

Administración de Base de Datos SQL Server



Eric Gustavo Coronel Castillo
INSTRUCTOR

CEPSUNI – 2021
LIMA - PERU

Índice

CAPÍTULO 1 INTRODUCCIÓN A SQL SERVER	4
QUÉ ES SQL SERVER	4
VENTAJAS	9
SERVICIOS	10
<i>Contexto</i>	<i>10</i>
<i>Tecnologías de Microsoft SQL Server.....</i>	<i>11</i>
<i>SQL Server Database Engine</i>	<i>12</i>
<i>SQL Server Agent</i>	<i>12</i>
<i>SQL Server Analysis Services.....</i>	<i>12</i>
<i>SQL Server Browser.....</i>	<i>12</i>
<i>SQL Server Integration Services.....</i>	<i>12</i>
<i>SQL Server VSS Witer</i>	<i>13</i>
<i>SQL Server Reporting Services</i>	<i>13</i>
<i>SQL Server Master Data Services.....</i>	<i>13</i>
<i>SQL Server Data Quality Services.....</i>	<i>13</i>
HERRAMIENTAS DE TERCEROS.....	14
<i>Administración de la base de datos de SQL Server.....</i>	<i>14</i>
<i>Monitoreo de rendimiento de SQL Server.....</i>	<i>14</i>
<i>Información de SQL server</i>	<i>14</i>
CAPÍTULO 2 PREPARANDO EL ENTORNO DE TRABAJO	15
EDICIONES DE SQL SERVER.....	15
REQUISITOS GENERALES	17
HERRAMIENTAS.....	18
<i>Herramientas básicas</i>	<i>18</i>
<i>Base de datos de ejemplo</i>	<i>19</i>
<i>SQL Server Management Studio.....</i>	<i>20</i>
<i>Administrador de configuración de SQL Server.....</i>	<i>21</i>
CONECTARSE CON SSMS.....	22
AUTORIZACIÓN DE CONEXIONES ADICIONALES	26
<i>Crear un inicio de sesión con autenticación de Windows.....</i>	<i>26</i>
<i>Crear un inicio de sesión con autenticación SQL Server.....</i>	<i>31</i>
CONEXIÓN REMOTA	34
<i>Contexto</i>	<i>34</i>

<i>Habilitar protocolos</i>	<i>34</i>
<i>Configurar un puerto fijo.....</i>	<i>36</i>
<i>Abrir puertos del firewall</i>	<i>38</i>
<i>Conexión desde otro equipo.....</i>	<i>43</i>
CAPÍTULO 3 TRABAJANDO CON BASES DE DATOS	46
ARQUITECTURA DE BASES DE DATOS SQL SERVER	46
<i>Estructura Lógica</i>	<i>46</i>
<i>Estructura Física</i>	<i>47</i>
CREACIÓN DE UNA BASE DE DATOS.....	49
<i>Contexto</i>	<i>49</i>
<i>Concepto de base de datos</i>	<i>49</i>
<i>Crear base de datos usando T-SQL</i>	<i>50</i>
<i>Creación una base de datos usando SSMS</i>	<i>52</i>
GESTIÓN DE ARCHIVOS.....	54
<i>Grupos de archivos</i>	<i>54</i>
<i>Grupo de archivos predeterminado.....</i>	<i>54</i>
<i>Ejemplo de archivos y grupos de archivos</i>	<i>55</i>
PRACTICA	57
<i>Ejercicio 1</i>	<i>57</i>
<i>Ejercicio 2</i>	<i>58</i>

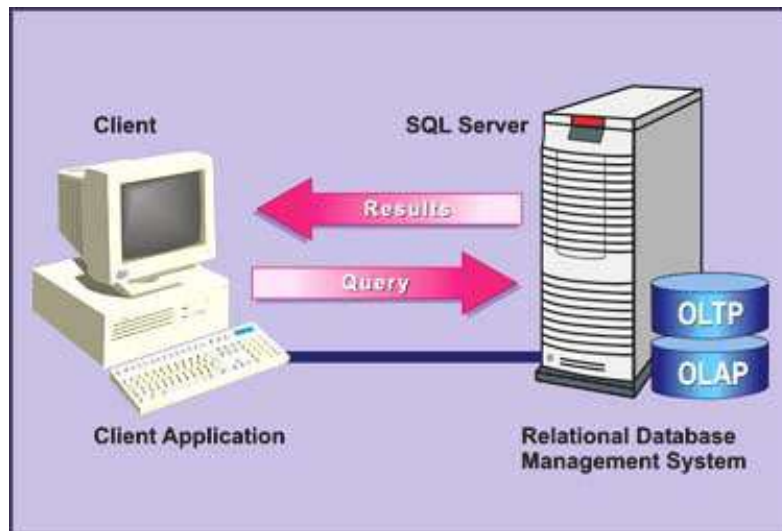
Capítulo 1

INTRODUCCIÓN A SQL SERVER

QUÉ ES SQL SERVER



Las bases de datos son exactamente lo que dice su nombre, un almacén de información que se administra mediante un motor o gestor de base de datos y un lenguaje de base de datos. SQL Server es el gestor de base de datos de Microsoft, que se utiliza para almacenar información trabajando en conjunto con sitios web y otros tipos de aplicaciones. Es un gestor de bases de datos relacionales; cuenta con una interfaz que conecta la base de datos con las aplicaciones en las cuales se genera y se utiliza dicha información.



SQL son las siglas de **S**tructured **Q**uery **L**anguage (lenguaje de consulta estructurado) y es el lenguaje que utilizan los gestores de bases de datos relacionales y el que deben utilizar las aplicaciones para que puedan interactuar con las bases de datos.

<i>id-cliente</i>	<i>nombre-cliente</i>	<i>calle-cliente</i>	<i>ciudad-cliente</i>
19.283.746	González	Arenal	La Granja
01.928.374	Gómez	Carretas	Cerceda
67.789.901	López	Mayor	Peguerinos
18.273.609	Abril	Preciados	Valsain
32.112.312	Santos	Mayor	Peguerinos
33.666.999	Rupérez	Ramblas	León
01.928.374	Gómez	Carretas	Cerceda

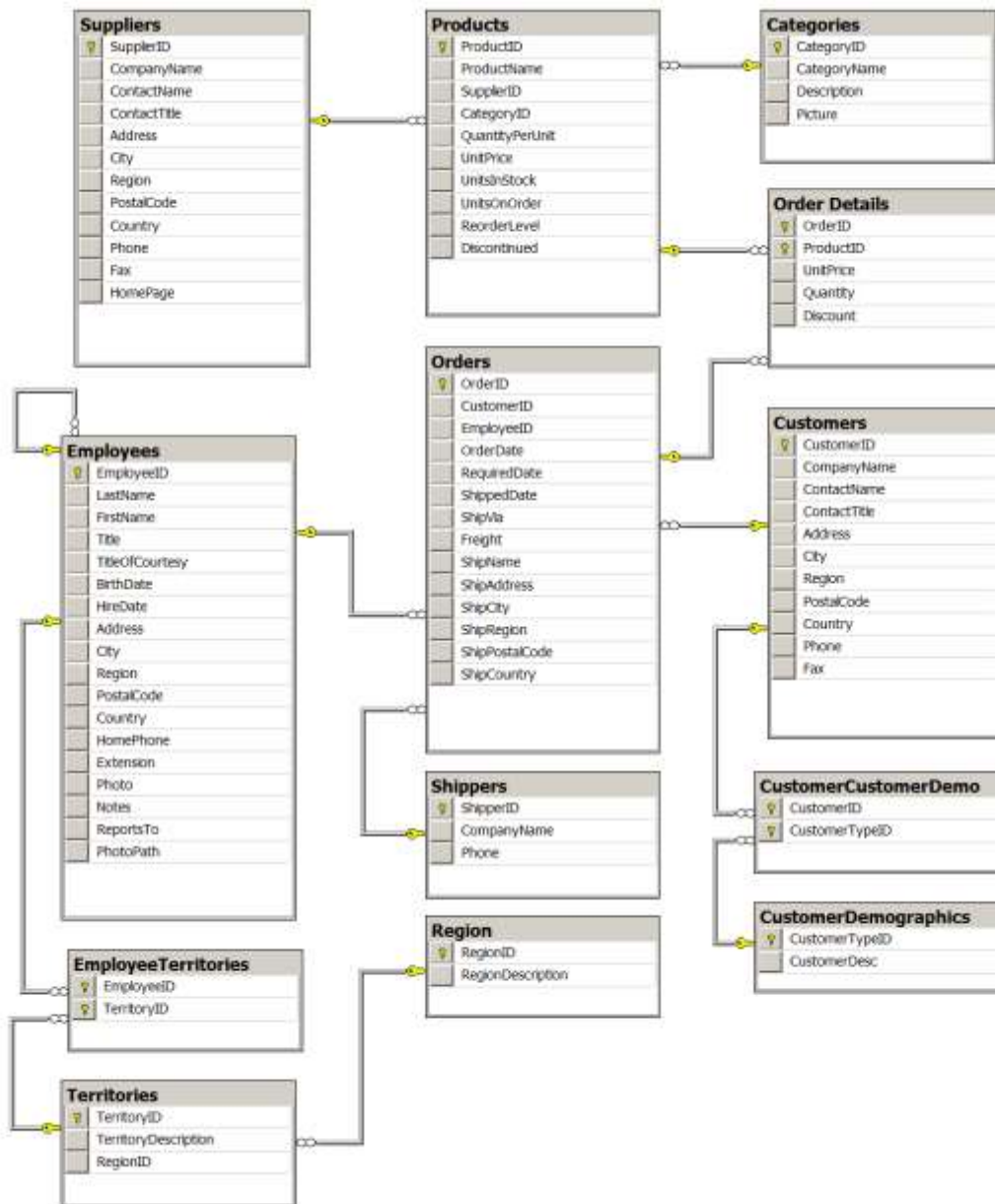
Tabla *cliente*

<i>número-cuenta</i>	<i>saldo</i>	<i>id-cliente</i>	<i>número-cuenta</i>
C-101	500	19.283.746	C-101
C-215	700	19.283.746	C-201
C-102	400	01.928.374	C-215
C-305	350	67.789.901	C-102
C-201	900	18.273.609	C-305
C-217	750	32.112.312	C-217
C-222	700	33.666.999	C-222
		01.928.374	C-201

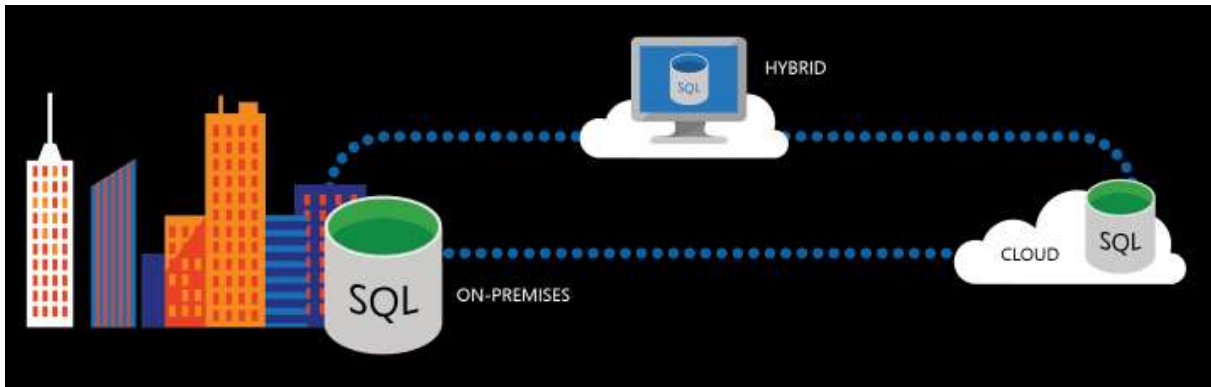
Tabla *cuenta*

Tabla de relación *cliente-cuenta*

El componente principal de una base de datos relacional es la **tabla**, la cual está formada por filas y columnas, y existe relación entre las tablas de una base de datos, lo que permite tener información consistente, los datos de una tabla tienen reglas a las que se les llama restricciones, como por ejemplo clave primaria, clave foránea, clave única, entre otras.



