

# Capítulo 1

## Transferencia de información, desde y hacia SQL Server

**Al finalizar el capítulo, el alumno podrá:**

- Identificar la herramienta de importación y exportación adecuada para cada caso.
- Importar datos hacia la base de datos.
- Exportar datos desde la base de datos.
- Aplicar las mejores prácticas de transferencia de información.
- Identificar el funcionamiento general de Integration Services.

### **Temas**

1. Herramientas de exportación e Importación de datos
2. Importación de datos hacia SQL Server
3. Exportación de datos desde SQL Server
4. Configurar la base de datos para agilizar la transferencia de datos
5. Servicios de integración

## 1. Herramientas de exportación e importación de datos

### Herramientas de exportación e importación de datos

**Herramienta de línea de comando**


- Bulk Copy Program (BCP ) → Programa de copia masiva

**Herramientas de Transact-SQL**

- BULK INSERT
- OPENROWSET con el comando BULK (masivo)

**Herramientas gráficas**

- Asistente de importación y exportación de datos



1 - 4

Copyright © Todos los Derechos Reservados - Cibertec Perú S.A.C.

La capacidad para transformar datos en información útil es uno de los objetivos clave de cualquier empresa u organización, para la toma de decisiones. Los datos suelen proceder de varios orígenes, cada uno de los cuales se desarrolló para satisfacer una necesidad específica de la empresa u organización. Puede suceder que estos orígenes de datos, usen almacenamiento de datos en archivos o sistemas de base de datos diferentes, que tienen datos duplicados o almacenan los datos en estructuras o tipos de datos, diferentes.

En estos casos se presenta la necesidad de convertir, limpiar, normalizar, resumir y combinar datos, especialmente cuando proceden de sistemas heredados o de orígenes externos.

Una forma potente de procesar estos datos es a través de SQL Server Integration Services (SSIS), herramienta que proporciona una arquitectura flexible, rápida y escalable que permite la integración de datos, de forma eficaz, en entornos empresariales actuales.

Prácticamente todas las soluciones de bases de datos requieren algún tipo de transferencia de datos.

Normalmente, el proceso de transferir datos de un sistema a otro, implica los pasos siguientes:

- Extraer datos de algún origen homogéneo o heterogéneo.
- Transformar los datos (opcional).
- Cargar los datos en una base de datos de destino.

Algunos escenarios comunes para la transferencia de datos son:

- Trasladar los datos entre diversos sistemas.
- Consolidar datos en un lugar, a partir de varios orígenes.
- Mover datos de un sistema antiguo a un sistema nuevo.
- Mover datos de archivos, como Excel o texto, a una base de datos SQL Server.

## 1.1 **Herramientas de transferencia de información**


SQL Server tiene varias herramientas para transferir datos. Cada una de ellas es adecuada para escenarios concretos:

- Herramienta BCP. Bulk Copy Program o programa de copia masiva es una herramienta de línea de comandos para importar y exportar los datos archivos de texto. Es una herramienta útil cuando es necesario importar o exportar datos, desde la línea de comandos o un archivo de proceso por lotes. Proporciona un rendimiento rápido con una carga mínima para el sistema.
- Comando BULK INSERT. BULK INSERT es la versión de Transact-SQL del BCP y tiene muchos de los mismos parámetros, aunque solo sirve para importar datos hacia SQL Server. Es útil cuando es necesario incluir la funcionalidad de importación de datos en la lógica de Transact-SQL.
- Comando OPENROWSET. Incluye información de conexión que es necesaria para acceder a datos remotos, utilizando una fuente de datos OLE DB. Es una alternativa al uso de servidores vinculados (linked server) y es un método ad hoc de conectarse a datos remotos que se requieren con poca frecuencia.
- Asistente para importación y exportación. Forma parte de SSIS. Es el método más simple para crear un paquete. Puede importar y exportar de SQL Server, archivos de texto, Microsoft Office Access, Excel, BD OLE y proveedores de Microsoft ADO.NET. Además, puede usarse para crear paquetes SSIS simples.
- SQL Server Integration Services (SSIS). SQL Server Integration Services es un sistema avanzado para la transferencia y transformación de datos. Proporciona una arquitectura flexible, rápida y escalable que permite la integración de datos de forma eficaz. La transferencia y transformación de datos puede ser muy sencilla mediante SSIS, pero también puede resultar una compleja extracción, validación y transformación de datos.

## 2. Importación de datos hacia SQL Server 2016

### Importación de datos hacia SQL Server 2016

Usando el asistente




Clic derecho en la base de datos : **Tareas / Importar datos...**

- 1 Definir el origen externo de los datos
- 2 Especificar columnas de origen
- 3 Definir el destino de los datos
- 4 Definir la tabla de destino y el mapeo de datos
- 5 Definir si va a almacenar el paquete para cargas posteriores
- 6 Ejecutar la carga

1 - 9

Copyright © Todos los Derechos Reservados - Cibertec Perú S.A.C.



