



E-Commerce de productos para mascotas

Migración de Datos

Carlos Montes

Santiago Trigos

Versión: 0200

FICHA DE CONTROL DE CAMBIOS				
PROYECTO		E-COMMERCE DE PRODUCTOS PARA MASCOTAS		
DOCUMENTO		PLAN DE MIGRACIÓN		
VERSIÓN		VERSIÓN N° 3		
FECHA CREACIÓN		29 DE JUNIO DE 2021		
FECHA CAMBIO				
RESPONSABLE S		SANTIAGO TRIGOS, CARLOS MONTES		
LÍDER				
HISTORIAL				
FECHA	NÚMERO DE VERSIÓN	OBSERVACIONES	AUTOR(ES)	VER
29/06/21	Versión 1	Creación del documento	Santiago Trigos, Carlos Montes	
20/05/21	Version3	Modificación del informe	Santiago Trigos Carlos Montes	Graciela Arias
27/06/21	Version4	Ingreso plan de migracion	Santiago Trigos Carlos Montes	

INTRODUCCIÓN

La migración de datos y estructura de datos consiste en la transferencia de materiales digitales de un origen a un destino, dicho proceso requiere de unos pasos específicos como: identificar las fuentes de datos a migrar, identificar las estructuras diseñadas para almacenar los datos, observar el formato que tienen los datos con el objetivo de identificar las conversiones necesarias para adaptarse a las nuevas necesidades, realizar pruebas previas a la migración con el fin de identificar oportunamente las excepciones presentes propiamente en las estructuras y datos actuales que pudiesen surgir en el proceso de la migración y por último la ejecución de la migración en su totalidad y su validación.

Existen diversos motivos para realizar una migración de estructura de datos y de datos, entre los más relevantes se tienen los siguientes:

1. Cumplir con nuevos requerimientos de usuario o software.
2. Actualización de versiones.
3. Estandarización de plataforma tecnológica.
4. Reducción de costo al optar por software libre.
5. Migración de sistemas legados.
6. Ampliación de soporte con otros proveedores.

Independientemente del motivo por el cual se generó la necesidad de la migración, es importante gestionarlo de forma planificada, utilizando técnicas y fases que permitan organizar la ejecución a través de la atención de los puntos de control clave que surjan en el proceso.

JUSTIFICACIÓN

Se requiere llevar a cabo una migración desde el gestor de base de datos de MySql a SQL Server, este ejercicio es de tipo educativo, pero es fundamental que se realice, para evaluar, analizar y entender la importancia de una migración de base de datos, y de todos los aspectos relacionados con la misma.

Hoy en día, las empresas son muy variadas, y es por tal razón que en el mercado existen varios sistemas gestores de bases de datos, para pequeñas, medianas, y grandes empresas. Cada uno de ellos presenta sus desventajas y ventajas. Sin importar si se tienen datos registrados, por ejemplo en MySql y se quieren pasar a SQL Server, en este caso lo fundamental es la información, los datos contenidos en dicho gestor.

Tener en cuenta que el proceso de migración no se realiza por arte de magia, y que aunque existe software para realizar migraciones, no hay un sistema de información que sea 100% confiable.

Hay muchos factores que intervienen en un proceso de migración: tiempo, costo, escalabilidad, tamaño de la base de datos, estructura, restricciones, complejidad, soporte técnico, experiencia, planeación, control y seguimiento, compatibilidad, etc. A continuación se va a realizar un plan de migración de base de datos MySql a SQL Server. Se espera que este documento sea de gran utilidad y sirva como base para posibles migraciones en el sistema de información de un E-Commerce.