Los conjuntos de tablas de una base de datos se pueden graficar en un diagrama de base de datos, tal como se ilustra en la imagen.



Microsoft está en constante optimización de los servicios de SQL Server, pero también, añadiendo nuevos servicios para responder a los diferentes tipos de necesidades de las empresas y los negocios.

Entre sus ventajas que ofrece SQL Server encontramos las siguientes:

Líder del sector

Cree aplicaciones de misión crítica para procesamiento de transacciones en línea (OLTP) con escalabilidad, rendimiento y alta disponibilidad.

Seguridad avanzada

Proteja los datos en reposo y en movimiento. SQL Server es la base de datos menos vulnerable según la base de datos de vulnerabilidades del NIST.

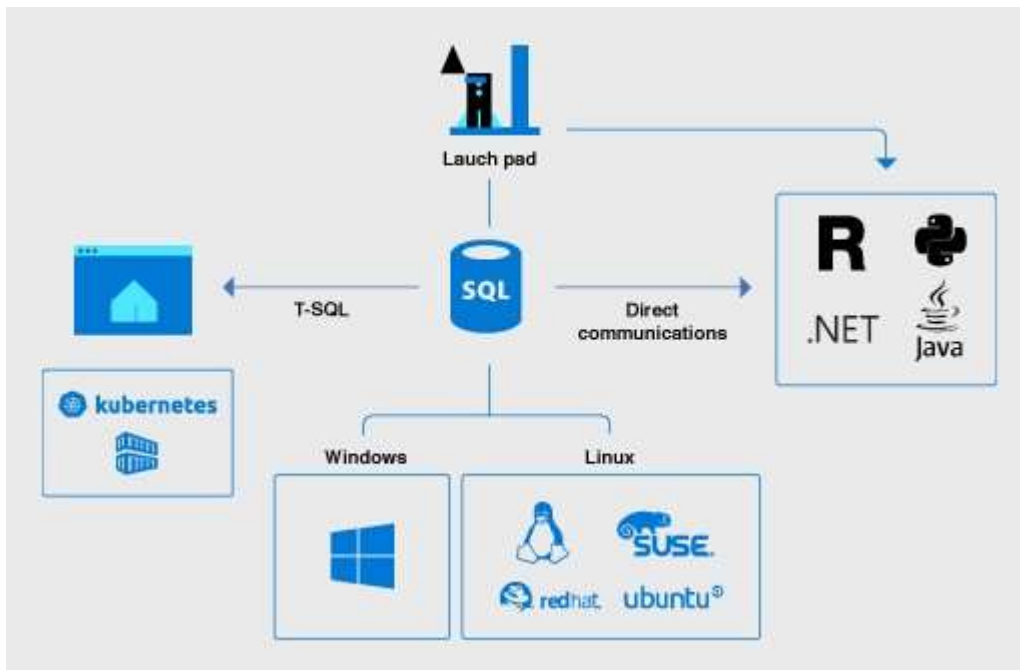
BI móvil integral

Transforme datos en conocimiento práctico. Proporcione información en cualquier dispositivo, con o sin conexión, a una quinta parte del coste de otras soluciones de autoservicio.

Diseñado para la nube híbrida

Obtenga una plataforma y herramientas coherentes para mover más fácilmente las cargas de trabajo entre su centro de datos, la nube privada o Microsoft Azure.

VENTAJAS



A continuación, se describen las ventajas y desventajas:

1. Soporte de transacciones.
2. Permite escalabilidad, estabilidad y seguridad.
3. Soporta procedimientos almacenados.
4. Incluye un potente entorno gráfico de administración, que permite el uso de comandos DDL y DML gráficamente.
5. 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.
6. Permite administrar información de otros servidores de la red desde un solo punto.
7. Es un producto muy maduro con mucha base implantada y muy buena comunidad.

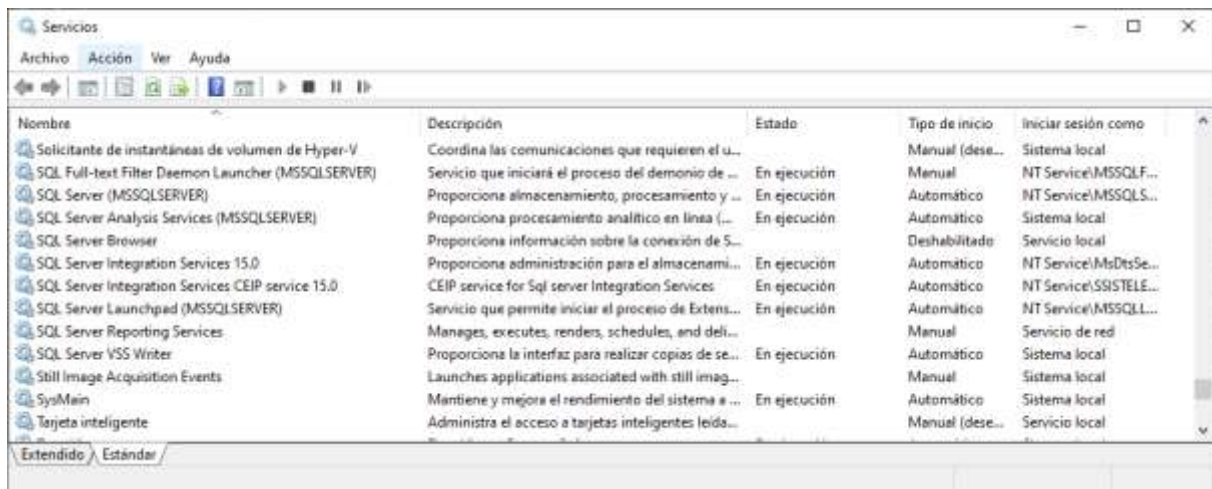
SERVICIOS

Contexto



Tener un repositorio de datos no es suficiente, hoy en día los motores de base de datos están evolucionando y adaptando a las nuevas necesidades que el mercado exige, y SQL Server se puede considerar como una plataforma de muchos servicios centrados en los datos.

Tecnologías de Microsoft SQL Server



The screenshot shows the Windows Services console with a list of services. The columns are: Nombre, Descripción, Estado, Tipo de inicio, and Iniciar sesión como. The services listed include Solicitante de instantáneas de volumen de Hyper-V, SQL Full-text Filter Daemon Launcher (MSSQLSERVER), SQL Server (MSSQLSERVER), SQL Server Analysis Services (MSSQLSERVER), SQL Server Browser, SQL Server Integration Services 15.0, SQL Server Integration Services CEIP service 15.0, SQL Server Launchpad (MSSQLSERVER), SQL Server Reporting Services, SQL Server VSS Writer, Still Image Acquisition Events, SysMain, and Tarjeta inteligente.

Nombre	Descripción	Estado	Tipo de inicio	Iniciar sesión como
Solicitante de instantáneas de volumen de Hyper-V	Coordina las comunicaciones que requieren el u...		Manual (dese...	Sistema local
SQL Full-text Filter Daemon Launcher (MSSQLSERVER)	Servicio que iniciará el proceso del demonio de ...	En ejecución	Manual	NT Service\MSSQLF...
SQL Server (MSSQLSERVER)	Proporciona almacenamiento, procesamiento y ...	En ejecución	Automático	NT Service\MSSQLS...
SQL Server Analysis Services (MSSQLSERVER)	Proporciona procesamiento analítico en línea (...)	En ejecución	Automático	Sistema local
SQL Server Browser	Proporciona información sobre la conexión de S...		Deshabilitado	Servicio local
SQL Server Integration Services 15.0	Proporciona administración para el almacenami...	En ejecución	Automático	NT Service\MSDtsSe...
SQL Server Integration Services CEIP service 15.0	CEIP service for Sql server Integration Services	En ejecución	Automático	NT Service\SSISTELE...
SQL Server Launchpad (MSSQLSERVER)	Servicio que permite iniciar el proceso de Extens...	En ejecución	Automático	NT Service\MSSQLL...
SQL Server Reporting Services	Manages, executes, renders, schedules, and deli...		Manual	Servicio de red
SQL Server VSS Writer	Proporciona la interfaz para realizar copias de se...	En ejecución	Automático	Sistema local
Still Image Acquisition Events	Launches applications associated with still imag...		Manual	Sistema local
SysMain	Mantiene y mejora el rendimiento del sistema e ...	En ejecución	Automático	Sistema local
Tarjeta inteligente	Administra el acceso a tarjetas inteligentes leída...		Manual (dese...	Servicio local

Microsoft SQL Server es un conjunto completo de tecnologías y herramientas de administración y análisis de datos para el entorno empresarial.

Cuando se instala SQL Server se habilitan una serie de servicios o programas que se ejecutan en segundo plano y el administrador de base de datos debe ser capaz de identificarlos para gestionarlos adecuadamente.

- SQL Server Database Engine
- SQL Server Agent
- SQL Server Analysis Services
- SQL Server Browser
- SQL Server Integration Services
- SQL Server VSS Writer
- SQL Server Reporting Services
- Master Data Services (Servicios de Datos Maestros)
- Data Quality Services (Servicios de Calidad de Datos)
- Replication (Replicación ó Duplicación)

SQL Server Database Engine

SQL Server Database Engine (Motor de Base de Datos) es el servicio básico del producto y se encarga de las operaciones para el almacenamiento, procesamiento y manejo de la seguridad de los datos. Controla el acceso de las aplicaciones a la base de datos y ejecuta las transacciones.

El motor de base de datos le permite crear bases de datos relacionales para procesamiento de transacciones en línea (OLTP) o procesamiento analítico de datos en línea (OLAP), incluyendo la creación de los objetos de la base de datos: tablas, índices, vistas, procedimientos almacenados, y usuarios.

SQL Server Agent

Este servicio se encarga de controlar las tareas programadas y dispara las replicas que se encuentren habilitadas.

SQL Server Analysis Services

SQL Server Analysis Services (SSAS) te permite construir y desplegar bases de datos para análisis como soporte a las soluciones de inteligencia de negocios.

SSAS te permite diseñar, construir y administrar bases de datos multidimensionales con data consolidada de las bases de datos transaccionales u operacionales y de otros orígenes de datos, utilizando la técnica de datos agregados.

SQL Server Browser

Servicio que facilita la conexión con los clientes, por ejemplo, permitiendo la conexión sin especificar el puerto.

SQL Server Integration Services

SQL Server Integration Services (SSIS) es una plataforma que brinda la posibilidad de crear soluciones de integración de datos a nivel empresarial. Permite migrar datos desde diversos orígenes, transformarlos para estandarizarlos, y almacenarlos en un repositorio común. Una de sus aplicaciones más comunes es la actualización de datawarehouses.

SSIS se basa en la construcción de paquetes que contienen las tareas a ejecutar. Los paquetes permiten programar tareas casi sin escribir líneas de código.

SQL Server VSS Witer

Servicio que permite tener acceso a los archivos de datos, incluso cuando esté corriendo la instancia.

SQL Server Reporting Services

SQL Server Reporting Services (SSRS) es una plataforma que a través de un conjunto de herramientas y servicios permite acceder a una variedad de orígenes de datos para generar y publicar reportes en diferentes formatos.

Incluye APIs que permiten a los desarrolladores crear reportes a la medida de las necesidades de los usuarios.

SQL Server Master Data Services

SQL Server Master Data Services (MDS) es la solución SQL Server para la administración de la data maestra. Mediante Master Data Management (MDM) puedes definir listas de datos no transaccionales para mantenimiento de la data maestra.

Master Data Services incluye jerarquías, seguridad granular, transacciones, control de versiones de datos y reglas de negocios.

SQL Server Data Quality Services

SQL Server Data Quality Services (SSDQS) es una solución basada en el conocimiento de la calidad de datos. Te permite la construcción de una base de conocimiento de calidad de los datos (Data Quality Knowledge Base o DQKB) y utilizarla para realizar varias tareas de calidad de datos críticas como corrección, enriquecimiento, normalización y consolidación.