### 2.1 El asistente para importación y exportación de datos

Es una herramienta gráfica útil y simple de usar, en caso se requiera copiar y ejecutar transformaciones simples de datos, desde y hacia SQL Server, teniendo en cuenta que los datos sean compatibles.

El asistente guía, paso a paso, al usuario en la ejecución de la transferencia de datos y permite grabar la configuración como un paquete de SQL Server para ejecuciones posteriores.

Además, puede copiar datos entre orígenes de datos para los que esté disponible un proveedor de datos de .NET Framework administrado o un proveedor OLE DB nativo. La lista de proveedores disponibles incluye los orígenes de datos siguientes:

- SQL Server
- Archivos planos
- Microsoft Office Access
- Microsoft Office Excel

Los pasos a seguir para ejecutar el asistente para exportar datos son los siguientes:

1. Definir el origen de los datos
2. Especificar columnas de origen
3. Definir el destino de los datos
4. Definir la tabla de destino y el mapeo de datos
5. Definir el almacenamiento del paquete para cargas posteriores.
6. Ejecutar la carga.

A continuación, se detalla cada uno de los pasos.

1. Definir el origen de los datos.

Origen de datos: Origen de archivo plano

Seleccione un archivo y especifique sus propiedades y formato.

Nombre de archivo: D:\SQLdata\Carga Test Examinar...

Configuración regional: Español (Perú) ☐ Unicode

Página de códigos: 1252 (ANSI - Latin I)

Formato: Delimitado

Calificador de texto: <ninguno>

Delimitador de filas de encabezados: {CR}{LF}

Filas de encabezados que se omitirán: 0

☒ Nombres de columna de la primera fila de datos

2. Especificar columnas de origen.

Especifique los caracteres que delimitan el archivo de origen:

Delimitador de filas: {CR}{LF}

Delimitador de columnas: Coma (,)

Vista previa de las filas 2-7:

c1	c2
1	a
2	b
3	c
4	d
5	e
6	f

Actualizar Restablecer columnas

## 3. Definir el destino de los datos.

Destino: SQL Server Native Client 11.0

Nombre del servidor: DBLEARNER-01\SQL130

Autenticación

☒ Usar autenticación de Windows

☐ Usar autenticación de SQL Server

Nombre de usuario:

Contraseña:

Base de datos: WideWorldImporters

Actualizar

Nueva...

## 4. Definir la tabla de destino y el mapeo de datos.

Tablas y vistas:

Origen: D:\SQLdata\CargaTest.txt

Destino: [dbo].[CargaTest]

Asignaciones de columnas

Origen: D:\SQLdata\CargaTest.txt

Destino: [dbo].[CargaTest]

☒ Crear tabla de destino

☐ Eliminar filas en la tabla de destino

☐ Anexar filas a la tabla de destino

☐ Quitar y volver a crear la tabla de destino

☐ Habilitar la inserción de identidad

Editar SQL...

Asignaciones:

Origen	Destino	Tipo	Acept...	Tamaño	Precisión	Esc...
c1	Campo1	int	<input checked="" type="checkbox"/>			
c2	Campo2	varchar	<input checked="" type="checkbox"/>	50		

5. Definir el almacenamiento del paquete para cargas posteriores.

☒ Ejecutar inmediatamente

☐ Guardar el paquete SSIS

☒ SQL Server

☐ Sistema de archivos













Nivel de protección de paquetes:

Cifrar la información confidencial con una clave de usuario

Contraseña:

Vuelva a escribir la contraseña:

6. Ejecutar la carga.

 <b>Correcto</b>		11 Total	0 Error
		11 Correcto	0 Advertencia
Detalles:			
	Acción	Estado	Mensaje
	Inicializando la tarea Flujo de datos	Correcto	
	Inicializando conexiones	Correcto	
	Configurando comando SQL	Correcto	
	Configurando la conexión de origen	Correcto	
	Configurando conexión de destino	Correcto	
	Validando	Correcto	
	Preparar para la ejecución	Correcto	
	Ejecutar previamente	Correcto	
	Ejecutando	Correcto	
	Copiando en [dbo].[CargaTest]	Correcto	<a href="#">6 filas transferidas</a>
	Ejecutar posteriormente	Correcto	

## 2.2 Bulk Insert

Es una herramienta de carga de datos que importa un archivo desde una fuente externa de datos, en una tabla de la base de datos. Se requiere especificar un formato para establecer la estructura de los datos de destino.

La instrucción Bulk Insert de Transact-SQL, se ejecuta en el mismo proceso que SQL Server, por lo que comparte el mismo espacio de direcciones de memoria. Como el que abre los archivos de datos es un proceso de SQL Server, no es posible la copia de datos entre procesos.