ANÁLISIS PREVIO DEL SISTEMA ACTUAL Y FINAL

Antes de realizar la migración, se debe realizar un análisis previo del sistema gestor de base de datos actual (MySQL) y del sistema gestor de base de datos final (SQL Server). Esto se realiza para tener una visión general de las ventajas y/o desventajas de realizar dicha migración, además se estima el nivel de complejidad de llevar a cabo dicho proceso. Estos elementos son:

Número de registros filas:

Tablas:

Carrito: 0 registros
Categoria: 8 registros
Compra: 5 registros
Detalle_compra: 5 registros
Inventario: 0 registros
Login: 2 registros
Movimientos: 0 registros
Método_pago: 3 registros
Producto: 57 registros
Proveedor: 17 registros
Rol: 0 registros
Stock: 0 registros
Tipo_movimiento: 0 registros
Usuario: 14 registros

Número de atributos (NA)

Atributos Carrito: idCarrito, Producto_idProducto, Num_items

Atributos Categoría: idCategoría, Tipo_prod

Atributos Compra: idCompra, Usuario_idUsuario, Método_pago_idMétodo_pago, Fech_comp

Atributos Detalle_compra: idDetalle_compra, Producto_idProducto, Compra_idCompra, Cant

Atributos Inventario: idInventario, Entradas, Salidas, Saldo

Atributos Login: idLogin, Usuario, Contraseña

Atributos Movimientos: idMovimientos, Producto_idProducto, Fecha, Tipo, Cant

Atributos Método_pago: idMétodo_pago, Tipo_pago

Atributos Producto: idProducto, Proveedor_idProveedor, Inventario_idInventario, Categoría_idCategoría, Nom_prod, Prec_prod, Descu_prod, Fech_venc, Imag_prod, Descr_prod

Atributos Proveedor: idProveedor, Nom_prov, Tel_prov, Direc_prov, Email_prov, Estado_prov

Atributos Rol: idRol, Nom_rol

Atributos Stock: idStock, Producto_idProducto, Smín, Smáx, Sfísic, Sdisp, Pend_recib, Pend_serv

Atributos Tipo_movimiento: idTipo_movimiento, Movimientos_idMovimientos, Nom_tipo

Atributos Usuario: idUsuario, Login_idLogin, Rol_idRol, Nom1_Us, Ape1_Us, Doc_Us, Email_Us, Tel_Us, Direc_Us

Otros aspectos a evaluar

Hay una serie de factores comunes a cualquier base de datos que influyen en la complejidad de la migración y que hay que ponderar:

- Cantidad y tipo de SQL propietario que se use.
- Calidad de datos.
- Existencia de documentación del sistema.
- Requisitos de diseño tales como la alta disponibilidad y replicación.
- Software de terceros dependencias.
- Cambio en el sistema operativo y/o cambio de la plataforma de hardware como consecuencia de la migración.
- Cualificación y experiencia del personal involucrado en la migración.
- La disponibilidad de un equipo dedicado para el desarrollo de la migración.
- Imposibilidad de detener los cambios y los nuevos desarrollos sobre el código a migrar.
- Tiempo máximo que se permite para la realización de la migración (ventana de corte).

ANALISIS DE MIGRACIÓN

En este punto deben identificarse los cambios específicos que hay que realizar para transformar cada elemento de la base de datos MySQL en un elemento de SQL Server que funcione de la misma manera. También hay que identificar los cambios a realizar en el código de la propia aplicación para que funcione sobre una base de datos SQL Server. Identificando tipos de datos, funciones, triggers, elementos DML, procedimientos, etc., no soportados en SQL Server.

Por otro lado, se va a mencionar algunos tipos de datos MySQL y su equivalente en SQL Server:

Datos en MySQL y SQL Server:

Tipos de datos para texto en MySQL

Tipo de datos	Descripción
CHAR (tamaño)	Tiene una cadena de longitud fija (puede contener letras, números y caracteres especiales). El tamaño fijo se especifica entre paréntesis. Puede almacenar hasta 255 caracteres
VARCHAR (tamaño)	Tiene una cadena de longitud variable (puede contener letras, números y caracteres especiales). El tamaño máximo se especifica entre paréntesis. Puede almacenar hasta 255 caracteres. Nota: si agrega un valor mayor que 255, se convertirá en un tipo de texto
TINYTEXT	Tiene una cadena con una longitud máxima de 255 caracteres
TEXT	Tiene una cadena con una longitud máxima de 65.535 caracteres
BLOB	Para BLOB (Objetos grandes binarios). Almacena hasta 65.535 bytes de datos
MEDIUMTEXT	Tiene una cadena con una longitud máxima de 16,777,215 caracteres
MEDIUMBLOB	Para BLOB (Objetos grandes binarios). Tiene capacidad para 16.777.215 bytes de datos
LONGTEXT	Tiene una cadena con una longitud máxima de 4.294.967.295 caracteres
LOB	Para BLOB (Objetos grandes binarios). Tiene capacidad para 4.294.967.295 bytes de datos
ENUM (x, y, z, etc.)	Permite ingresar una lista de valores posibles. Puede enumerar hasta 65535 valores en una lista ENUM. Si se inserta un valor que no está en la lista, se insertará un valor en blanco. Nota: los valores se ordenan en el orden en que los ingresas. Ingrese los valores posibles en este formato: ENUM ('X', 'Y', 'Z')
SET	Similar a ENUM, excepto que SET puede contener hasta 64 elementos de lista y puede almacenar más de una opción

