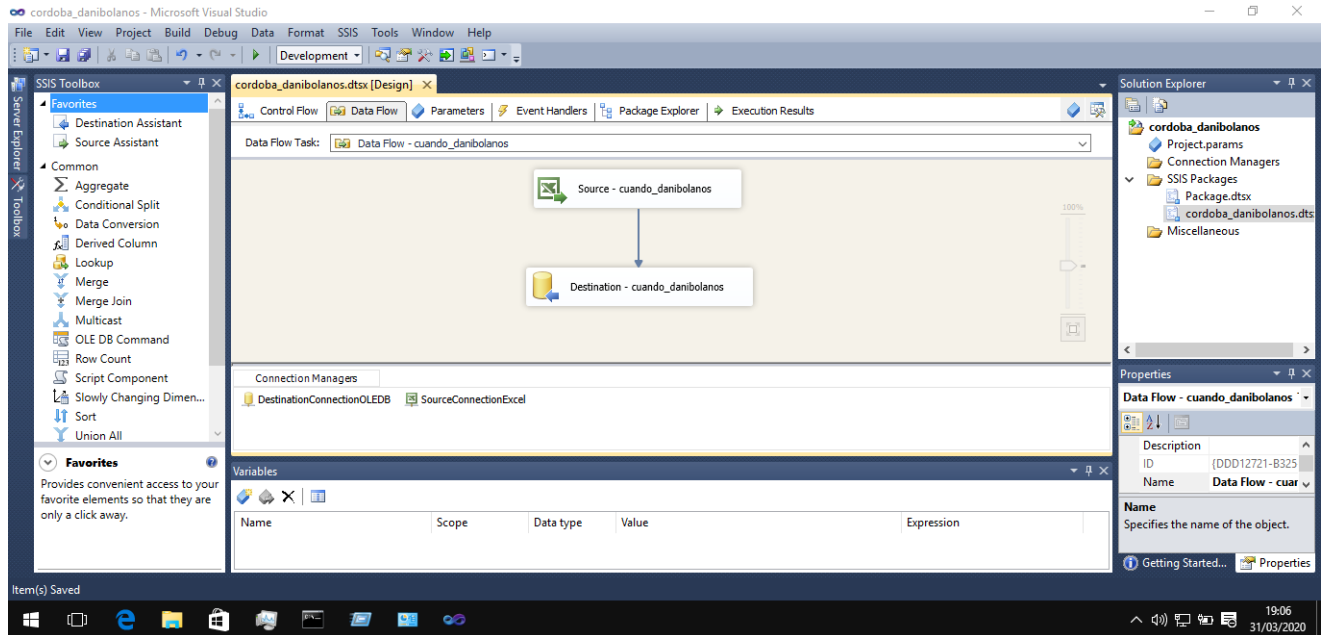


Figura 23: Flujo de datos para la tabla **cuando_danibolanos**.

Cada «Data Flow Task» contiene la fuente y destino de cada tabla definiendo así cada una con un flujo de datos específico.