HERRAMIENTAS DE TERCEROS

Administración de la base de datos de SQL Server

Mejorar considerablemente la productividad de los DBA, asegurarse un rendimiento máximo de la base de datos y mitigar los riesgos de los cambios. Las herramientas de administración de base de datos **Toad for SQL Server** y **ApexSQL** proporcionan capacidades líderes de la industria y flujos de trabajo intuitivos para administrar fácilmente más bases de datos.

Monitoreo de rendimiento de SQL Server

Optimice fácilmente e implemente ajustes con **Spotlight on SQL Server Enterprise**. Garantice el máximo rendimiento gracias al monitoreo, el diagnóstico y la optimización inigualables. Obtenga acceso a los detalles que necesita para un óptimo ajuste de SQL Server. Obtenga información general intuitiva sobre el estado y el rendimiento, las acciones y alertas automatizadas, y el soporte para los dispositivos móviles.

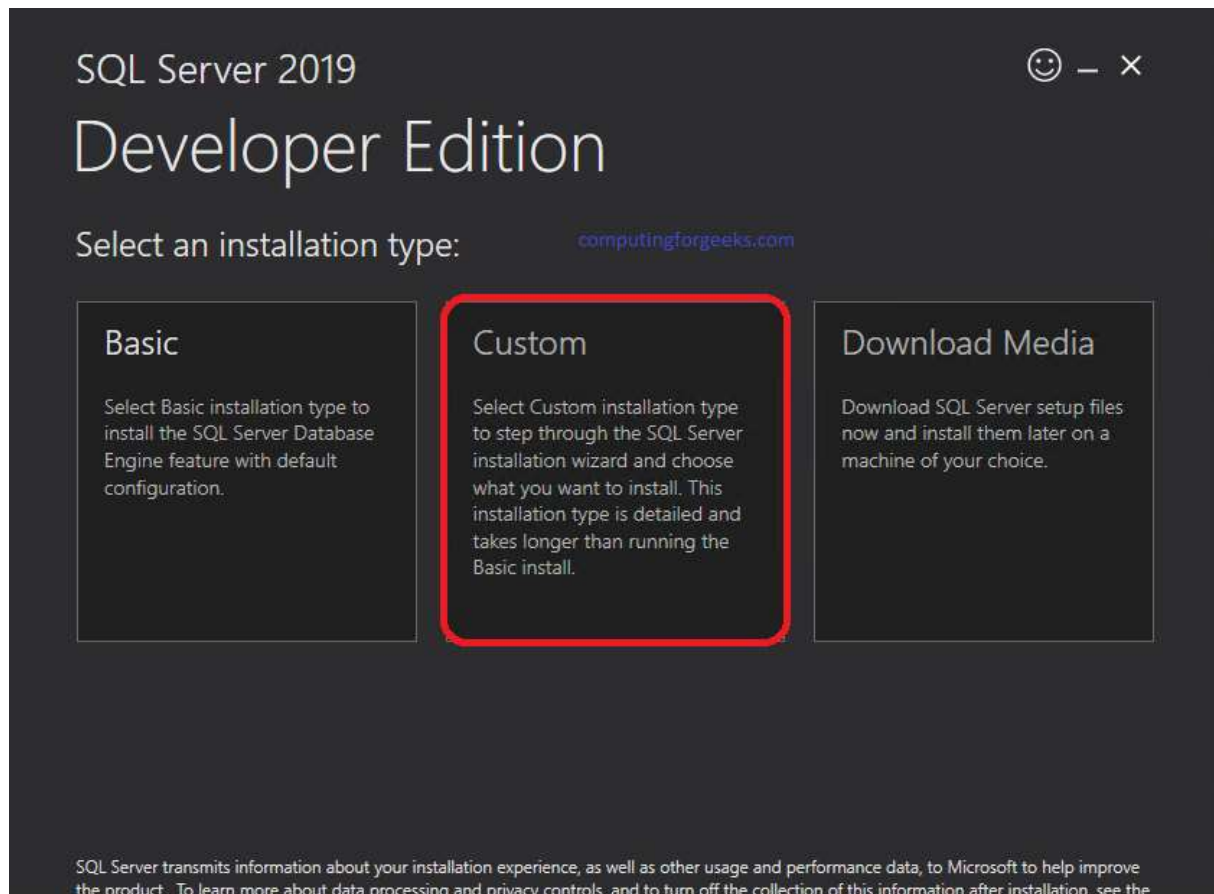
Información de SQL server

Utilice **Foglight** para monitorear fácilmente entornos de bases de datos complejos y diversos. Obtenga información valiosa de todas las áreas de los recursos de la base de datos, análisis de la carga de trabajo y control de cambios. Reciba un panorama completo de sus entornos multiplataforma, incluidos código abierto y NoSQL, para que pueda colaborar con todos los equipos a fin de encontrar y solucionar los problemas de rendimiento, antes de que pongan en riesgo su base de datos.

Capítulo 2

PREPARANDO EL ENTORNO DE TRABAJO

EDICIONES DE SQL SERVER



Estas son las ediciones disponibles de SQL Server:

Express : Edición gratuita. Incluye funcionalidad del core-RDBMS. Limitado a 10G de tamaño de disco. Ideal para desarrollo y pruebas.

Edición Estándar : Incluye funcionalidad central y capacidades de Business Intelligence.

Edición Enterprise : Edición SQL Server con todas las funciones. Incluye capacidades avanzadas de seguridad y almacenamiento de datos.

Edición Developer : Incluye todas las características de la Edición Enterprise y sin limitaciones, y se puede descargar y utilizar de forma gratuita solo con fines de desarrollo.

Después de descargar/adquirir SQL Server, la instalación se ejecuta con **SQLSetup.exe**, que está disponible como una GUI o un programa de línea de comandos.

La instalación a través de cualquiera de estos requerirá que especifique una clave de producto y ejecute alguna configuración inicial que incluya funciones habilitadas, servicios separados y la configuración de los parámetros iniciales para cada uno de ellos. Los servicios y características adicionales se pueden habilitar en cualquier momento ejecutando el programa SQLSetup.exe en la línea de comandos o en la versión de la GUI.

El lenguaje de desarrollo de SQL Server es **Transact-SQL** (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crear tablas y definir relaciones entre ellas (DDL).

REQUISITOS GENERALES

Para que tengas un funcionamiento aceptable del SQL Server se realizan las siguientes recomendaciones:

Característica	Mínimo	Recomendable
Procesador	Core i3, AMD Athlon	Intel Core i5, AMD Ryzen o superiores
Memoria RAM	4 GB	16 GB
Espacio en disco duro	512 MB	2 GB
Sistema operativo	Para escritorio Windows 10 Pro	

Opciones de instalación recomendadas:

Opción	Valor
Tipo de instancia	Instancia por defecto
Modo de autenticación	Mixta
Analysis Services	Multidimensional

Para más información, consultar el siguiente enlace:

<https://docs.microsoft.com/es-es/sql/database-engine/install-windows/install-sql-server>

HERRAMIENTAS

Motor de base de datos de SQL Server incluye varias herramientas. En este documento se describen las primeras herramientas que necesitará para empezar a trabajar con este motor de base de datos.

Se puede obtener acceso a todas las herramientas desde el menú Inicio. No siempre se instalan las mismas herramientas, como SQL Server Management Studio, de forma predeterminada. Debe seleccionarlas como parte de los componentes cliente durante la instalación.

Herramientas básicas

- **SQL Server Management Studio**

Es la herramienta principal para administrar el motor de base de datos y escribir código Transact-SQL. Se encuentra en el shell de Visual Studio. No se incluye en SQL Server Express, pero está disponible como descarga independiente en el Centro de descarga de Microsoft.

- **SQL Server Configuration Manager**

Se instala con SQL Server y las herramientas cliente. Permite habilitar protocolos de servidor, configurar opciones de protocolo, como los puertos TCP, configurar servicios de servidor para que se inicien automáticamente y configurar equipos cliente para que se conecten de la forma preferida. Esta herramienta configura los elementos de conectividad más avanzados, pero no habilita las características.

Base de datos de ejemplo

Las bases de datos de ejemplo no están incluidas en el proceso de instalación de SQL Server.

La mayoría de los ejemplos descritos en los Libros en pantalla usan la base de datos ejemplo **AdventureWorks**, si vas a usar esta base de datos debes descargarla e instalarla para que puedas tenerla en tu servidor.

A continuación, tienes el enlace desde donde puedes descargar los scripts de las bases de datos ejemplo:

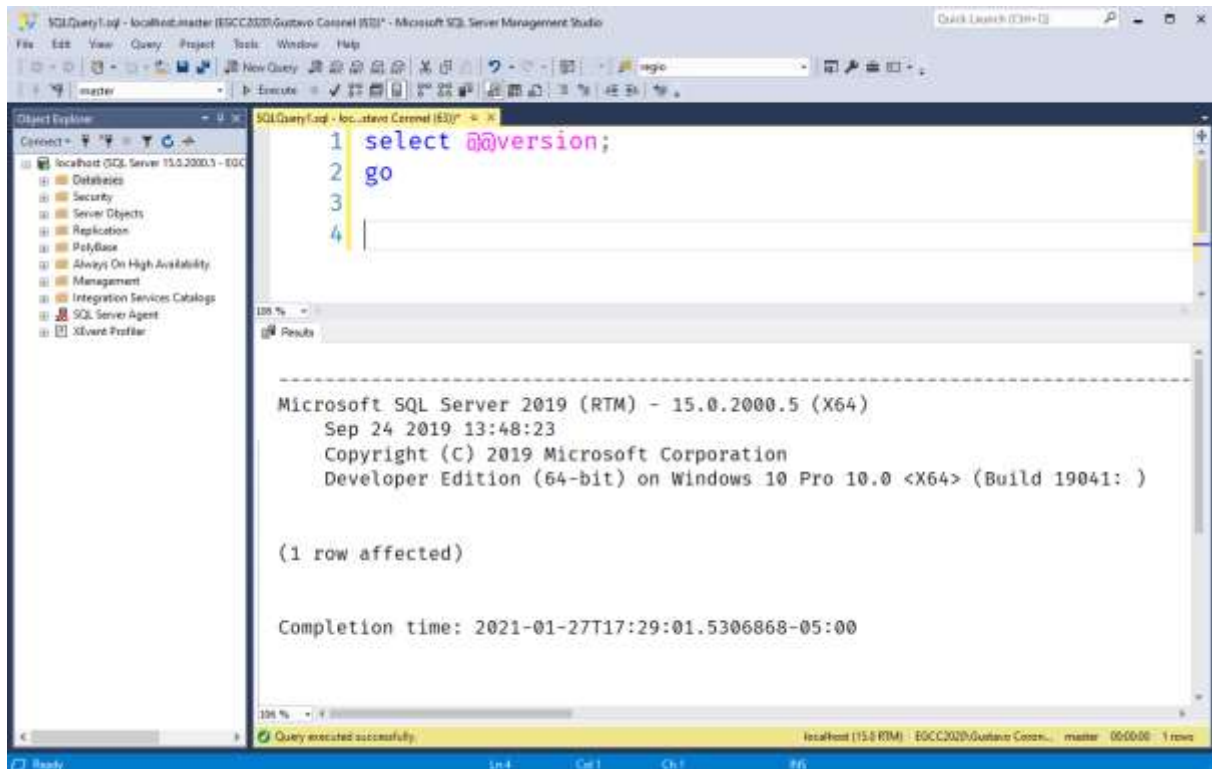
<https://github.com/microsoft/sql-server-samples/tree/master/samples/databases>

En el siguiente repositorio también vas a encontrar scripts para crear bases de datos ejemplo:

<https://github.com/gcoronelc/databases>

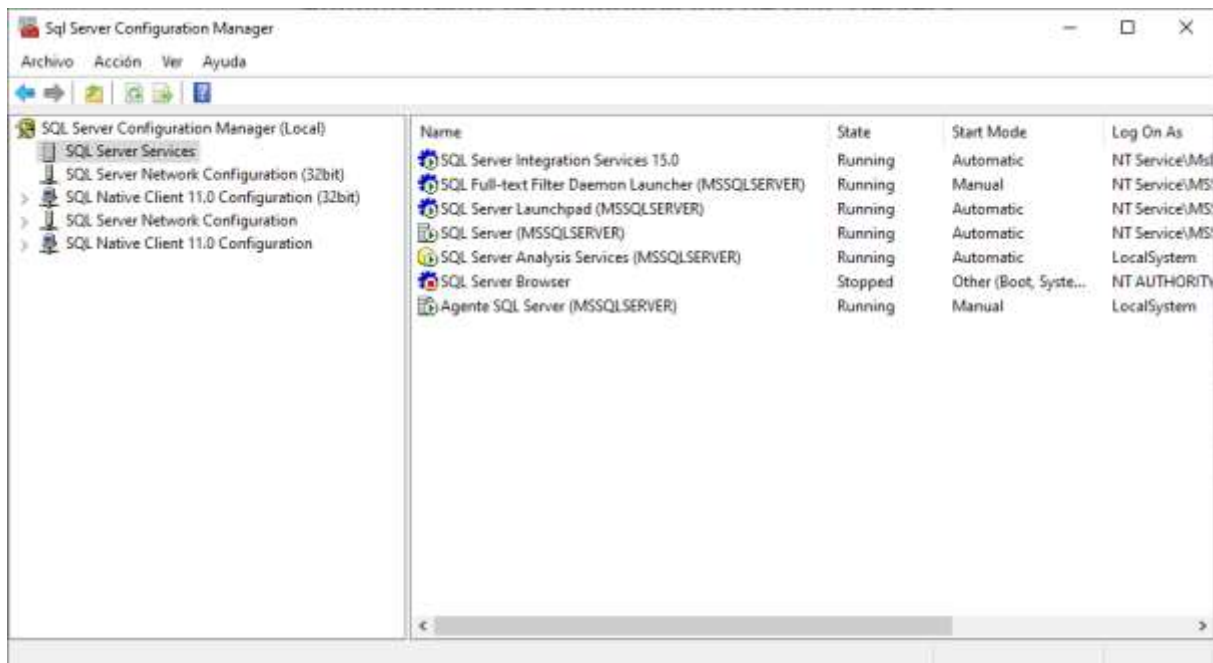
SQL Server Management Studio

En el menú **Inicio**, debes buscar y ejecutar **SQL Server Management Studio**.



Administrador de configuración de SQL Server

En el menú **Inicio**, debes buscar y ejecutar **Administrador de configuración de SQL Server**.



Esta herramienta te permite:

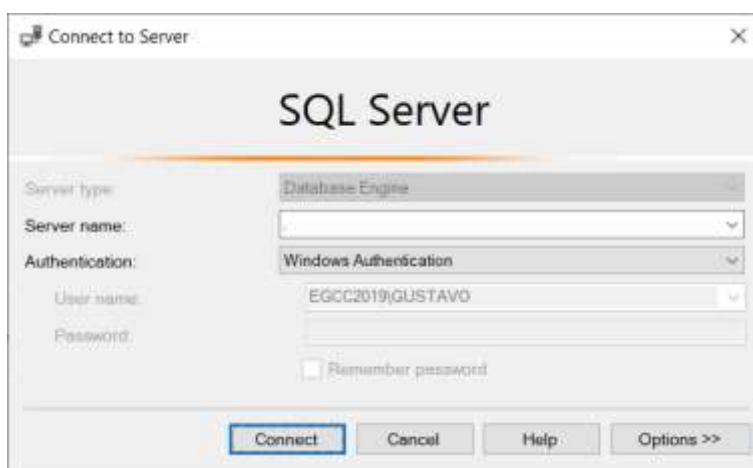
- Gestionar los servicios del servidor.
- Gestionar los protocolos de acceso al servidor.

CONECTARSE CON SSMS

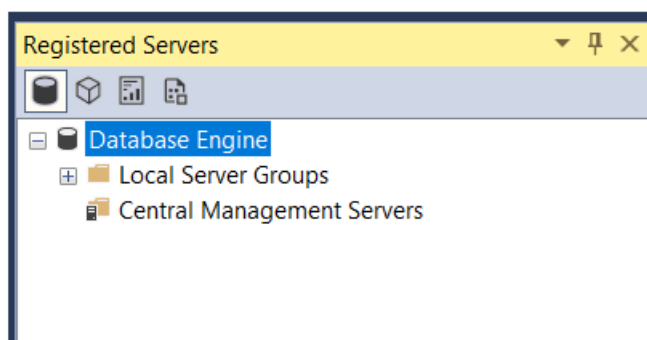
Resulta sencillo conectarse al Motor de base de datos desde herramientas que se ejecutan en el mismo equipo si conoce el nombre de la instancia y si se conecta como miembro del grupo Administradores del equipo. Los procedimientos siguientes deben realizarse en el mismo equipo en el que se ha instalado SQL Server.

Para determinar el nombre de la instancia de motor de base de datos:

1. Inicia una sesión en **Windows** como miembro del grupo **Administradores** y ejecuta **SQL Server Management Studio**.
2. En el cuadro de diálogo **Conectar con el servidor**, hacer clic en **Cancelar**.

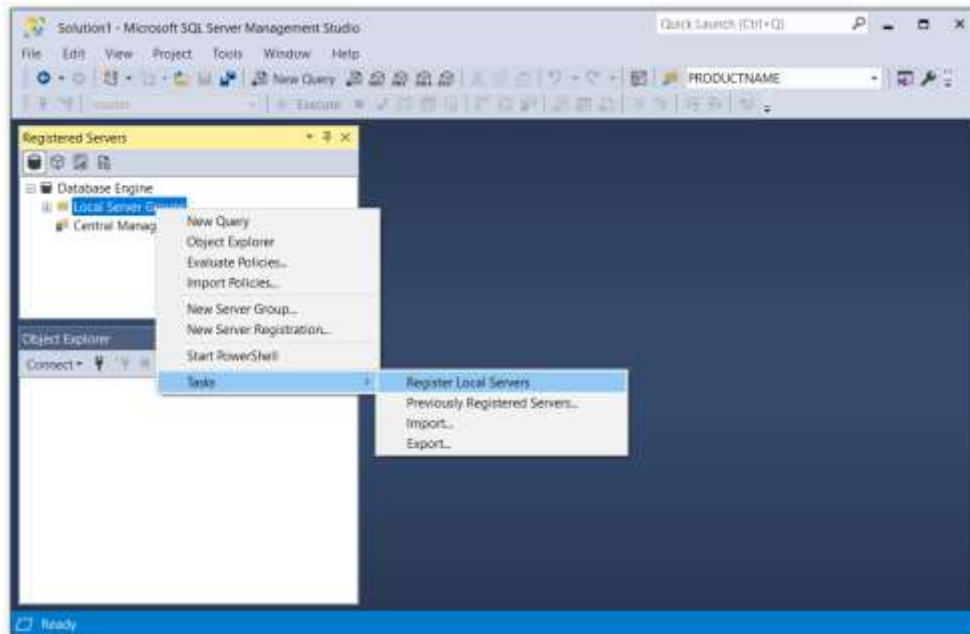


3. Si no aparece la vista **Servidores registrados**, en el menú **Ver**, debes hacer click en **Servidores registrados**.

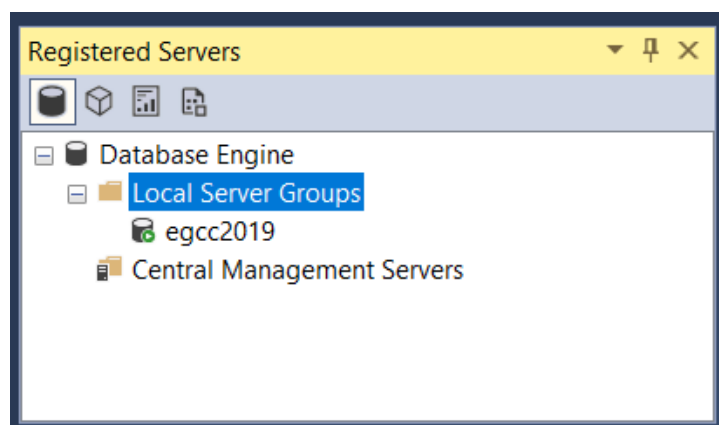


4. Con la opción **Motor de base de datos** seleccionado en la barra de herramientas de la vista **Servidores registrados**, expandir **Motor de base de**

datos, hacer clic con el botón secundario en **Grupos de servidores locales**, seleccione **Tareas** y, a continuación, hacer clic en **Registrar servidores locales**.

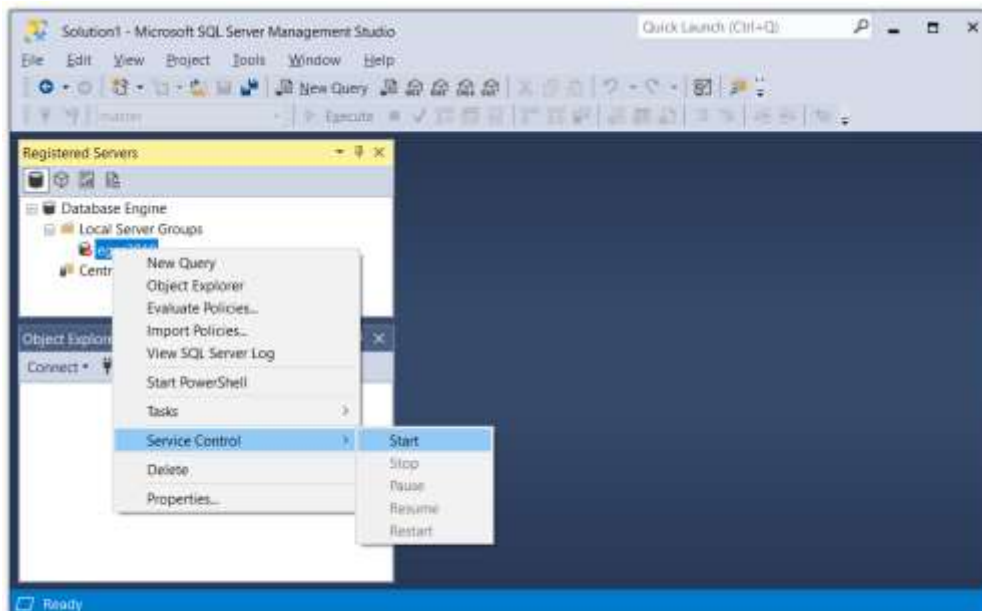


5. Se muestran todas las instancias de Motor de base de datos instaladas en el equipo. La instancia predeterminada no tiene nombre y aparece como el nombre del equipo. Una instancia con nombre aparece como el nombre del equipo seguido de una barra inversa (\) y del nombre de la instancia. En SQL Server Express, la instancia se denomina **<nombre_equipo>\sqlexpress** a no ser que se haya cambiado el nombre durante la instalación.