Tipos de datos numéricos en MySQL

Tipo de datos	Descripción
TINYINT (tamaño)	-128 a 127 normal. 0 a 255 SIN FIRMAR *. La cantidad máxima de dígitos se puede especificar entre paréntesis
SMALLINT (tamaño)	-32768 a 32767 normal. 0 a 65535 SIN FIRMAR *. La cantidad máxima de dígitos se puede especificar entre paréntesis
MEDIUMINT (tamaño)	-8388608 a 8388607 normal. 0 a 16777215 SIN FIRMAR *. La cantidad máxima de dígitos se puede especificar entre paréntesis
INT (tamaño)	-2147483648 a 2147483647 normal. 0 a 4294967295 SIN FIRMAR *. La cantidad máxima de dígitos se puede especificar entre paréntesis
BIGINT (tamaño)	-9223372036854775808 a 9223372036854775807 normal. 0 a 18446744073709551615 SIN FIRMAR *. La cantidad máxima de dígitos se puede especificar entre paréntesis
FLOAT (tamaño, d)	Un pequeño número con un punto decimal flotante. La cantidad máxima de dígitos se puede especificar en el parámetro de tamaño. El número máximo de dígitos a la derecha del punto decimal se especifica en el parámetro d
DOBLE (tamaño, d)	Un número grande con un punto decimal flotante. La cantidad máxima de dígitos se puede especificar en el parámetro de tamaño. El número máximo de dígitos a la derecha del punto decimal se especifica en el parámetro d
DECIMAL (tamaño, d)	Un DOBLE almacenado como una cadena, lo que permite un punto decimal fijo. La cantidad máxima de dígitos se puede especificar en el parámetro de tamaño. El número máximo de dígitos a la derecha del punto decimal se especifica en el parámetro d

Tipos de datos para fechas en MySQL

Tipo de datos	Descripción
DATE ()	Una fecha. Formato: AAAA-MM-DD Nota: el rango admitido es de '1000-01-01' a '9999-12-31'
DATETIME ()	* Una combinación de fecha y hora. Formato: AAAA-MM-DD HH: MI: SS Nota: el rango admitido es de '1000-01-01 00:00:00' a '9999-12-31 23:59:59'
TIMESTAMP ()	* Una marca de tiempo. Los valores de TIMESTAMP se almacenan como el número de segundos desde la época de Unix ('1970-01-01 00:00:00' UTC). Formato: AAAA-MM-DD HH: MI: SS Nota: el rango admitido es de '1970-01-01 00:00:01' UTC a '2038-01-09 03:14:07' UTC
TIME ()	Un tiempo. Formato: HH: MI: SS Nota: el rango admitido es de '-838: 59: 59' a '838: 59: 59'
YEAR ()	Un año en formato de dos o cuatro dígitos. Nota: Valores permitidos en formato de cuatro dígitos: de 1901 a 2155. Valores permitidos en formato de dos dígitos: 70 a 69, que representan los años de 1970 a 2069

