**INSTITUTO TECNOLÓGICO DE COSTA RICA I SEMESTRE 2020**

**INGENIERÍA EN COMPUTACIÓN**

**IC-4301 BASES DE DATOS I**

**Tarea Programada 1 – Partes de automóviles**

Joseph Tenorio Pereira (2019064588)

Jose Pablo Muñoz Montero (2019061904)

1. ***Introducción***

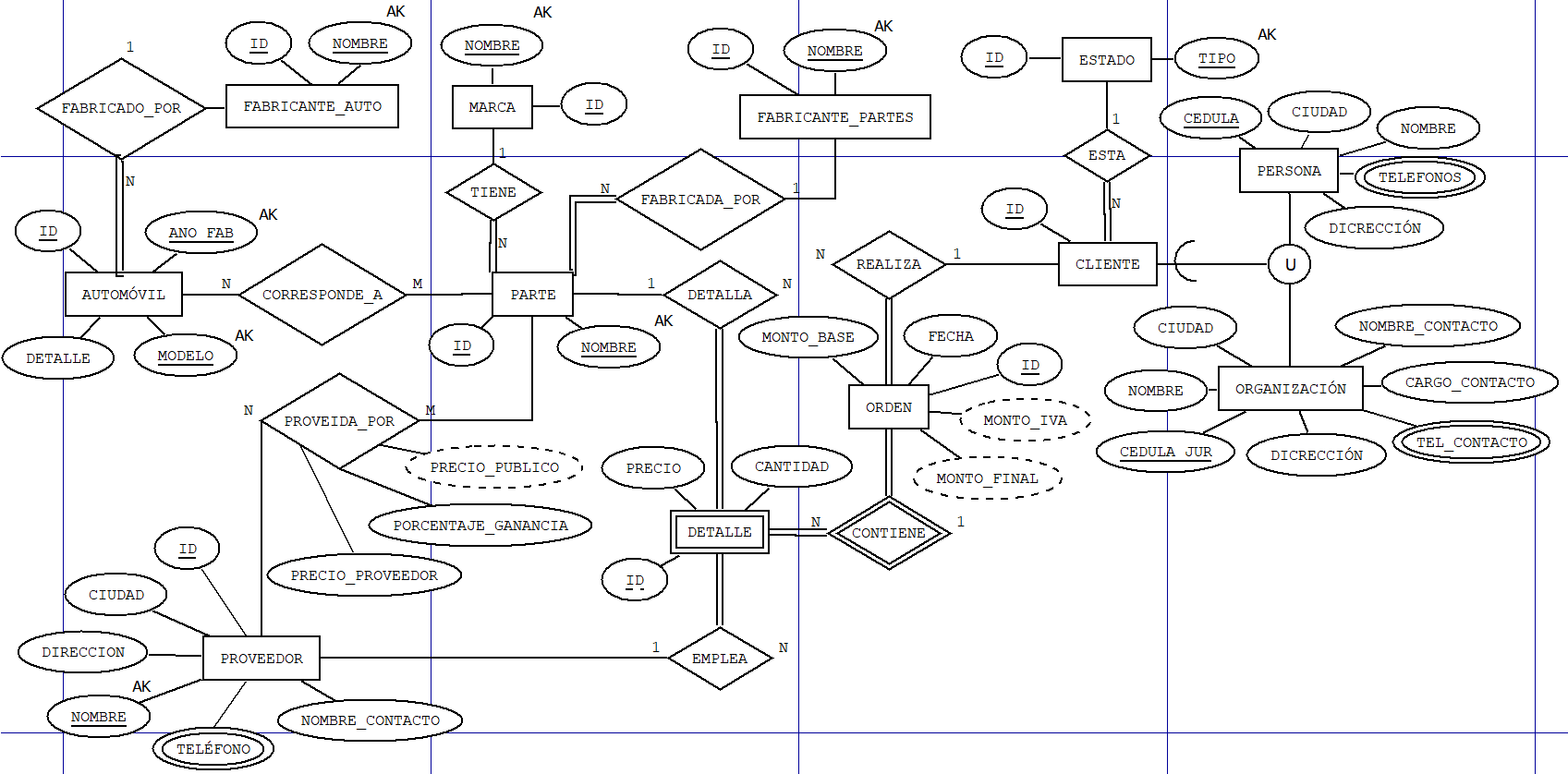
El presente proyecto tiene por objetivo la implementación de una base de datos y aplicación respectiva para un negocio de venta de repuestos. Dicha implementación busca crear una herramienta que permita guardar y administrar la información de varios aspectos del negocio. A grandes rasgos, se requiere almacenar y manipular la información referente a aspectos tales como clientes, partes, proveedores y órdenes. Una vez diseñada una base de datos que se adecue al almacenamiento de la información antes mencionada, se procedió a la construcción de una aplicación de usuario para mediar el uso de la misma.