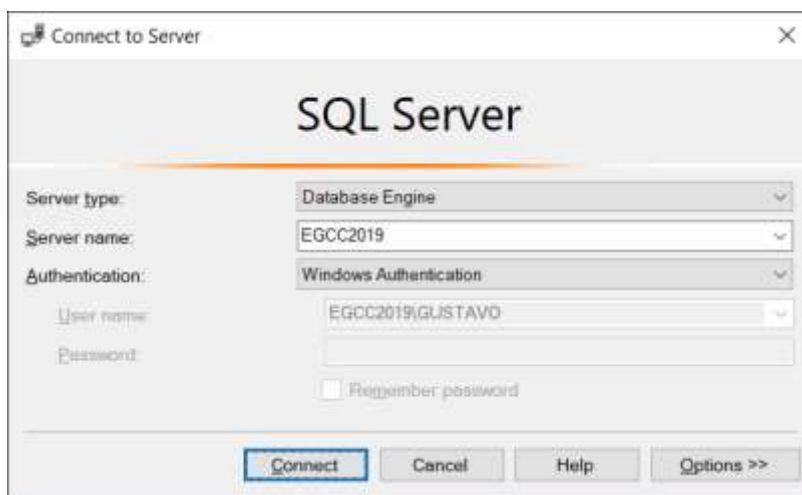
Para comprobar que el motor de base de datos está en ejecución:

6. En **Servidores registrados**, si el nombre de la instancia de SQL Server tiene un punto verde con una flecha blanca junto al nombre, el motor de base de datos está en ejecución y no es necesario realizar ninguna otra acción.
7. Si el nombre de la instancia de SQL Server tiene un punto rojo con un cuadrado blanco junto al nombre, el motor de base de datos se encuentra detenido. Haga clic con el botón secundario en el nombre de motor de base de datos, haga clic en **Control de servicios** y, a continuación, haga clic en **Iniciar**. Después de un cuadro de diálogo de confirmación, el motor de base de datos debería iniciarse y el color del punto debería cambiar a verde con una flecha blanca.

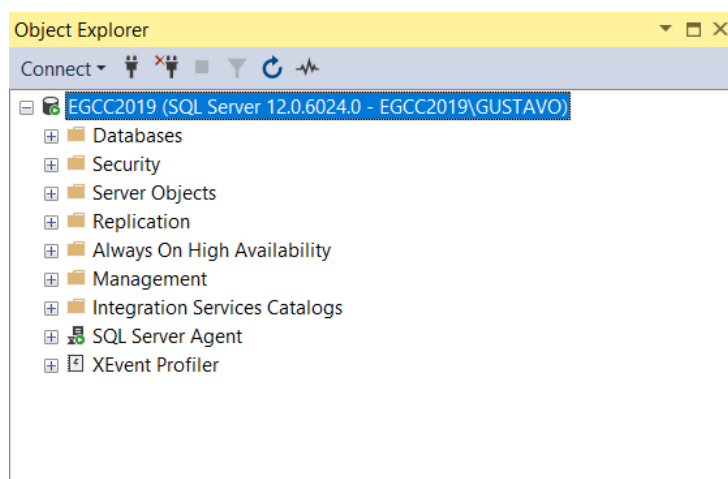


Para conectarse al motor de base de datos:

1. En **Management Studio**, en el menú **Archivo**, haga clic en **Conectar Explorador de objetos**. Se abre el cuadro de diálogo **Conectar con el servidor**. En el control **Tipo de servidor** se muestra el último tipo de componente utilizado.



2. Seleccione **Motor de base de datos**.
3. En el control **Nombre del servidor**, escriba el nombre de la instancia del motor de base de datos. Para la instancia predeterminada de SQL Server, el nombre de servidor es el nombre del equipo. Para una instancia con nombre de SQL Server, el nombre del servidor tiene el siguiente formato: **<nombre_equipo>\<nombre_instancia>**, a continuación, tiene un ejemplo: **EGCC2019\SQLEXPRESS**.
4. Hacer click en **Conectar**.



AUTORIZACIÓN DE CONEXIONES ADICIONALES

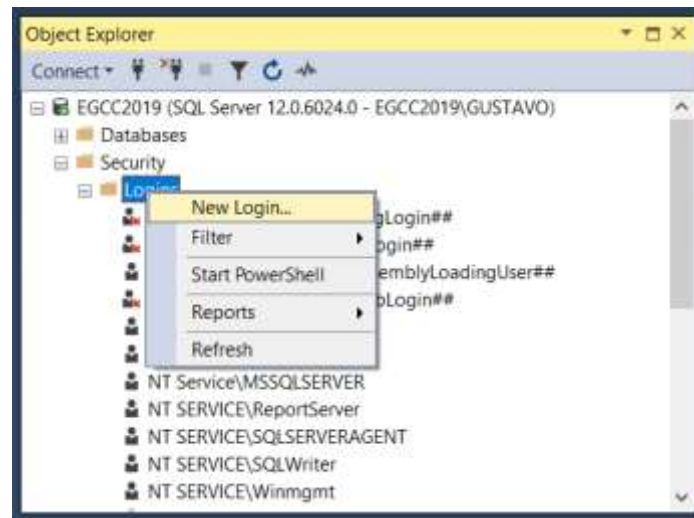
Ahora que te has conectado a SQL Server como administrador, una de las primeras tareas que debes realizar es autorizar a otros usuarios a conectarse. Para ello debes crear un inicio de sesión y concédale autorización para obtener acceso a una base de datos como usuario.

Los inicios de sesión pueden ser inicios de sesión con autenticación de Windows, que utilizan las credenciales de Windows, o bien inicios de sesión con autenticación de SQL Server, que almacenan la información de autenticación en SQL Server y son independientes de las credenciales de Windows.

Se recomienda utilizar la autenticación de Windows siempre que sea posible.

Crear un inicio de sesión con autenticación de Windows

1. En la tarea anterior, se conectó al motor de base de datos utilizando Management Studio. En el **Explorador de objetos**, expanda la instancia del servidor, expanda **Seguridad**, haga clic con el botón secundario en **Inicios de sesión** y, a continuación, haga clic en **Nuevo inicio de sesión**.



2. Aparecerá el cuadro de diálogo **Inicio de sesión - Nuevo**.

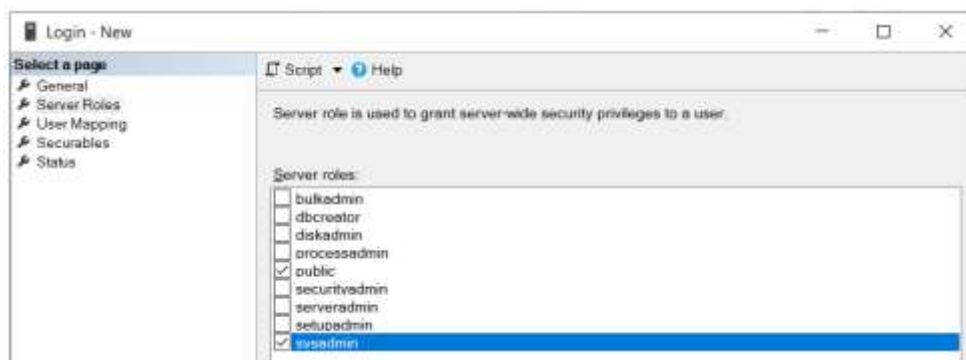
3. En la página **General**, en el control **Nombre de inicio de sesión**, escriba un inicio de sesión de Windows con el formato **<domain>\<login>**.



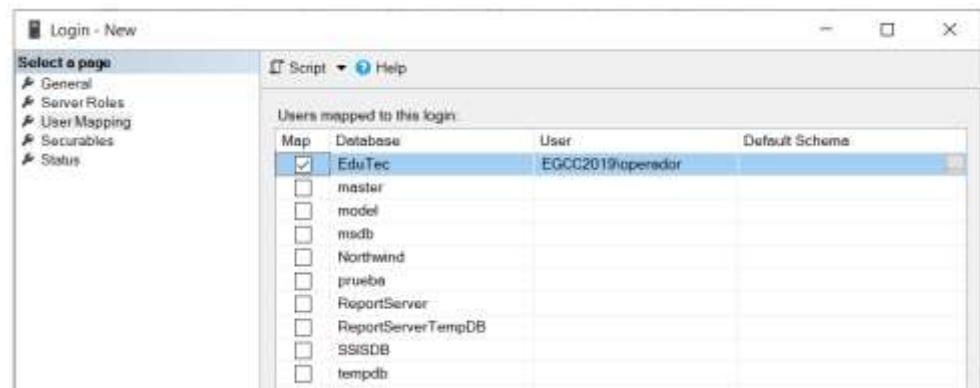
4. En el control **Base de datos predeterminada**, seleccione **AdventureWorks**, si está disponible. Si no lo está, seleccione **master**.



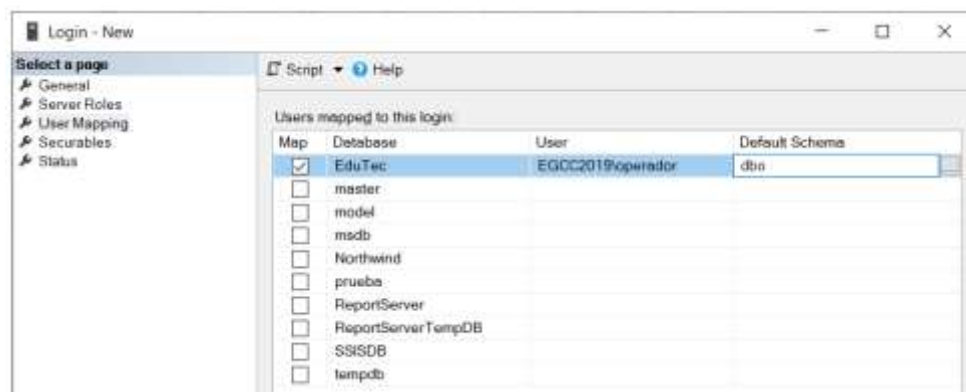
5. En la página **Roles del servidor**, si el nuevo inicio de sesión va a ser administrador, haga clic en **sysadmin**; de lo contrario, déjelo en blanco.



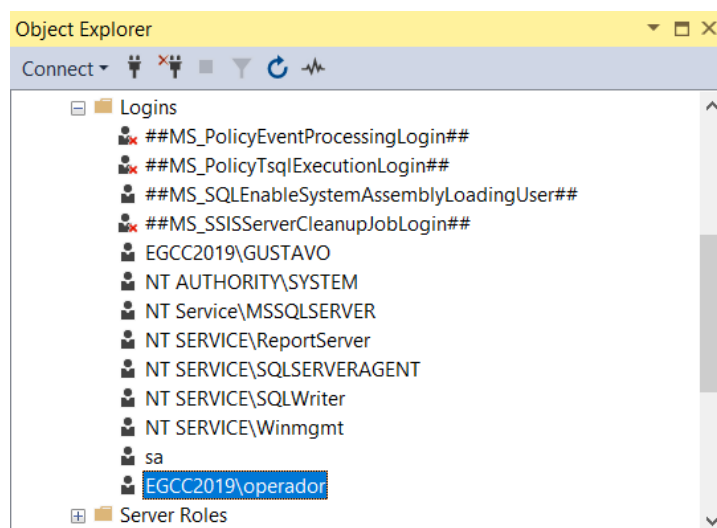
6. En la página **Asignación de usuarios**, seleccione **Asignar** para la base de datos **EduTec**, si está disponible. Si no lo está, seleccione master. Observe que en la columna **Usuario** se ha rellenado con el inicio de sesión. Al cerrar el cuadro de diálogo, se creará el usuario en la base de datos.



7. En el cuadro **Esquema predeterminado**, escriba **dbo** para asignar al usuario el esquema del propietario de la base de datos.

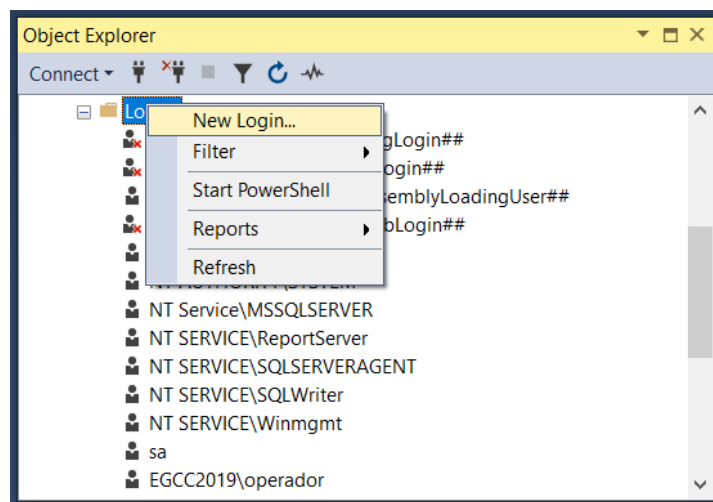


8. Acepte los valores predeterminados de los cuadros **Elementos protegibles** y **Estado** y haga clic en **Aceptar** para crear el inicio de sesión.