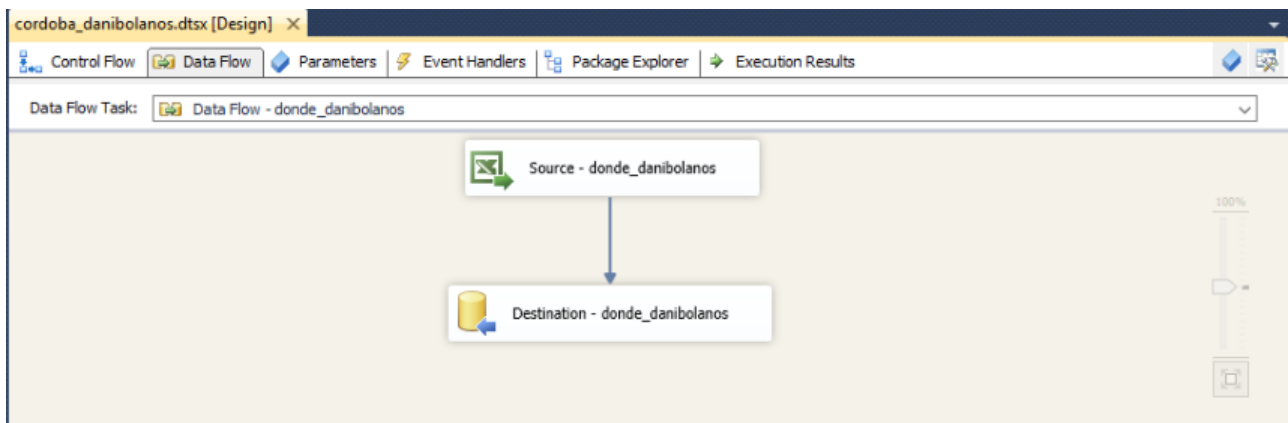


Figura 24: Flujo de datos para **donde_danibolanos**.

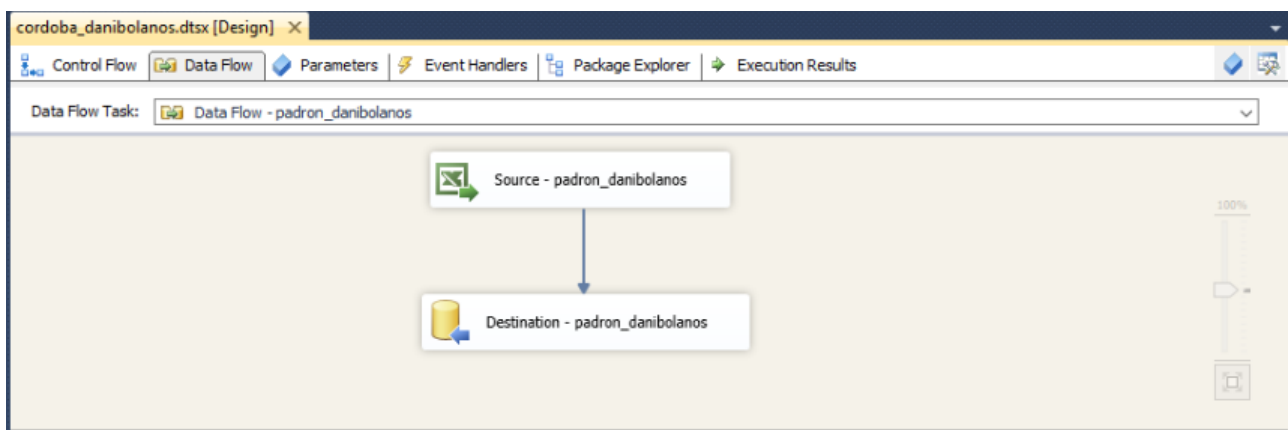


Figura 25: Flujo de datos para **padron_danibolanos**.