La sintaxis es la siguiente:

#### BULK INSERT

```
[ database_name . [ schema_name ] . | schema_name . ] [ table_name | view_name ]
FROM 'data_file'
[ WITH
(
[ [ , ] BATCHSIZE = batch_size ]
[ [ , ] CHECK_CONSTRAINTS ]
[ [ , ] FIELDTERMINATOR = 'field_terminator' ]
[ [ , ] FIRSTROW = first_row ]
[ [ , ] FIRE_TRIGGERS ]
[ [ , ] FORMATFILE = 'format_file_path' ]
[ [ , ] MAXERRORS = max_errors ]
[ [ , ] ORDER ( { column [ ASC | DESC ] } [ ,...n ] ) ]
[ [ , ] ROWS_PER_BATCH = rows_per_batch ]
[ [ , ] ROWTERMINATOR = 'row_terminator' ]
[ [ , ] TABLOCK ]
[ [ , ] ERRORFILE = 'file_name' ]
)]
```

**Nota**

*En la sintaxis se han indicado los parámetros de uso común, pero si se desean revisar los parámetros completos, se pueden consultar los vínculos en las lecturas de este módulo.*

La instrucción Bulk Insert acepta la indicación Tablock, que permite al usuario especificar el comportamiento de bloqueo que se va a utilizar.

En carga masiva con Bulk Insert, Tablock especifica que se utilice el bloqueo de nivel de tabla de actualización masiva durante la carga masiva. Esto permite mejorar el rendimiento de la operación de carga masiva, ya que se reduce el conflicto de bloqueo en la tabla. Asimismo, Tablock es un parámetro obligatorio para el registro masivo cuando el modo de recuperación de la base de datos está configurado en Registro Masivo (BULK\_LOGGED) o Simple.



Bulk Insert es un comando T-SQL, por lo tanto se puede utilizar en transacciones de usuario:

```
BULK INSERT tempdb.dbo.TipoCambio
FROM ,c:\fuentes\TipoCambio.txt'
WITH
(
    FIELDTERMINATOR = '|',
    ROWTERMINATOR = '|n',
    CHECKCONSTRAINTS
)
```

En este caso, se está leyendo de forma masiva el archivo TipoCambio.txt y se están utilizando opciones para definir los terminadores de campo y registro, además, se está indicando que las restricciones deben ser verificadas.

### 2.3 Openrowset

Openrowset se utiliza para importar datos de diferentes fuentes a una Base de SQL Server en cualquiera de sus versiones. Algunas de esas fuentes de datos pueden ser una base de Microsoft Access, un libro de trabajo en Excel, un archivo de texto por algún tipo de delimitador, entre otros.

La sintaxis es la siguiente:

```
OPENROWSET
( { ,nombre_provider' , { ,fuente' ; ,usuario' ; 'password'
  | ,cadena_provider' }
  , { [ catalogo. ] [ esquema. ] objeto
    | 'query'
  }
  | BULK ,archivo_datos_fuente' ,
    { FORMATFILE = ,ruta_archivo_formato' [ <opciones_bulk> ]
    | SINGLE_BLOB | SINGLE_CLOB | SINGLE_NCLOB }
  } )
```

**Nota**

En la sintaxis se han indicado los parámetros de uso común, pero si se desean revisar los parámetros completos, se pueden consultar los vínculos en las lecturas de este módulo.

Openrowset puede ser usado como parte de la sentencia Select, de modo que se pueden consultar fuentes externas como si fueran expresiones de tabla, las cuales se pueden combinar con otras como tablas (físicas, temporales, vistas, funciones con valor de tabla, etc.)

Si se desea procesar información de fuentes externas de manera eventual, Openrowset es una solución adecuada. Si esos datos externos son utilizados con frecuencia se recomienda utilizar servidores vinculados. Openrowset se utiliza en conjunto con la sentencia Select. Mientras que con la opción Bulk se procesan los datos como carga masiva.

```
SELECT * FROM OPENROWSET(  
    BULK 'c:\fuentes\TipoCambio.csv',  
    FORMATFILE = 'c:\fuentes\TC.fmt',  
    FIRSTROW = 2) AS a;  
GO
```

En este caso se está leyendo, de forma masiva el archivo **TipoCambio.csv** y se está utilizando el archivo de formato **TC.fmt**.

Es necesario usar un archivo de formato para definir los tipos de columna del resultado. Cuando se especifica SINGLE\_CLOB, SINGLE\_BLOB o SINGLE\_NCLOB no es necesario usar el archivo de formato.

Los archivos de formato pueden ser: XML y no XML.

## 2.4 Bulk Copy Program (BCP)

La utilidad BCP es una herramienta de línea de comando que copia datos de forma masiva desde y hacia SQL Server, en un archivo de datos en un formato especificado por el usuario. Se puede usar para importar un número elevado de filas nuevas en tablas de SQL Server o para exportar datos de tablas a archivos tipo Excel o texto.

Además, BCP permite importar y exportar datos almacenados en tablas de una base de datos.