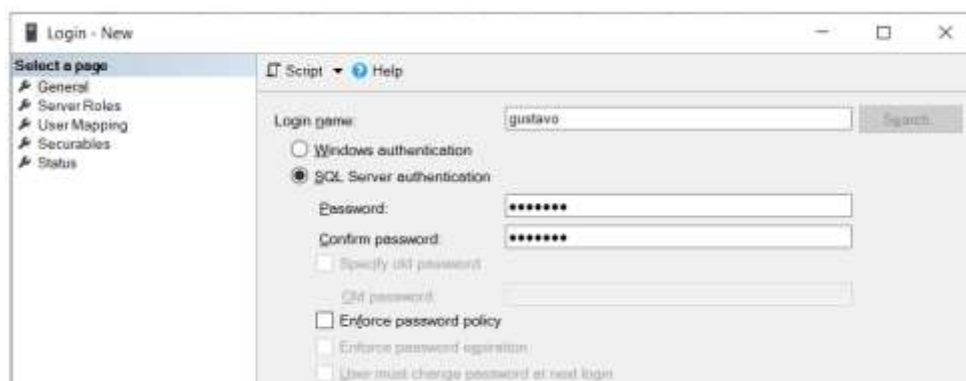


Crear un inicio de sesión con autenticación SQL Server

1. En el **Explorador de objetos**, expanda la instancia de servidor, expanda **Seguridad**, haga clic con el botón secundario en **Inicios de sesión** y, a continuación, haga clic en **Nuevo inicio de sesión**.



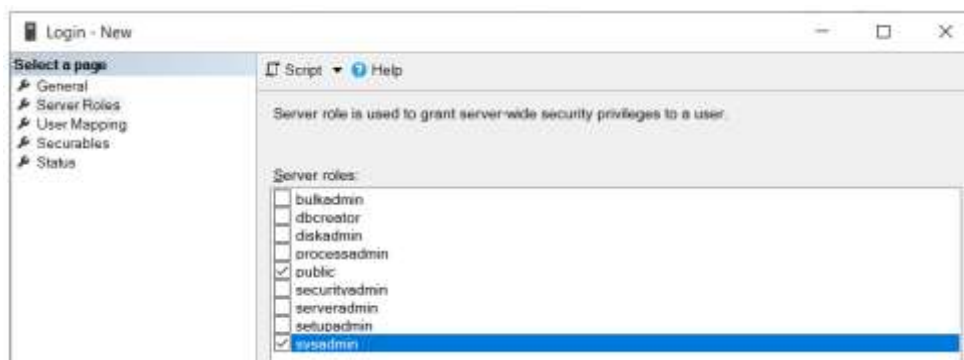
2. Aparecerá el cuadro de diálogo **Inicio de sesión - Nuevo**.
3. En la página **General**, en el cuadro **Nombre de inicio de sesión**, escriba un inicio de sesión, por ejemplo, **gustavo**. Seleccione **Autenticación de SQL Server** y escriba una contraseña para el nuevo inicio de sesión, también debe deshabilitar la opción **Exigir expiración de contraseña**.



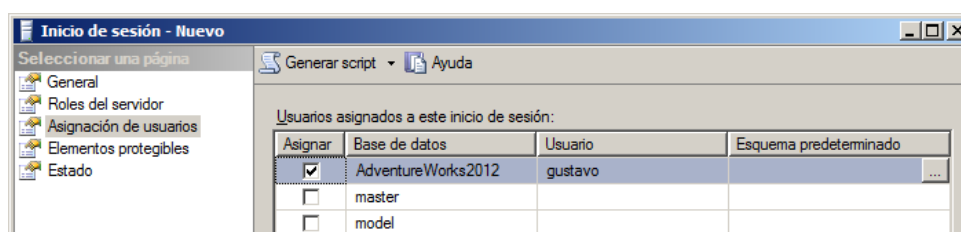
4. En el cuadro **Base de datos predeterminada**, seleccione **EduTec**, si está disponible. Si no lo está, seleccione master.



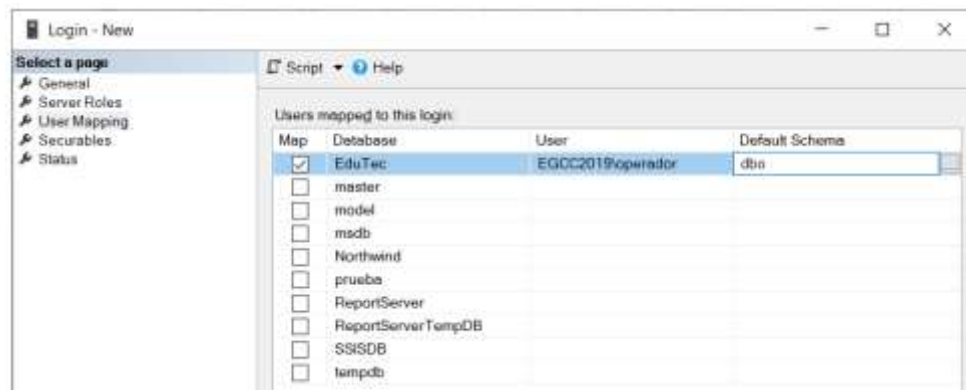
5. En la página **Roles del servidor**, si el nuevo inicio de sesión va a ser administrador, haga clic en **sysadmin**; de lo contrario, déjelo en blanco.



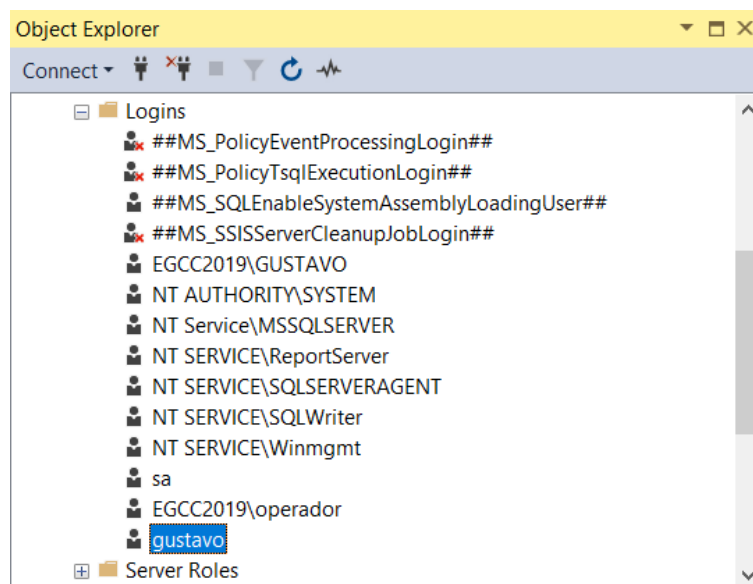
6. En la página **Asignación de usuarios**, seleccione **Asignar** para la base de datos EduTec, si está disponible. si no lo está, seleccione master. Observe que el cuadro **Usuario** se ha rellenado con el inicio de sesión. Al cerrar el cuadro de diálogo, se creará el usuario en la base de datos.



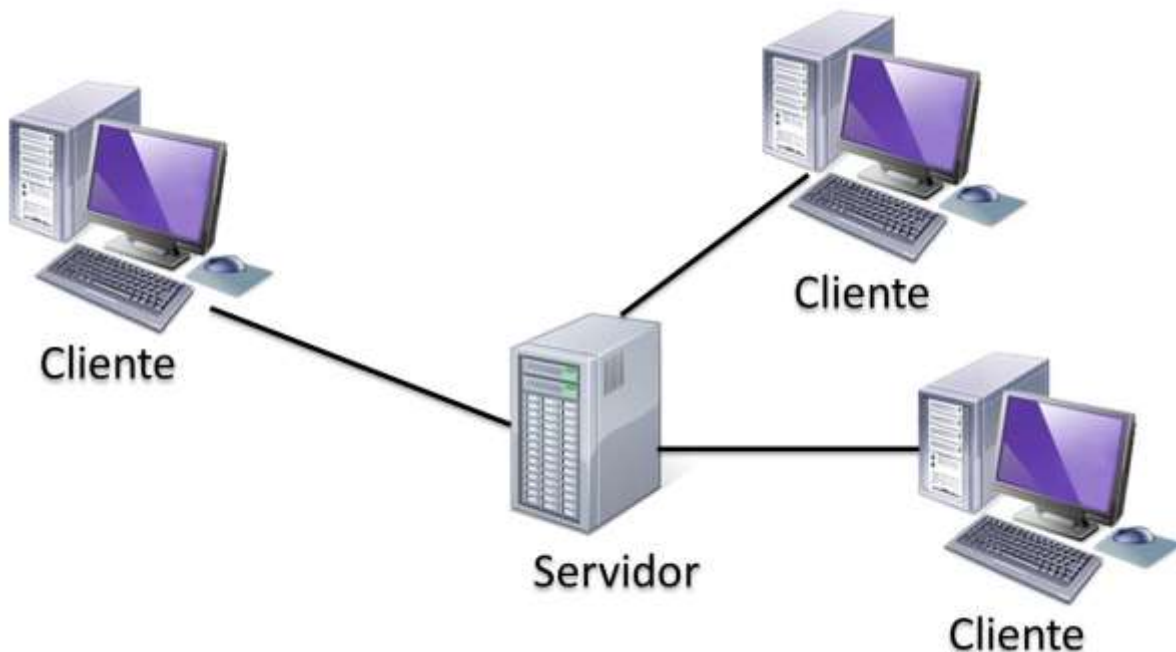
7. En el cuadro **Esquema predeterminado**, escriba **dbo** para asignar al usuario el esquema del propietario de la base de datos.



8. Acepte los valores predeterminados de los cuadros **Elementos protegibles** y **Estado** y haga clic en **Aceptar** para crear el inicio de sesión.



CONEXIÓN REMOTA



Contexto

Cuando se instala SQL Server, por defecto, para mejorar la seguridad, no se puede obtener acceso al motor de base de datos desde otros equipos de la red. En esta sección aprenderás a habilitar los protocolos, configurar los puertos y configurar el Firewall de Windows para conectarse desde otros equipos.

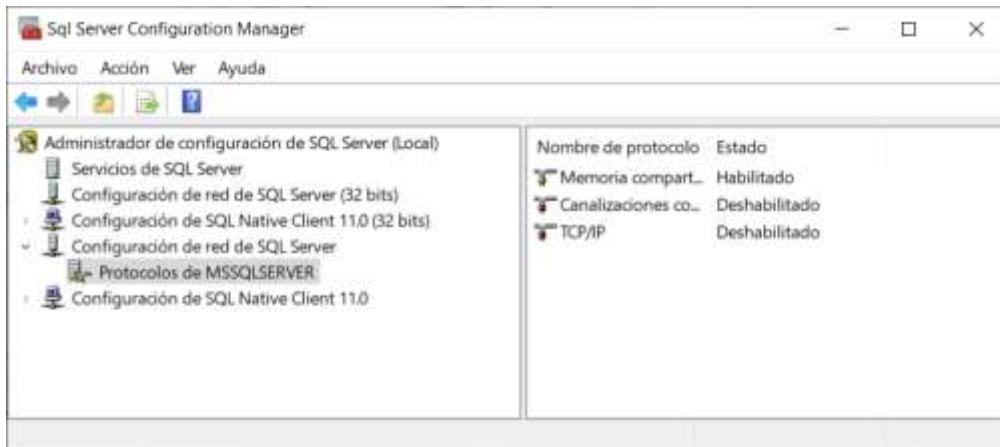
Habilitar protocolos

Para mejorar la seguridad, SQL Server se instalan con conectividad de red limitada. Las conexiones al motor de base de datos se pueden realizar desde herramientas que se ejecuten en el mismo equipo, no desde otros equipos. Si tiene previsto realizar las tareas de desarrollo en el mismo equipo que motor de base de datos, no necesita habilitar otros protocolos. Management Studio se conectará a motor de base de datos mediante el protocolo de memoria compartida. Este protocolo ya está habilitado.