Tipos de datos para texto en SQL Server

Tipo de datos	Descripción	Tamaño máximo	Almacenamiento
char (n)	Cadena de caracteres de ancho fijo	8,000 caracteres	Ancho definido
varchar (n)	Cadena de caracteres de ancho variable	8,000 caracteres	2 bytes + número de caracteres
varchar (max)	Cadena de caracteres de ancho variable	1,073,741,824 caracteres	2 bytes + número de caracteres
text	Cadena de caracteres de ancho variable	2 GB de datos de texto	4 bytes + número de caracteres
nchar	Cadena Unicode de ancho fijo	4.000 caracteres	Ancho definido x 2
nvarchar	Ancho de cadena Unicode	4.000 caracteres	
nvarchar (max)	Ancho de cadena Unicode	536,870,912 caracteres	
ntext	Ancho de cadena Unicode	2 GB de datos de texto	
binary (n)	Cadena binaria de ancho fijo	8,000 bytes	
varbinary	Cadena binaria de ancho variable	8,000 bytes	
varbinary (max)	Cadena binaria de ancho variable	2 GB	
image	Cadena binaria de ancho variable	2 GB	

Tipos de datos para fechas en SQL Server

Tipo de datos	Descripción	Almacenamiento
datetime	Del 1 de enero de 1753 al 31 de diciembre de 1999, con una precisión de 3,33 milisegundos	8 bytes
datetime2	Desde el 1 de enero de 0001 hasta el 31 de diciembre de 1999, con una precisión de 100 nanosegundos	6-8 bytes
smalldatetime	Del 1 de enero de 1900 al 6 de junio de 2079 con una precisión de 1 minuto	4 bytes
date	Almacenar una fecha solamente. Del 1 de enero de 0001 al 31 de diciembre de 9999	3 bytes
time	Almacenar un tiempo solo con una precisión de 100 nanosegundos	3-5 bytes
datetimeoffset	Lo mismo que datetime2 con la adición de un desplazamiento de zona horaria	8-10 bytes
timestamp	Almacena un número único que se actualiza cada vez que se crea o modifica una fila. El valor de la marca de tiempo se basa en un reloj interno y no corresponde a tiempo real. Cada tabla puede tener una sola variable de marca de tiempo	

Tipos de datos numéricos en SQL Server

Tipo de datos	Descripción	Almacenamiento
bit	Entero que puede ser 0, 1 o NULL	
tinyint	Permite números enteros de 0 a 255	1 byte
smallint	Permite números enteros entre -32,768 y 32,767	2 bytes
int	Permite números enteros entre -2,147,483,648 y 2,147,483,647	4 bytes
bigint	Permite números enteros entre -9,223,372,036,854,775,808 y 9,223,372,036,854,775,807	8 bytes
decimal (p, s)	Números de escala y precisión fijos. Permite números de -10^{38+1} a 10^{38-1} . El parámetro p indica el número total máximo de dígitos que se pueden almacenar (tanto a la izquierda como a la derecha del punto decimal). p debe ser un valor de 1 a 38. El valor predeterminado es 18. El parámetro s indica la cantidad máxima de dígitos almacenados a la derecha del punto decimal. s debe ser un valor de 0 a p. El valor predeterminado es 0	5-17 bytes
numeric (p, s)	Números de escala y precisión fijos. Permite números de -10^{38+1} a 10^{38-1} . El parámetro p indica el número total máximo de dígitos que se pueden almacenar (tanto a la izquierda como a la derecha del punto decimal). p debe ser un valor de 1 a 38. El valor predeterminado es 18. El parámetro s indica la cantidad máxima de dígitos almacenados a la derecha del punto decimal. s debe ser un valor de 0 a p. El valor predeterminado es 0	5-17 bytes
smallmoney	Datos monetarios de -214,748.3648 a 214,748.3647	4 bytes
money	Datos monetarios de -922,337,203,685,477.5808 a 922,337,203,685,477.5807	8 bytes
float (n)	Datos del número de precisión flotante desde $-1.79E + 308$ a $1.79E + 308$. El parámetro n indica si el campo debe contener 4 u 8 bytes. float (24) contiene un campo de 4 bytes y float (53) contiene un campo de 8 bytes. El valor predeterminado de n es 53.	4 u 8 bytes
real	Datos numéricos de precisión flotante desde $-3.40E + 38$ a $3.40E + 38$	4 bytes

Ventajas y desventajas de SQL Server:

Ventajas

- Contiene la facilidad de soporte de transacciones.
- Contiene escalabilidad, estabilidad y seguridad.
- También puede Soportar procedimientos almacenados.
- Puede Incluir también un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
- También permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y las terminales o clientes de la red sólo acceden a la información.
- Además permite administrar información de otros servidores de datos.
- También Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Contiene también una base de datos relacional por lo cual también se puede incluir en la licencia de base de base datos multidimensional.
- Es un producto muy maduro con mucha base implantada y muy buena comunidad.