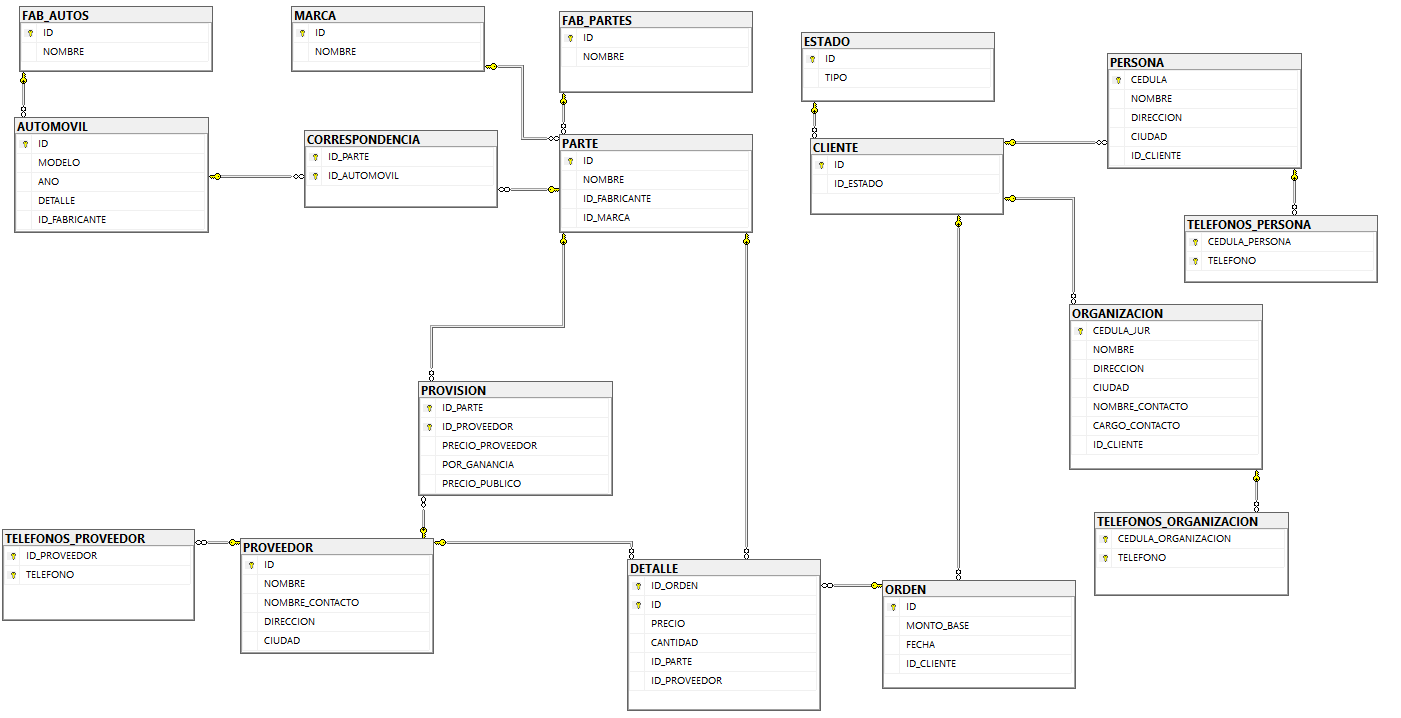
1. ***Ambiente de Desarrollo***

Las herramientas utilizadas para el desarrollo de este proyecto son:

* Microsoft SQL Server Management Studio 2018
* Visual Studio 2019 (incluyendo la carga de trabajo de almacenamiento y procesamiento de datos y los proyectos de integración de servicios de *SQL Server*, obtenibles por medio del paquete *SQL Server Data Tools*)
* Apache NetBeans 11
* DiaChart

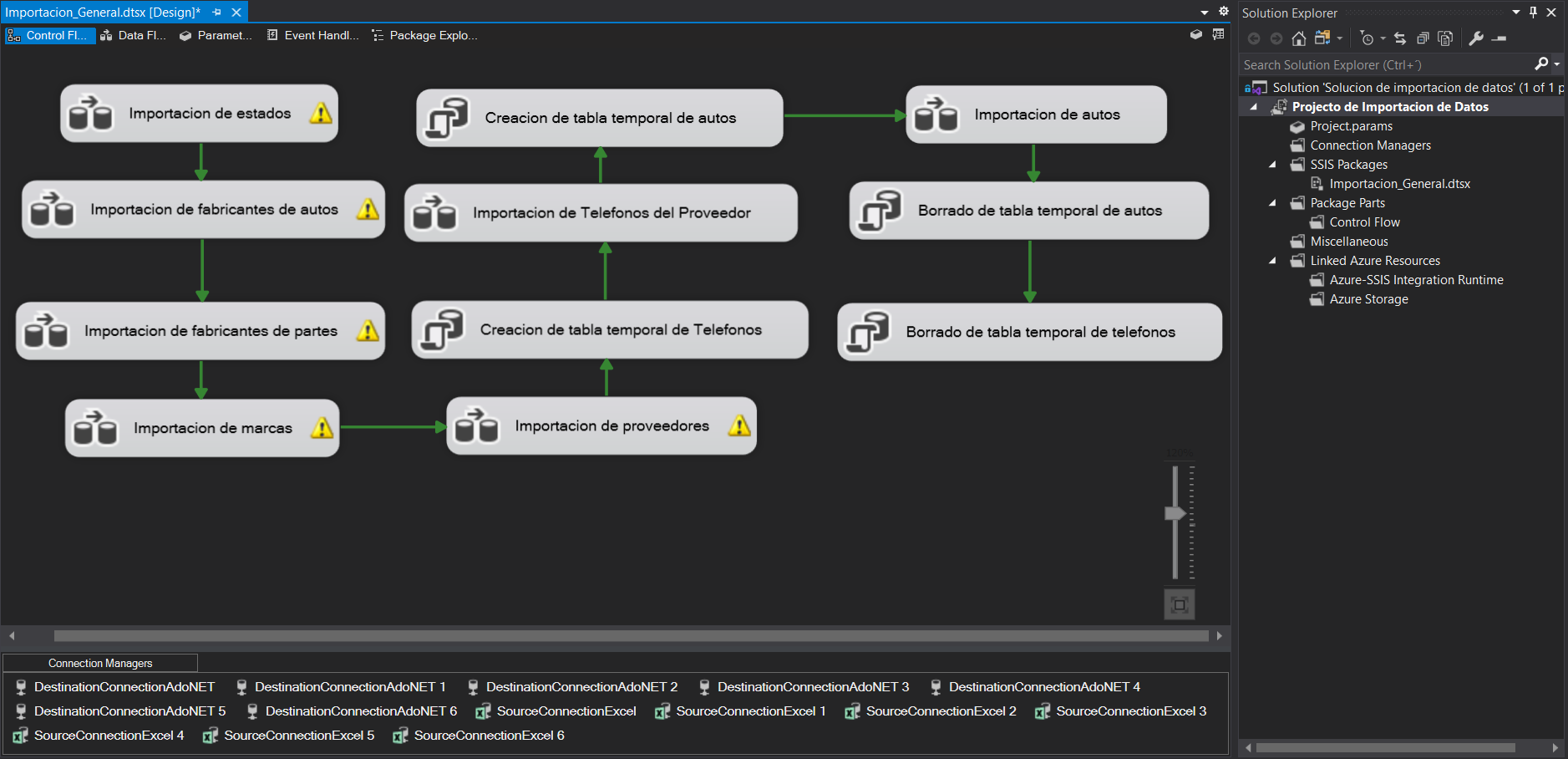
1. ***Diagramas y Diccionario de Datos***