La sintaxis es la siguiente:

```
bcp [database_name.] schema.{table_name | view_name | "query" {in data_file | out  
data_file | queryout data_file | format nul}
```

```
[-a packet_size]  
[-b batch_size]  
[-c]  
[-d database_name]  
[-e err_file]  
[-f format_file]  
[-F first_row]  
[-h"hint [...n]" ]  
[-i input_file]  
[-K application_intent]  
[-L last_row]  
[-m max_errors]
```

```
[ -o output_file ]  
[ -P password ]  
[ -r row_term ]  
[ -S [server_name[instance_name]] ]  
[ -t field_term ]  
[ -T ]  
[ -U login_id ]  
[ -x ]  
/?
```

**Nota**

*En la sintaxis se han indicado los parámetros de uso común, pero si desean revisar los parámetros completos, se pueden consultar los vínculos en las lecturas de este módulo.*

Existen dos limitaciones que se deben considerar cuando se use esta utilidad:

- Tiene capacidades limitadas para transformar los datos, es decir, si los datos que requieren ser insertados en la base de datos necesitan pasar por alguna transformación compleja o validación, BCP no es la herramienta correcta.
- Tiene capacidades limitadas para el manejo de errores, es decir, que la herramienta puede reconocer que un error ocurrió mientras se encontraba cargando información, pero tiene acciones limitadas para reaccionar contra ese error, como sí lo haría SSIS.

Por consiguiente, BCP solo se usa cuando se desea cargar muchos datos de manera directa a una tabla.

Por ejemplo:

```
C:\> Administrador: C:\Windows\system32\cmd.exe  
D:\SQLdata>bcp tempdb.dbo.TipoCambio in TipoCambio.csv -T -f tc.xml
```

El parámetro **-T** indica que se va a hacer una conexión de confianza, lo que significa que va a ser ejecutado con los permisos asignados al usuario de red, en este caso el usuario debe tener permisos de **SELECT/INSERT** en **dbo.TipoCambio**, ya que son los mínimos privilegios para insertar registros, pero también necesita permisos de **ALTER TABLE**, ya que por default el BCP deshabilita los desencadenadores y las restricciones.

El parámetro **-f** indica que se va a utilizar un archivo de formato, el cual define la estructura de los datos de destino. El contenido del archivo de formato TC.xml es el siguiente:

```
<?xml version="1.0"?>
<BCPFORMAT xmlns="http://schemas.microsoft.com/sqlserver/2004/bulkload/format"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <RECORD>
    <FIELD ID="1" xsi:type="CharTerm" TERMINATOR="," MAX_LENGTH="7"/>
    <FIELD ID="2" xsi:type="CharTerm" TERMINATOR="," MAX_LENGTH="3"
COLLATION="SQL_Latin1_General_CP1_CI_AS"/>
    <FIELD ID="3" xsi:type="CharTerm" TERMINATOR="," MAX_LENGTH="25"
COLLATION="SQL_Latin1_General_CP1_CI_AS"/>
    <FIELD ID="4" xsi:type="CharTerm" TERMINATOR="\r\n" MAX_LENGTH="24"/>
  </RECORD>
  <ROW>
    <COLUMN SOURCE="1" NAME="Llave" xsi:type="SQLINT"/>
    <COLUMN SOURCE="2" NAME="Moneda" xsi:type="SQLCHAR"/>
    <COLUMN SOURCE="3" NAME="TipoCambio" xsi:type="SQLDECIMAL" PRECISION="18"
SCALE="5"/>
    <COLUMN SOURCE="4" NAME="Fecha" xsi:type="SQLDATETIME"/>
  </ROW>
</BCPFORMAT>
```

### 3. Exportación de datos desde SQL Server 2016

**Exportación de datos desde SQL Server 2016**

Usando el asistente para exportar



Clic derecho en la base de datos : **Tareas / Exportar datos...**

- 1 Definir el origen interno de los datos
- 2 Definir archivo destino de los datos
- 3 Definir destino y mapeo de datos
- 4 Definir si va a almacenar el paquete para exportaciones posteriores
- 5 Ejecutar la exportación

1 - 22

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



#### 3.1 Asistente de Importación de datos

Pasos a seguir para exportar datos:

1. Definir el origen interno de los datos.
2. Definir archivo destino de los datos.
3. Definir destino y mapeo de datos.
4. Definir el almacenamiento del paquete para exportaciones posteriores.
5. Ejecutar la exportación.