Desventajas

- La principal desventajas seria que contiene una gran cantidad de memoria RAM para la instalación y utilización del software.
- La relación de calidad- precio está muy debajo comparado con Oracle.
- Es imposible utilizar para prácticas porque se prohíben muchas cosas.
- Tiene muchos bloqueos a nivel de página con un tamaño de página fijo y demasiado pequeño.
- También tiene una pésima implementación de los tipos de datos y variables.
- Solo tiene soporte con sistema operativo Windows.
- Contiene límites de conexiones simultáneas para las páginas.
- MSSQL usa Address Windowing Extensión (AWE) para hacer el direccionamiento de 64-bit. Esto le impide usar la administración dinámica de memoria y sólo le permite alojar un máximo de 64GB de memoria compartida.
- MSSQL está atado a la plataforma del sistema operativo sobre la cual se instala.

ESTRATEGIA DE MIGRACIÓN (PLAN DE MIGRACIÓN)

ELABORAR PLAN DE EJECUCION

El plan de Ejecución contiene todas las tareas necesarias para pasar del sistema origen al sistema destino. Entre esas tareas está la implementación de todas las transformaciones que se hayan identificado, tanto de base de datos como de código, pero también la elaboración de todos los planes asociados a la ejecución de la migración, como pueden ser:

Plan de Marcha atrás

Si en el proceso de migración llega a suceder algo imprevisto o por si el cliente decide no realizar la migración, es importante realizar un back up antes de empezar a desarrollar el plan de tramitación, esto es por si pasa cualquier evento que no esté contemplado, y así dar una solución rápida y óptima.

Esta copia será la usada para la migración o importación a SQL Server, usaremos el fichero generado aquí y lo adaptaremos para poder ser ejecutado en SQL Server. Tener en cuenta que la documentación es un tema fundamental en este punto tanto de lo que había como de lo que existe actualmente. Es decir, configuraciones, usuarios, etc.

Plan de Implantación.

Contiene toda la información de los pasos a seguir para la realización del cambio de un SGBD al otro.

- 🌐 Desglose de Tareas de cada paso (implementadores, probadores,...)
- 🌐 Duración prevista de cada paso (Margen máximo de desviación)
- 🌐 Flujo de los pasos (dependencias entre tareas, posibilidad de paralización de las mismas)
- 🌐 Eventos o responsable que toma la decisión de éxito de la migración o marcha atrás de la misma.

PLAN DE EJECUCIÓN

Como se ha mencionado en este documento es fundamental realizar un estudio de funcionamiento de bases de datos en MySQL y SQL Server, para establecer que transformaciones se deben realizar.

A continuación, una vez identificados los trabajos necesarios para realizar la migración y asegurar la funcionalidad del sistema, se identifican las necesidades que generará la migración y se planifican las distintas actividades. En ese momento comenzará la construcción de los planes que se ejecutarán durante el resto del proceso de migración:

Plan de Formación:

Los desarrolladores del programa deben enterarse de cómo hacer las modificaciones pertinentes que se le deben hacer al código una vez esté hecha la migración de la base de datos, ya que al migrar el lenguaje de conexión a base de datos cambia.

Plan de Ejecución:

- Reemplazar los tipos de datos de MySQL con los de SQL Server.
- Cambiar los campos de autoincremento de MySQL (Auto_increment) con los de SQL Server.
- Revisar que todas las tablas hayan sido correctamente migradas.
- Cambiar el código en las consultas a la base de datos por el del nuevo gestor.

Plan de Soporte:

Incluirá todo lo necesario para gestionar las incidencias que los usuarios detecten durante la migración: detalle de los protocolos de comunicación de incidencias, niveles de servicio para la resolución, dedicación del equipo de soporte y demás factores clave en la definición de un servicio de atención a usuarios.