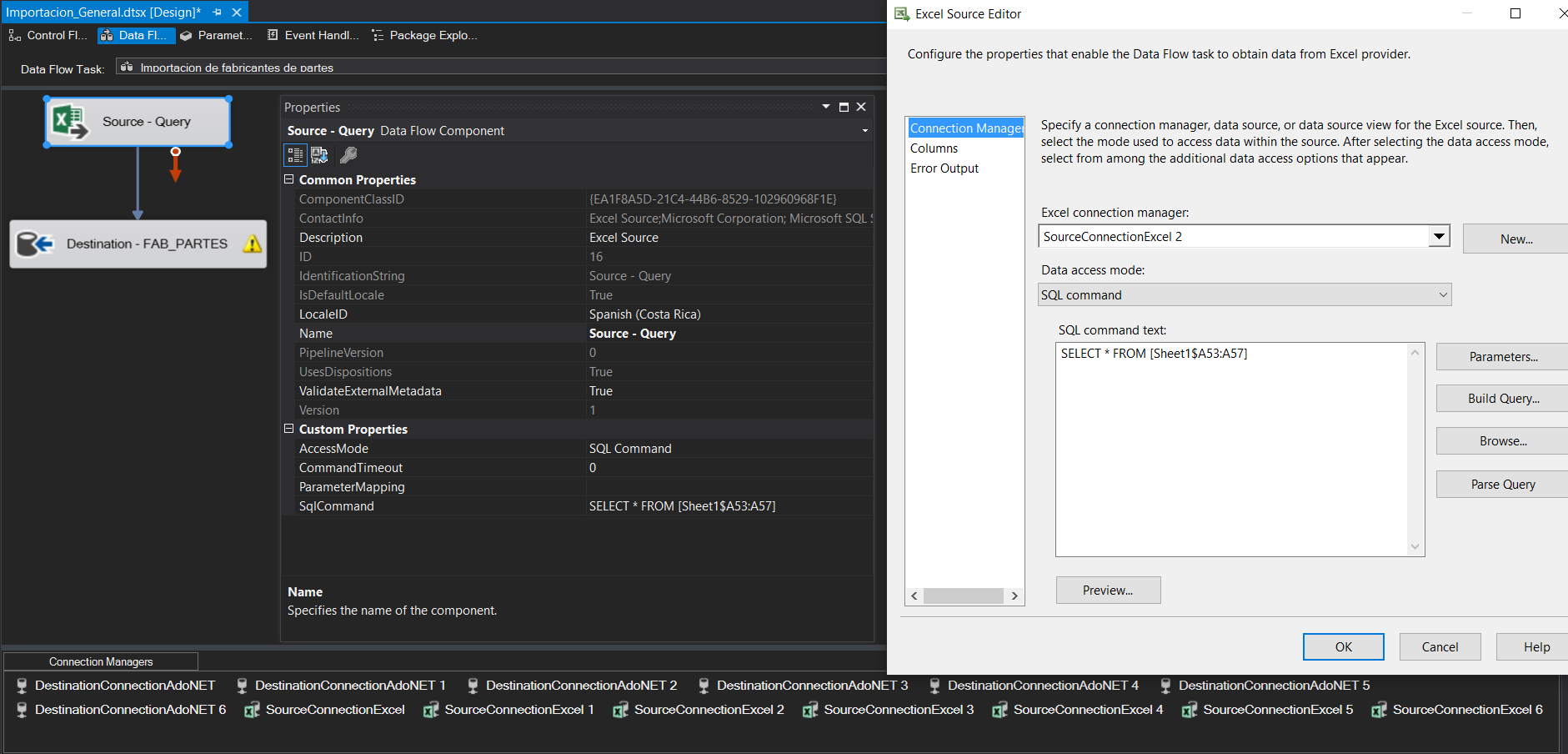
1. ***Instrucciones de Uso de la aplicación***
2. ***Importación de Datos Iniciales***