A continuación, se detallan cada uno de los pasos:

1. Definir el origen interno de los datos.

2. Definir archivo destino de los datos.

3. Definir destino y mapeo de datos.

Origen	Destino	Tipo	Acept...	Tamaño	Precisión	Esc...
PersonID	PersonID	Long	<input type="checkbox"/>			
FullName	FullName	LongText	<input type="checkbox"/>			
PreferredName	PreferredName	LongText	<input type="checkbox"/>			
SearchName	SearchName	LongText	<input type="checkbox"/>			
IsPermittedToLo...	IsPermittedToLo...	Short	<input type="checkbox"/>			
LogonName	LogonName	LongText	<input checked="" type="checkbox"/>			
IsExternalLogon...	IsExternalLogon...	Short	<input type="checkbox"/>			
HashedPassword	HashedPassword	LongBinary	<input checked="" type="checkbox"/>			
IsSystemUser	IsSystemUser	Short	<input type="checkbox"/>			

4. Definir el almacenamiento del paquete para exportaciones posteriores.

☒ Ejecutar inmediatamente

☐ Guardar el paquete SSIS

☒ SQL Server

☐ Sistema de archivos

Nivel de protección de paquetes:

Cifrar la información confidencial con una clave de usuario

Contraseña:

Vuelva a escribir la contraseña:

5. Ejecutar la exportación.

			11 Total	0 Error
			9 Correcto	0 Advertencia
Detalles:				
	Acción	Estado	Mensaje	
✓	Inicializando la tarea Flujo de datos	Correcto		
✓	Inicializando conexiones	Correcto		
✓	Configurando comando SQL	Correcto		
✓	Configurando la conexión de origen	Correcto		
✓	Configurando conexión de destino	Correcto		
✓	Validando	Correcto		
✓	Preparar para la ejecución	Correcto		
✓	Ejecutar previamente	Correcto		
✓	Ejecutando	Correcto		
▶	Copiando en 'Person'	5244 filas transferidas		
	Ejecutar posteriormente			

### 3.2 Bulk Copy Program (BCP)

En este capítulo, se explicarán los pasos a seguir para exportar datos:

Por ejemplo:

```
C:\>bcp "SELECT Name FROM Sales.Currency" queryout Currency.Name.dat -T -c
```

El parámetro **-T** indica que se va a hacer una conexión de confianza, lo que significa que va a ser ejecutado con los permisos asignados al usuario de red, en este caso el usuario debe tener permisos de escritura en la carpeta donde se creará el archivo.

El parámetro **-c** indica que el resultado tiene formato carácter.

La fuente de información para la exportación de datos es la tabla **Sales.Currency**, pero no se está exportando toda la información de dicha tabla si no el campo Name. Es resumen, en este caso la fuente está definida por la siguiente consulta:

```
SELECT Name FROM Sales.Currency
```

Es posible elaborar consultas más complejas con filtros WHERE y uniones de más de una tabla.



## 4. Configurar la base de datos para agilizar la transferencia de datos

### Configurar la base de datos para agilizar la transferencia

- Se recomienda deshabilitar índices, restricciones y desencadenadores:
  - No es necesario verificar las restricciones por cada línea cargada.
  - Los índices no deben ser actualizados durante la carga.
- Preferible cargar cuando usuarios no están trabajando.
- Si no se puede, minimizar bloqueo usando el Query Hint TABLOCK para acelerar la importación.
- Minimizar el uso de archivo de transacciones:
  - Modelos de recuperación BULK\_LOGGED o SIMPLE.
- Minimizar conversiones de datos:
  - Usar formatos nativos de SQL Server.

1 - 30

Copyright © Todos los Derechos Reservados - Cibertec Perú SAC.



Las operaciones de carga masiva pueden realizar un registro optimizado (es decir, solo se registran las asignaciones de páginas de registro y no los datos reales ni las filas de índice) en determinadas condiciones. Si la base de datos está configurada en modo de recuperación completa, el registro especializado para las operaciones en bloque y de ordenación no está habilitado. Si es posible, se debe establecer el modo de recuperación de la base de datos en recuperación de registro masivo (BULK\_LOGGED) o Recuperación simple (SIMPLE) durante las operaciones de carga masiva.

El esquema de la tabla de destino afecta a los planes de consultas de las operaciones de carga masiva, al registro optimizado y a la carga masiva simultánea y, en última instancia, repercute en el rendimiento de la operación de carga masiva.

### 4.1. Índices

La mayoría de las tablas contienen índices; la tabla de destino de la operación de carga masiva no es una excepción; sin embargo, la inclusión de índices en la tabla de destino afecta al rendimiento de la carga masiva.

Hay dos opciones para los índices de la tabla de destino:

- Borrar uno de ellos o varios.
- Realizar una carga masiva de los datos y volver a crear después los índices borrados, o no borrar ningún índice.

## **4.2. Restricciones**

Las restricciones se comprueban para cada fila insertada. Si procede, se recomienda deshabilitar la comprobación de restricciones. La única restricción que se puede deshabilitar es la restricción de comprobación. No se puede deshabilitar la restricción de unicidad, la restricción de clave principal / clave externa ni la restricción NOT NULL. Cuando la opción de restricción de comprobación se vuelva a habilitar, SQL Server deberá comprobar toda la tabla para volver a validar las restricciones. Por este motivo, no se recomienda deshabilitar restricciones durante la carga masiva incremental, ya que es más laborioso volver a validar la restricción para toda la tabla que aplicar restricciones a los datos incrementales.