Plan de Comunicación:

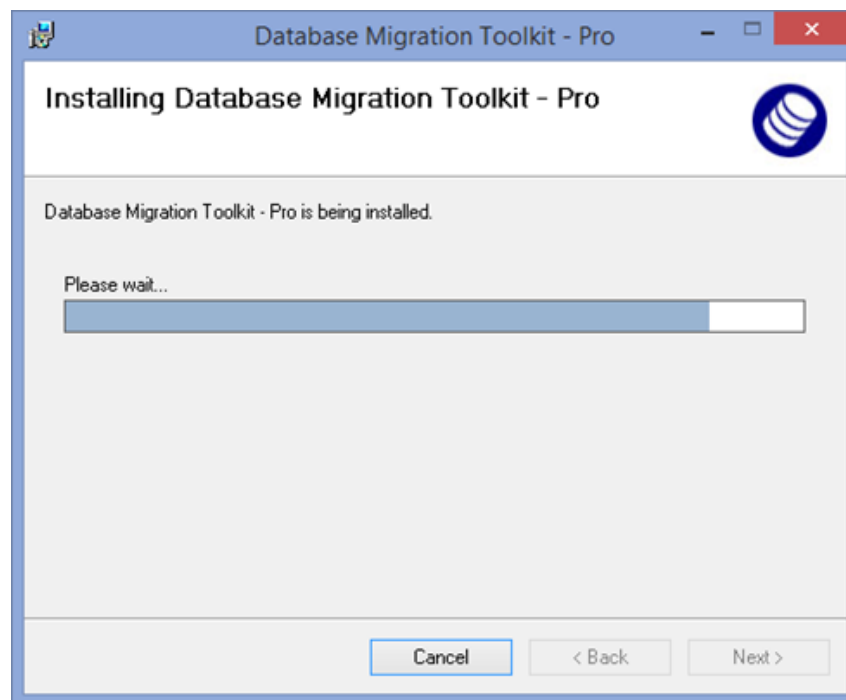
Tras la definición de los planes anteriores, es el momento de diseñar un plan de comunicación que mantenga al usuario final y a los equipos de desarrollo y soporte de la organización con un grado suficiente de información acerca del cambio que se va a producir, cómo informar de incidencias detectadas, etc.

Por último, una vez definidos los Escenarios Origen y Destino y confeccionados los distintos planes, se pone toda la información en común para generar un plan global que llamaremos Plan Estratégico de Migración.

Ejecución de migración

Instalación dbsofts

Una vez iniciado el instalador como administrador empieza la descarga de la herramienta de migración de datos.



Terminado el proceso de instalación se abre la siguiente ventana para configurar la base de datos inicial donde se selecciona la base de datos que quiere ser migrada.

Elige una Fuente de Datos
¿Desde dónde quiere copiar los datos? Puede copiar los datos desde una de las siguientes fuentes.

Fuente: MySQL

Servidor: localhost Puerto: 3306

Nombre de: root

Clave:

Base de Datos: e-commerce

Acerca de Configuración Cargar Tarea << Atrás Siguiente >> Salir

Configuramos y le damos siguiente, se abre la ventana que trae las opciones para configurar la base de datos destino y editar el nombre de base de datos destino

Elige un destino
¿Dónde quiere copiar los datos? Puede copiar datos a cualquiera de los siguientes destinos.