Una vez que se insertó la base de datos diseñada en una instancia de SQL Server, se procedió a la importación de los datos iniciales. Dicha carga inicial se realizó por medio de un proyecto de *integration services* de Visual Studio, el cual se contiene dentro de una solución también de Visual Studio. Los proyectos de *integration services* suelen constituirse de paquetes que, entre muchas otras funciones, son capaces de crear un flujo de datos entre un archivo de Microsoft Excel y una base de datos de SQL Server. En el caso del presente proyecto, se creó un único paquete que establece un flujo de datos entre el archivo de Excel brindado y la base de datos diseñada. En la siguiente imagen se puede aprecian las tareas que ocurren al ejecutar dicho paquete:

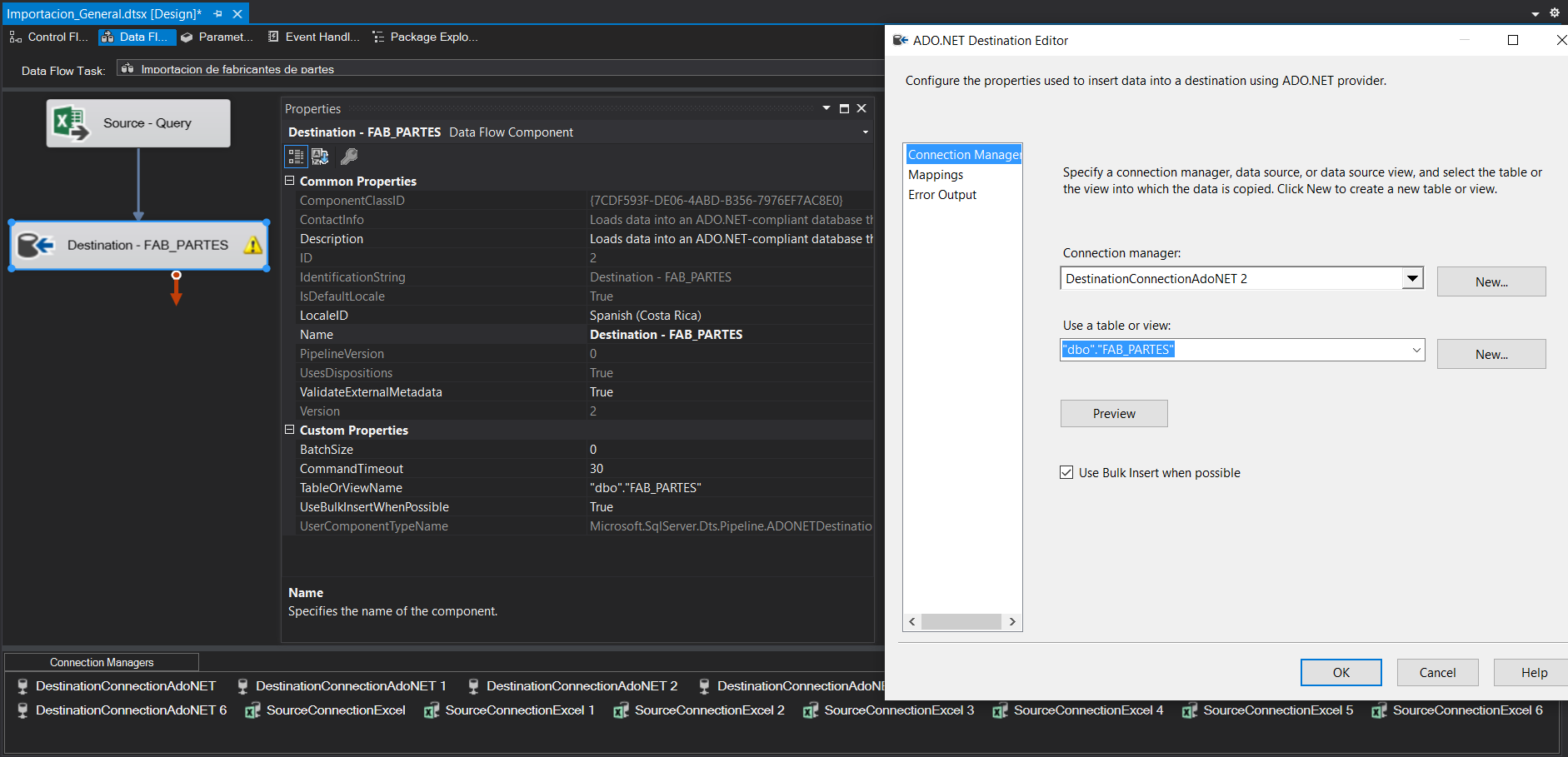


Como se puede observar en la imagen, dichos paquetes son también capaces de ejecutar consultas previamente almacenadas sobre la base de datos en cuestión. En el caso del paquete de importación construido, este crea y manipula dos tablas dentro de la base de datos destino. Asimismo, dichas tablas son eliminadas por el paquete una vez finalizado el movimiento de datos. En cuanto a las tareas de importación dentro del paquete, estas se diseñaron por medio del SQL Server Import and Export Wizard. Como ejemplo de lo anterior, a continuación, se muestran capturas del diseño de la tarea de importación de fabricantes de partes.

1. Propiedades y diseño de la fuente (Archivo de Excel)



1. Propiedades y diseño del destino (Tabla en la base de datos)



Cabe resaltar que este paquete de importación está diseñado para trabajar exclusivamente con el modelo de base de datos del proyecto, además de estar configurado para encontrar dicha base de datos en la instancia de SQL Server usada durante el desarrollo del mismo. Lo anterior ocasiona que el paquete no puede ejecutarse de inmediato en cualquier equipo, no obstante, basta con modificar la propiedad “ServerName” en los 6 *connection managers* de destino que usa el paquete. Si la instancia a la que apunta dicha propiedad permite la autenticación de Windows y posee una base de datos concorde a lo diseñado, la importación se puede realizar ejecutando el único paquete dentro de la solución de Visual Studio adjuntada a esta documentación. Sin embargo, se debe tomar en cuenta que dicha solución solo se puede abrir desde una instalación de Visual Studio que cumpla lo especificado en la sección de ambiente de desarrollo (2) del presente documento.