Una situación en la que puede ser conveniente deshabilitar restricciones es cuando los datos de entrada contienen filas que infringen las restricciones. Al deshabilitar las restricciones, puede cargar los datos y utilizar después, instrucciones de Transact-SQL para limpiarlos.

## **4.3. Desencadenadores**

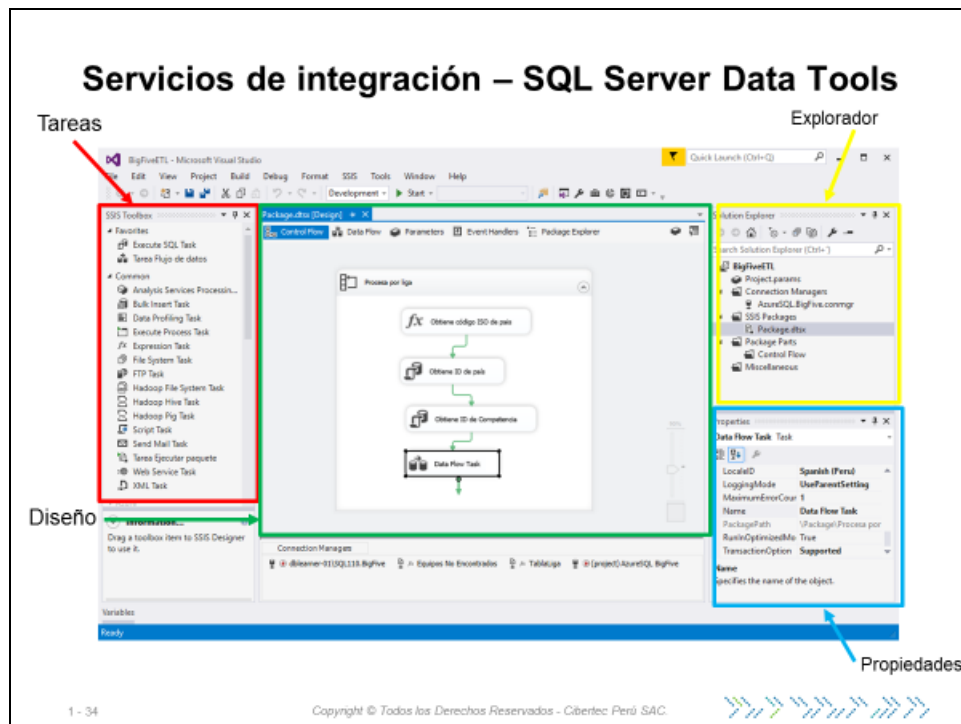
Si se han definido desencadenadores para las operaciones de inserción en la tabla de destino, se activarán para cada lote finalizado. Si procede, deshabilitar la ejecución de desencadenadores durante la operación de carga masiva.

## **4.4. Tablock**

Cuando la carga masiva se realiza en una tabla con uno o varios índices, Tablock impone un bloqueo de tabla X a las operaciones de carga masiva de forma que no sea posible la carga masiva simultánea.

Si no especifica Tablock, la carga masiva no adquiere el bloqueo de tabla, sino el bloqueo en filas o páginas. Sin embargo, los bloqueos de filas o páginas se pueden trasladar al bloqueo X, en función del tamaño del lote y de la actividad simultánea en la tabla de destino.