Destino: Microsoft SQL Server

Servidor: localhost\SQLEXPRESS Puerto: 0

Nombre de: sa ☒ Autenticación de Windows

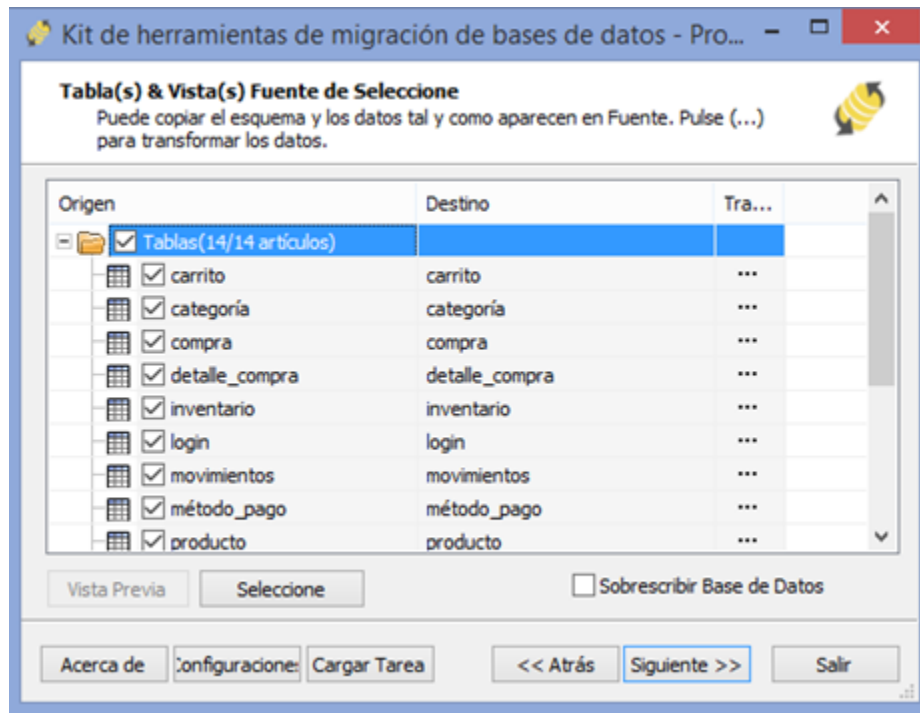
Clave:

Base de Datos: e-commerce

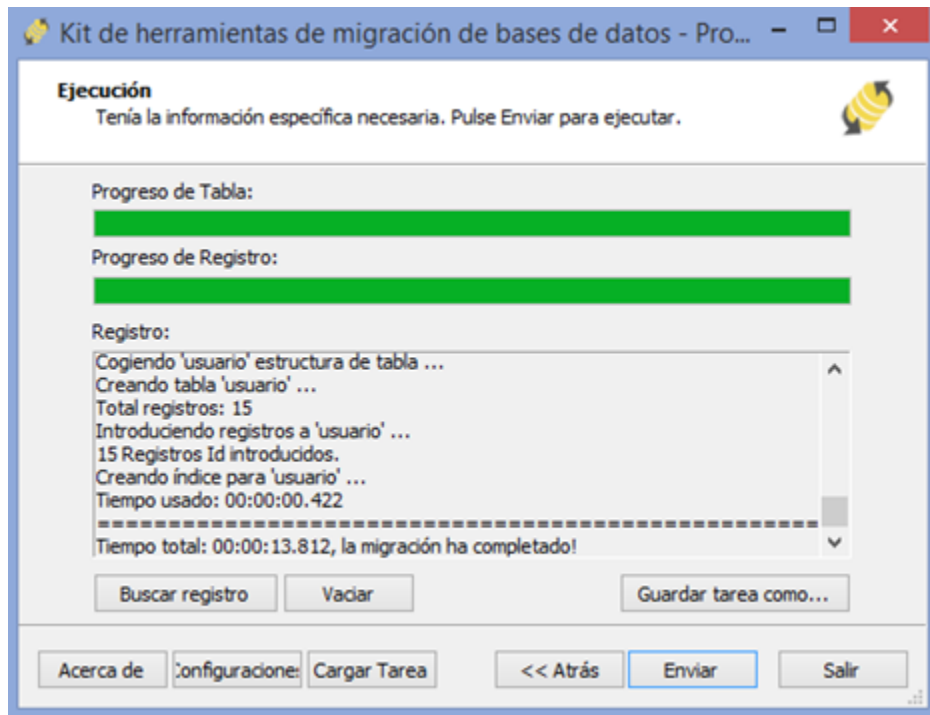
Esquema: dbo

Acerca de Configuración Cargar Tarea << Atrás Siguiente >> Salir

En la siguiente ventana trae todas las tablas de la base de datos de origen y las duplica en el gestor de bases destino, permitiendo modificar los nombres de sus tablas.



Una vez configurado todo le damos y siguiente y procederá a migrar los datos a la otra base de datos con una barra de carga.



Por ultimo la carga se llena completamente y termina de migrar los datos correctamente. Visualizamos la base de datos con tablas y registros migrada en el gestor SQL Server.