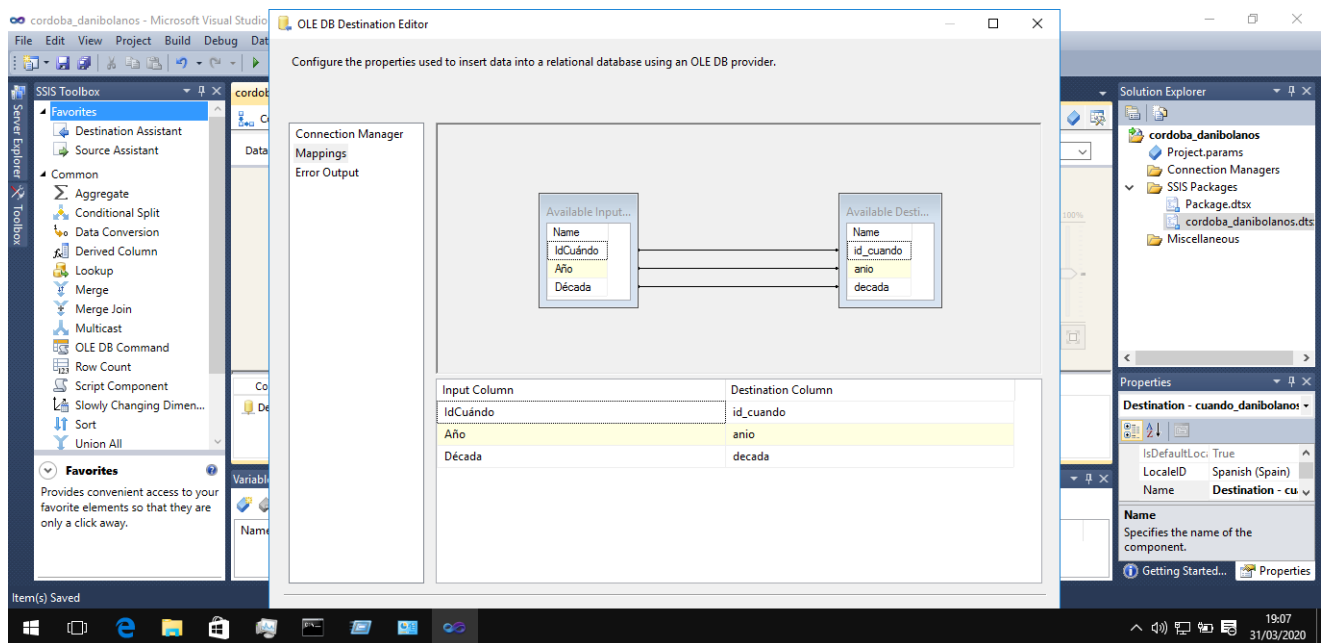


Figura 26: Correspondencias entre fuente y destino para **cuando_danibolanos**.

Podemos ver las correspondencias entre fuente de datos y el destino para cada flujo de datos creado.