Si tiene previsto conectarse al motor de base de datos desde otro equipo, debe habilitar un protocolo, como TCP/IP.

Cómo habilitar conexiones TCP/IP desde otro equipo:

1. En el menú **Inicio**, elija **Todos los programas**, **Microsoft SQL Server**, por último, **Administrador de configuración de SQL Server**.
2. En **Administrador de configuración de SQL Server**, expanda **Configuración de red de SQL Server** y, a continuación, haga clic en **Protocolos de <NombreInstancia>**.

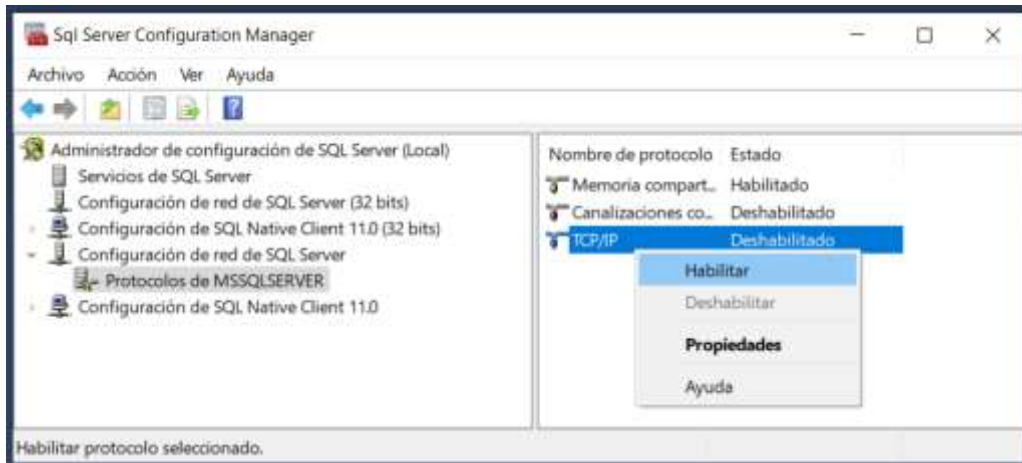


La instancia predeterminada (una instancia sin nombre) aparece como **MSSQLSERVER**. Si ha instalado una instancia con nombre, el nombre proporcionado aparece en la lista. Las versiones EXPRESS se instala como **SQLEXPRESS**, a menos que se haya cambiado el nombre durante la instalación.

3. En la lista de protocolos, haga clic con el botón secundario en el protocolo que desee habilitar (**TCP/IP**) y, a continuación, haga clic en **Habilitar**.

Nota

Debe reiniciar el servicio **SQL Server** después de realizar los cambios en los protocolos de red; sin embargo, esto se completa en la siguiente tarea.

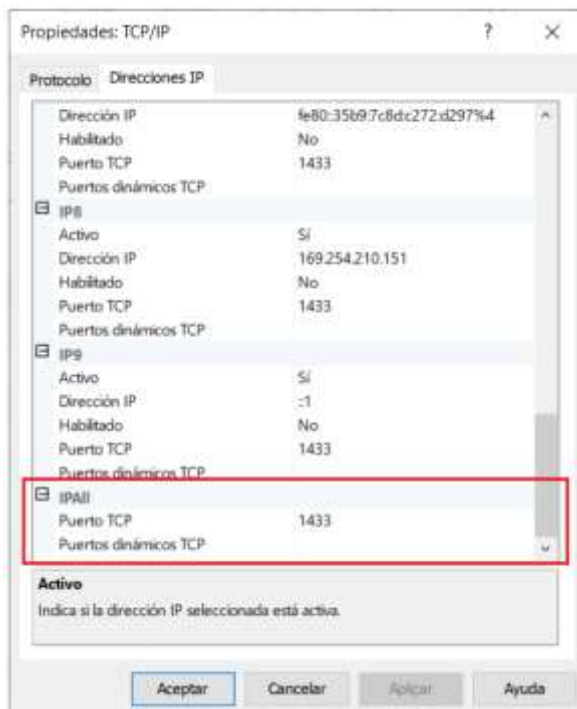


Configurar un puerto fijo

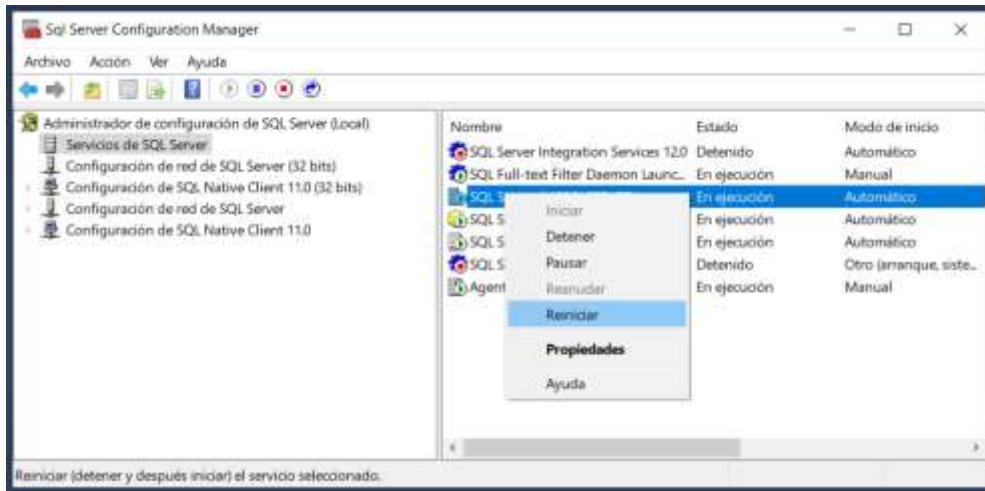
Para mejorar la seguridad, está activo el Firewall de Windows. Si desea conectarse a esta instancia desde otro equipo, debe abrir un puerto de comunicaciones en el firewall. La instancia predeterminada del motor de base de datos escucha en el puerto **1433**; por tanto, no tiene que configurar un puerto fijo. No obstante, las instancias con nombre incluidas las de SQL Server Express escuchan en puertos dinámicos. Para poder abrir un puerto en el firewall, debe configurar primero que el motor de base de datos escuche en un puerto específico conocido como puerto fijo o estático; de lo contrario, es posible que el Motor de base de datos escuche en un puerto distinto cada vez que se inicie.

Para configurar SQL Server para escuchar en un puerto específico:

1. En el **Administrador de configuración de SQL Server**, expanda **Configuración de red de SQL Server** y, a continuación, haga clic en la instancia de servidor que desee configurar, por ejemplo, **Protocolos de MSSQLSERVER**.
2. En el panel derecho, haga doble clic en **TCP/IP**.
3. En el cuadro de diálogo **Propiedades de TCP/IP**, haga clic en la pestaña **Direcciones IP**.



4. En el cuadro **Puerto TCP** de la sección **IPv4**, escriba un número de puerto disponible. Para este tutorial, se utilizará 49172.
5. Haga clic en **Aceptar** para cerrar el cuadro de diálogo y en **Aceptar** cuando aparezca una advertencia que indique que debe reiniciarse el servicio.
6. En el panel izquierdo, haga clic en **Servicios de SQL Server**.
7. En el panel derecho, haga clic con el botón secundario en la instancia de SQL Server y, a continuación, haga clic en **Reiniciar**. Cuando se reinicia el motor de base de datos, escucha en el puerto 49172.



Abrir puertos del firewall

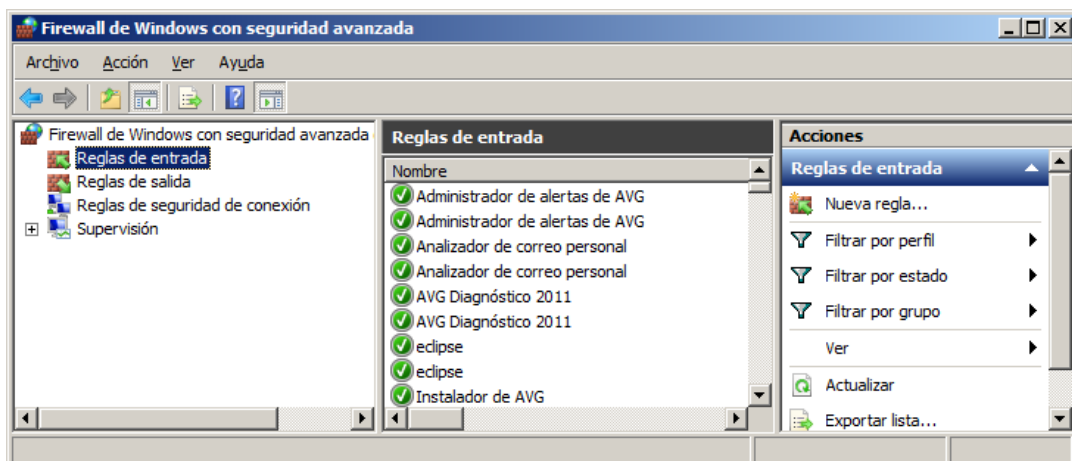
Los sistemas FIREWALL ayudan a evitar el acceso no autorizado a los recursos de los equipos.

Para conectarse a SQL Server desde otro equipo cuando el firewall está activo, debe abrir un puerto en el firewall.

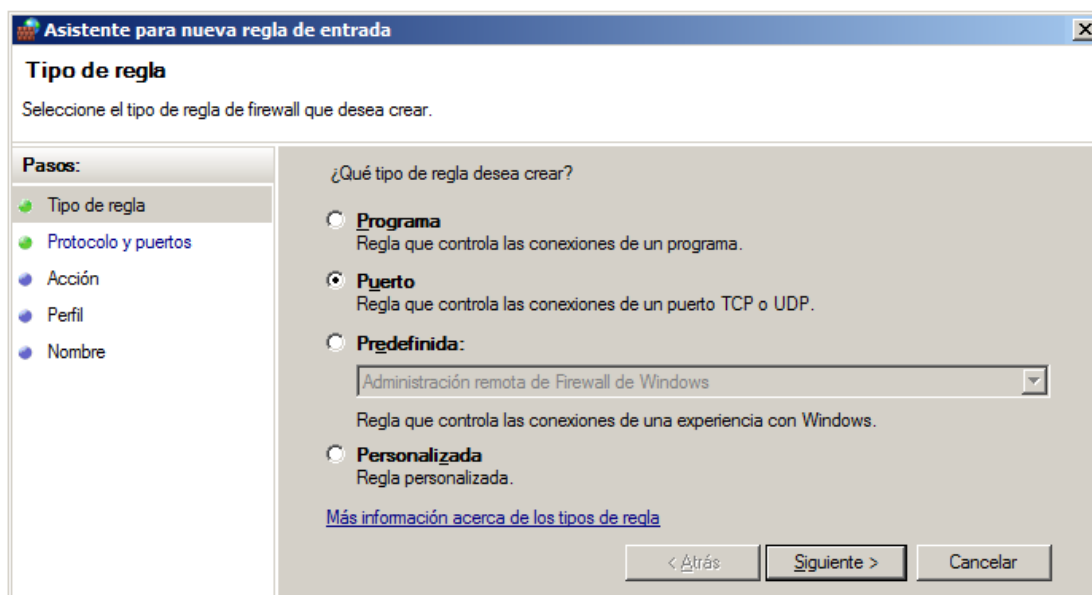
Después de configurar el motor de base de datos para utilizar un puerto fijo, siga estas instrucciones para abrir ese puerto en el Firewall de Windows. (No es necesario configurar un puerto fijo para la instancia predeterminada, porque ya está fijada en el puerto TCP 1433).