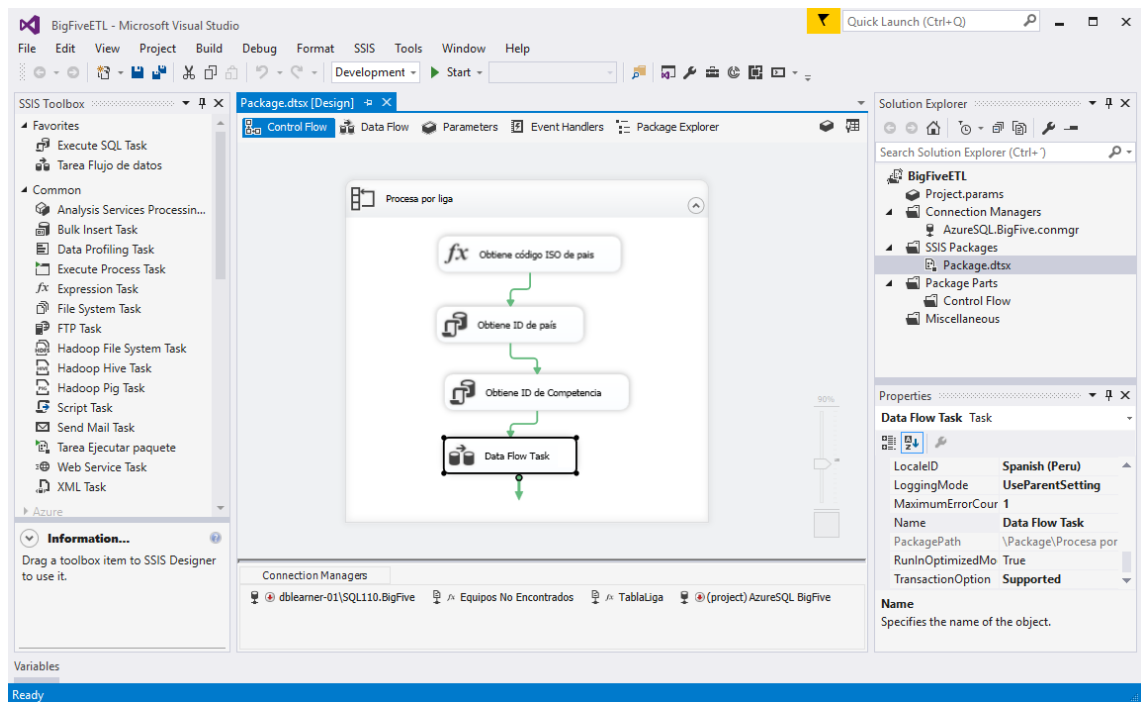
## 5. Servicios de integración



SSIS es una aplicación completa para ETL (extracción, transformación y carga por sus siglas en inglés). Es importante entender la terminología utilizada, el modo en que se estructuran los paquetes SSIS y las herramientas disponibles.

A continuación, se describen las herramientas, la terminología y los procesos de SSIS, los cuales disponen de una amplia gama de herramientas y asistentes, así como, de programas de líneas de comandos.

SSIS tiene dos motores: un motor en tiempo de ejecución y un motor de flujo de datos. El motor de flujo de datos controla las tareas de extracción, transformación y carga; el motor en tiempo de ejecución controla todo lo demás, incluidas la repetición y la comunicación con procesos externos.



## 5.1 Glosario

- **Paquete:** es una colección organizada de conexiones, elementos de flujo de control, elementos de flujo de datos, controladores de eventos, variables y configuraciones, que se ensamblan mediante las herramientas de diseño gráfico que proporciona SSIS o que se generan mediante programación. A continuación, se debe guardar el paquete finalizado en SQL Server 2005, el almacén de paquetes SSIS o el sistema de archivos. El paquete es la unidad de trabajo que se recupera, se ejecuta y se guarda.
- **Tarea:** realiza el trabajo en paquetes; SSIS incluye tareas para la realización de diversas funciones.
- **Contenedor:** proporcionan la estructura a los paquetes. Admiten los flujos de control repetidos en paquetes y agrupan tareas, así como, otros contenedores en unidades de trabajo significativas.

## 5.2 Herramientas y asistentes

- **Asistentes:** los asistentes de SSIS proporcionan una sencilla interfaz de usuario que puede usarse para llevar a cabo las tareas SSIS más habituales.
- **Data Tools:** SSIS ofrece un diseñador integrado, basado en Visual Studio. Juntos, proporcionan el principal entorno de desarrollo para crear paquetes SSIS.
- **Utilidades del símbolo de sistema:** SSIS incluye utilidades del símbolo de sistema que pueden usarse para ejecutar paquetes SSIS desde la línea de comandos o en un archivo de proceso por lotes.

- Herramientas administrativas: SSIS incluye varias herramientas administrativas que se pueden usar independientemente o a las que se puede obtener acceso a ellas, a través de SQL Server Management Studio (SSMS). Estas herramientas administrativas pueden usarse para configurar, implementar y ejecutar paquetes SSIS.

### 5.3 Niveles jerárquicos de operación

- Flujo de control: es el proceso de control de la ruta de ejecución de un paquete. Puede implantar condiciones para controlar la secuencia de las tareas, basándose en el éxito o error de una tarea anterior o en los resultados de una expresión.

Se puede agrupar tareas con propósitos organizativos o para proporcionar capacidades transaccionales. También se pueden repetir grupos de tareas para ejecutar la tarea un cierto número de veces o para cada miembro de una colección.

SSIS tiene tres tipos de componentes de flujo de control: contenedores, tareas y restricciones de precedencia.

- Los contenedores agrupan las tareas en unidades de trabajo y son compatibles con la repetición, a través del contenedor para cada miembro de una colección o hasta que una expresión determinada se evalúe como FALSE.
  - Las tareas realizan el trabajo en paquetes. Las tareas pueden:
    - Extraer, transformar y cargar datos.
    - Copiar datos.
    - Comunicarse con otros procesos.
    - Trabajar con objetos y datos de SQL Server.
    - Trabajar con objetos de Analysis Services y procesarlos.
    - Ejecutar trabajos administrativos como copias de seguridad.
  - Las tareas también contienen secuencias de comandos para implementar funcionalidades personalizadas.
  - Las restricciones de precedencia crean una secuencia ordenada de ejecución para los contenedores y las tareas.
- Flujo de datos: tiene como tarea especial abarcar las operaciones de extracción, transformación y carga de SSIS. Aparece en la página Flujo de control como una tarea normal, pero tiene su propia ficha porque está compuesta por varios componentes.