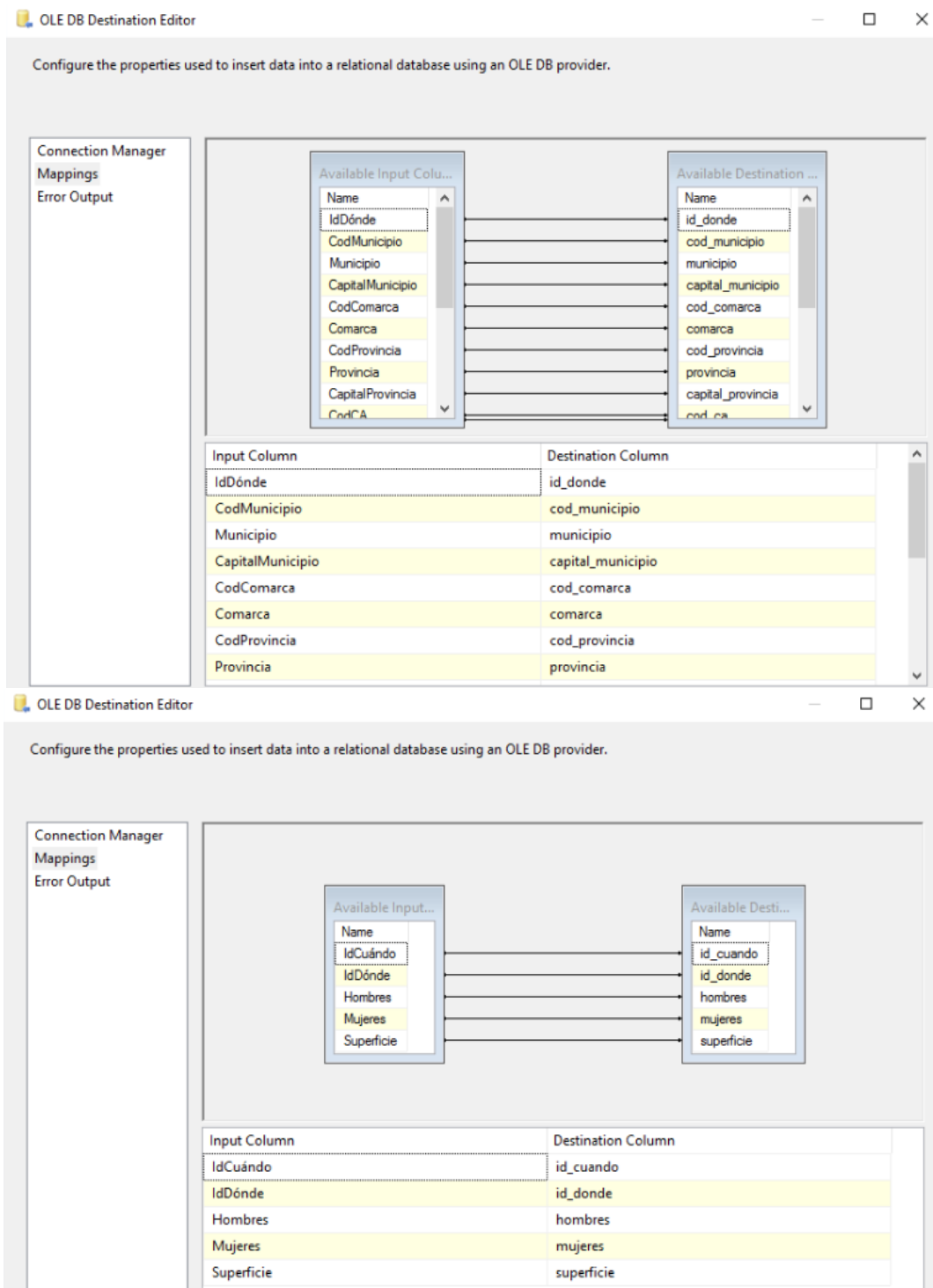


Figura 27: Correspondencias para **donde_danibolanos** y **padron_danibolanos**.