1. ***Resultados de los casos de prueba***

**CLIENTES**

|  |  |
| --- | --- |
| Insertar nuevos clientes personas | Éxito |
| Insertar nuevos clientes organizaciones | Éxito |
| Modificar un cliente viejo | Éxito |
| Suspender un cliente | Éxito |
| Listar clientes | Éxito |

**PARTES**

|  |  |
| --- | --- |
| Insertar nuevas partes | Éxito |
| Borrar una parte que no participa en una orden | Éxito |
| Borrar una parte que sí participa en una orden (intentar luego de insertar órdenes) | Éxito |
| Asociar partes con proveedores | Éxito |
| Asociar partes con tipos de automóviles | Éxito |
| Actualizar precios de una parte ofrecida por un proveedor | Éxito |
| Listar partes por tipos de automóviles | Éxito |

**ÓRDENES**

|  |  |  |
| --- | --- | --- |
| Localizar proveedores de una parte | | Éxito |
| Insertar una nueva orden y asociar a una orden la compra de una cantidad de partes | | Éxito |
|  | Caso 1 debe funcionar | Éxito |
| Caso 2 debe rechazarse  (cliente SUSPENDIDA) | Éxito |
| Cambiar estado a ACTIVO | Éxito |
| Caso 2 de nuevo, debe funcionar | Éxito |

1. ***Comentarios Finales***

El estado final del proyecto cumple satisfactoriamente los objetivos planteados al principio de su desarrollo. La base de datos diseñada se ajusta a la situación planteada y la aplicación desarrollada cumple todas las funcionalidades requeridas por dicho escenario, además, el aspecto final de la aplicación se encuentra amigable con el usuario. En cuanto a las limitaciones del estado final del proyecto, estas se encuentran casi exclusivamente en la importación de datos iniciales realizada. Lo anterior se debe a que, si bien dicha importación se realizó satisfactoriamente por medio de paquetes de proyecto de *integration* *services*, la naturaleza de dichos paquetes establece varios requisitos de software para su ejecución en varios equipos, tales como una instalación de Visual Studio y la configuración de los *connection managers* del paquete.