El origen extrae datos de almacenes de datos relacionales y no relacionales, como bases de datos de SQL Server, hojas de cálculo de Excel y bases de datos de Analysis Services. La transformación transforma los datos mediante la asignación del origen al destino y la limpieza, agregación y modificación de los datos, así como, del esquema. El destino carga datos en almacenes de datos o los almacena en conjuntos de datos en memoria.

El flujo de datos tiene tres tipos de componentes de flujo de datos: orígenes, destinos y transformaciones.

- Los orígenes extraen los datos de una amplia gama de almacenes de datos que incluyen bases de datos relacionales, archivos de texto y hojas de cálculo.
- Las transformaciones modifican los datos entre el origen y el destino, e incluyen la validación, agregación y asignación de datos.
- Los destinos cargan los datos en cualquiera de los almacenes de datos.

#### **5.4 Transformación de datos**

Cuando se seleccionan datos para importarlos al sistema, los datos podrían no estar en el formato necesario o no cumplir con sus reglas de validación. Con la transformación de los datos se pueden eliminar incoherencias. La transformación de los datos puede significar un simple cambio de los tipos de datos o una transformación compleja mediante código de procedimiento.

- Cambiar el formato de los datos: el cambio del formato de los datos podría implicar la simple modificación del tipo de datos char a varchar, pero también puede conllevar una transformación más compleja, como el cambio de un campo de fecha basado en cadenas a un tipo de datos datetime.
- Agregar los datos: a menudo, se necesitarán copiar datos desde un sistema de transaccional (OLTP) a un sistema para la toma de decisiones o datawarehouse (DW). Con frecuencia, el DW no requiere la granularidad fina de datos del sistema OLTP. Un DW suele generar informes resumidos y la transformación de los datos podría mejorar el rendimiento mediante la agregación previa de estos datos y el almacenamiento en un formulario resumido.
- Mantener la coherencia de los datos: al tratar con datos de sistemas diferentes, existe con frecuencia más de una versión de los mismos datos. Los nombres, direcciones y números de teléfono tienen, por lo general, ortografías o estructuras diferentes, además de la transformación de los datos puede garantizar la coherencia en el destino.
- Validar los datos: si se tiene cualquier duda sobre la validez de los datos de origen, deberían ser validados antes de ser colocados en el sistema de destino. Se pueden resolver automáticamente algunos errores de validación, pero es posible que se requiera hacerlo manualmente en el caso de que los errores sean más complejos.

#### **5.5 Implementación de paquetes**

SSIS cuenta con herramientas y características para implementar paquetes de un sistema a otro. El proceso de implementar paquetes consta de tres pasos:

1. Definir las configuraciones de paquetes.
2. Configurar la utilidad de implementación y generar el proyecto.
3. Copiar e implementar la generación.

- Configuraciones de paquetes: permiten actualizar los valores de propiedades en tiempo de ejecución. Esto es particularmente útil al trasladar un paquete de un sistema de desarrollo a un sistema de producción. Los nombres de servidor y las cadenas de conexión probablemente habrán cambiado y las configuraciones de paquetes pueden automatizar el proceso de modificación de estos valores.

Los valores que se desean aplicar pueden ser almacenados en archivos XML, tablas SQL Server, entradas del registro, o variables de entorno o de paquetes. Además, se puede agregar una referencia a esta ubicación de almacenamiento agregando una nueva configuración desde el menú SSIS de Business Intelligence Development Studio. El asistente para la configuración de paquetes se ejecuta automáticamente y guía a través del proceso de creación de una configuración de paquetes.

- Generar el proyecto: antes de poder generar el proyecto, se debe configurar la utilidad de implementación. Se tiene acceso a esta desde las propiedades de un proyecto que permitirá activar la utilidad de implementación de generación y especificar la ubicación de generación. También se puede especificar si se permiten cambios de configuración en tiempo de ejecución. Además, se pueden establecer estos cambios durante la instalación del paquete y reemplazarán las configuraciones del paquete. Una vez configurada la utilidad de implementación, se puede generar el proyecto.
- Implementar la generación: para implementar la generación, se debe copiar la carpeta de generación en el nuevo sistema y ejecutar el manifiesto. El manifiesto tendrá un nombre de archivo y nombre del proyecto. SSISDeploymentManifest. Se puede especificar si se desea efectuar la instalación en SQL Server o en el sistema de archivos, especificar las ubicaciones de almacenamiento y, opcionalmente, cambiar las configuraciones.