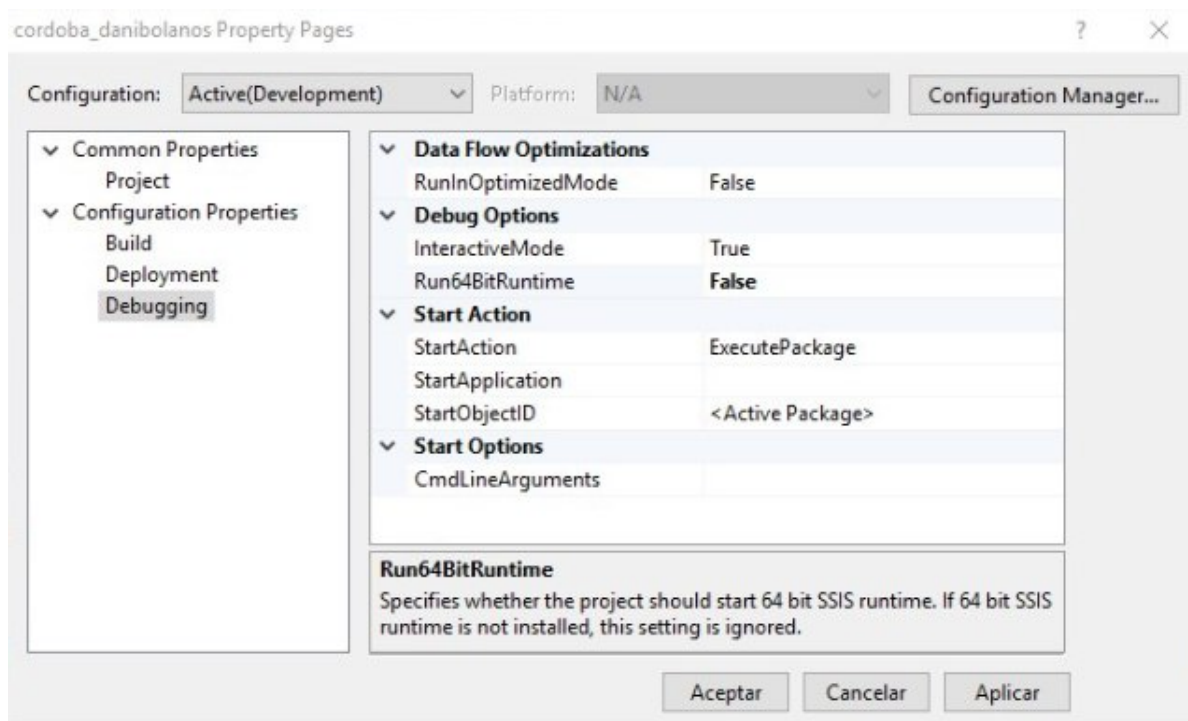


Figura 28: Solución al error de ejecución.

Para solucionar el problema al ejecutar el flujo de control, tenemos que acceder a las propiedades del proyecto **cordoba_danibolanos** y cambiar el valor de la variable «Run64BitRuntime» a «False».

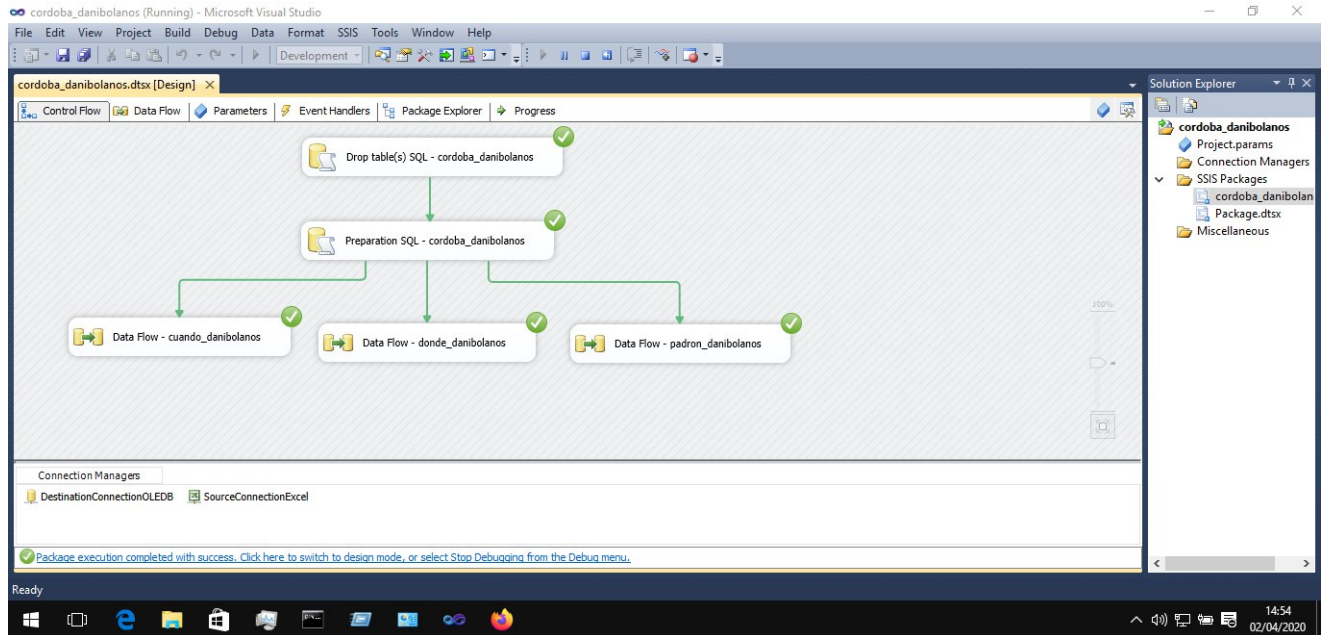


Figura 29: Ejecutamos el programa.

Finalmente podemos ver como el programa se ha ejecutado correctamente y sin errores.

4. Bibliografía.

Referencias

[1] Guion de prácticas de la asignatura.

[2] <https://stackoverflow.com/questions/7887011/how-to-drop-a-table-if-it-exists>