Para abrir un puerto en el Firewall de Windows para el acceso TCP:

1. En el menú **Inicio**, haga clic en **Ejecutar**, escriba **WF.msc** y, a continuación, haga clic en **Aceptar**.



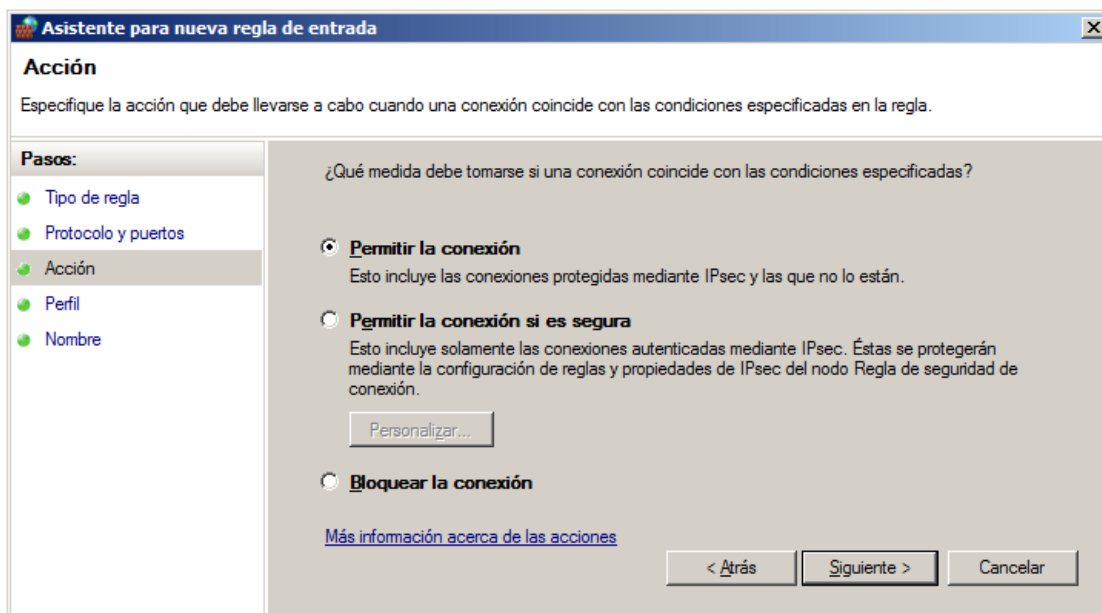
2. En la opción **Firewall de Windows con seguridad avanzada** del panel izquierdo, haga clic en **Reglas de entrada** y, a continuación, haga clic en **Nueva regla** en el panel **Acciones**.
3. En el cuadro de diálogo **Tipo de regla**, seleccione **Puerto** y, a continuación, haga clic en **Siguiente**.



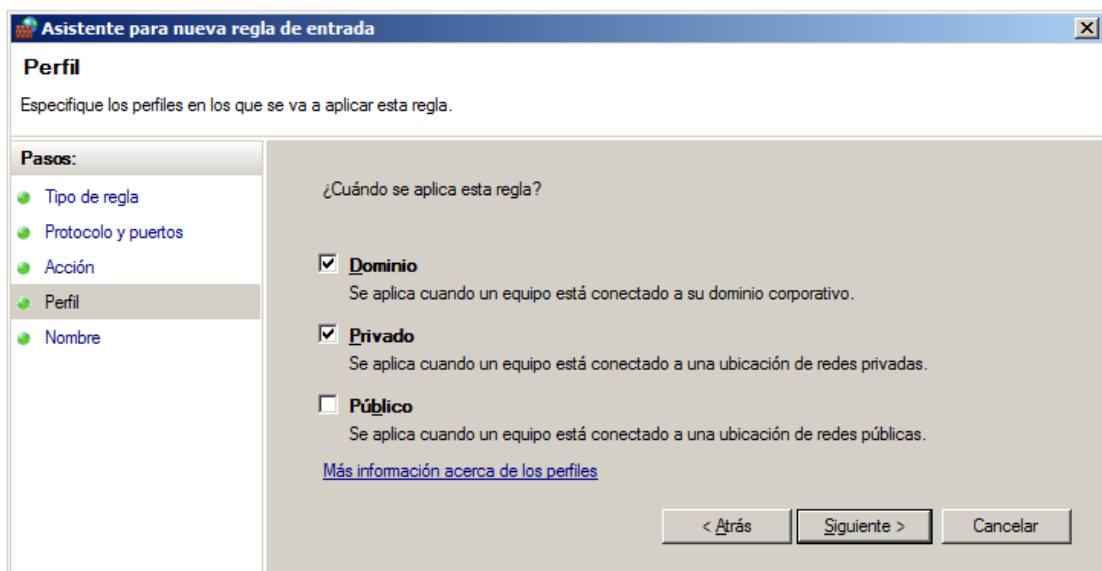
4. En el cuadro de diálogo **Protocolo y puertos**, seleccione **TCP**. Seleccione **Puertos locales específicos** y, a continuación, escriba el número de puerto de la instancia del motor de base de datos. Use **1433** para la instancia predeterminada. Escriba **49172** si está configurando una instancia con

nombre y ha configurado el puerto fijo de la tarea anterior. Haga clic en **Siguiente**.

5. En el cuadro de diálogo **Acción**, seleccione **Permitir la conexión** y, a continuación, haga clic en **Siguiente**.



6. En el cuadro de diálogo **Perfil**, seleccione los perfiles que describan el entorno de conexión del equipo cuando desee conectarse al motor de base de datos y, a continuación, haga clic en **Siguiente**.



7. En el cuadro de diálogo **Nombre**, escriba un nombre y una descripción para esta regla. Después, haga clic en **Finalizar**.

The screenshot shows a Windows-style dialog box titled "Asistente para nueva regla de entrada". The "Nombre" step is selected in the "Pasos:" list on the left. The main area contains a "Nombre:" label with a text box containing "SQL Server Laboratorio", and a "Descripción (opcional):" label with a text box containing "Configuración de puerto para prueba de laboratorio.". At the bottom right are three buttons: "< Atrás", "Finalizar" (highlighted with a dashed border), and "Cancelar".

Conexión desde otro equipo

Ahora que ha configurado el motor de base de datos para escuchar en un puerto fijo y ha abierto este puerto en el firewall, puede conectarse a SQL Server desde otro equipo.

Cuando el servicio **SQL Server Browser** está ejecutándose en el equipo servidor y el firewall haya abierto el puerto UDP 1434, la conexión se podrá realizar utilizando el nombre del equipo y el nombre de la instancia. Para mejorar la seguridad, el ejemplo no utiliza el servicio SQL Server Browser.

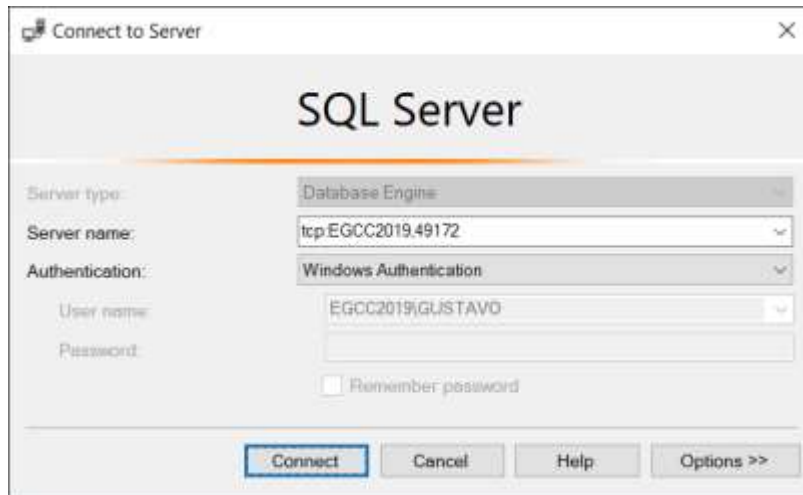
Para conectarse al motor de base de datos desde otro equipo:

1. En un segundo equipo que incluya las herramientas de cliente de SQL Server, inicie una sesión con una cuenta autorizada para conectarse a SQL Server y abra **Management Studio**.
2. En el cuadro de diálogo **Conectar con el servidor**, confirme **Motor de base de datos** en el cuadro **Tipo de servidor**.
3. En el cuadro **Nombre del servidor**, escriba **tcp:** para especificar el protocolo, seguido del nombre del equipo, una coma y el número de puerto. Para conectarse a la instancia predeterminada, el puerto 1433 está implícito y se puede omitir, por lo que deberá escribir **tcp:<nombre_computador>**. En nuestro ejemplo, escriba **tcp:<computer_name>,49172**.

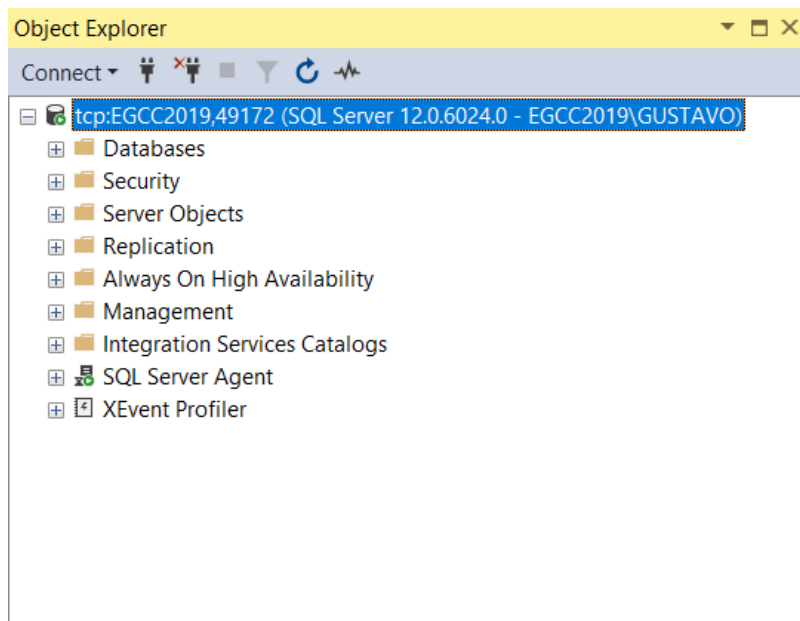
Nota

Si omite **tcp:** en el cuadro **Nombre del servidor**, el cliente probará todos los protocolos habilitados en el orden especificado en la configuración del cliente.

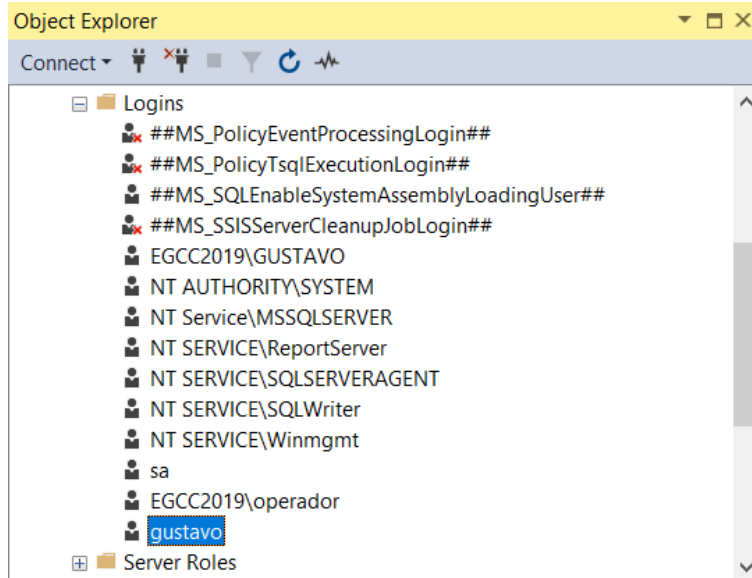
4. En el cuadro **Autenticación**, confirme **Autenticación de Windows** y, a continuación, haga clic en **Conectar**.



5. El resultado lo puedes ver en la siguiente imagen.



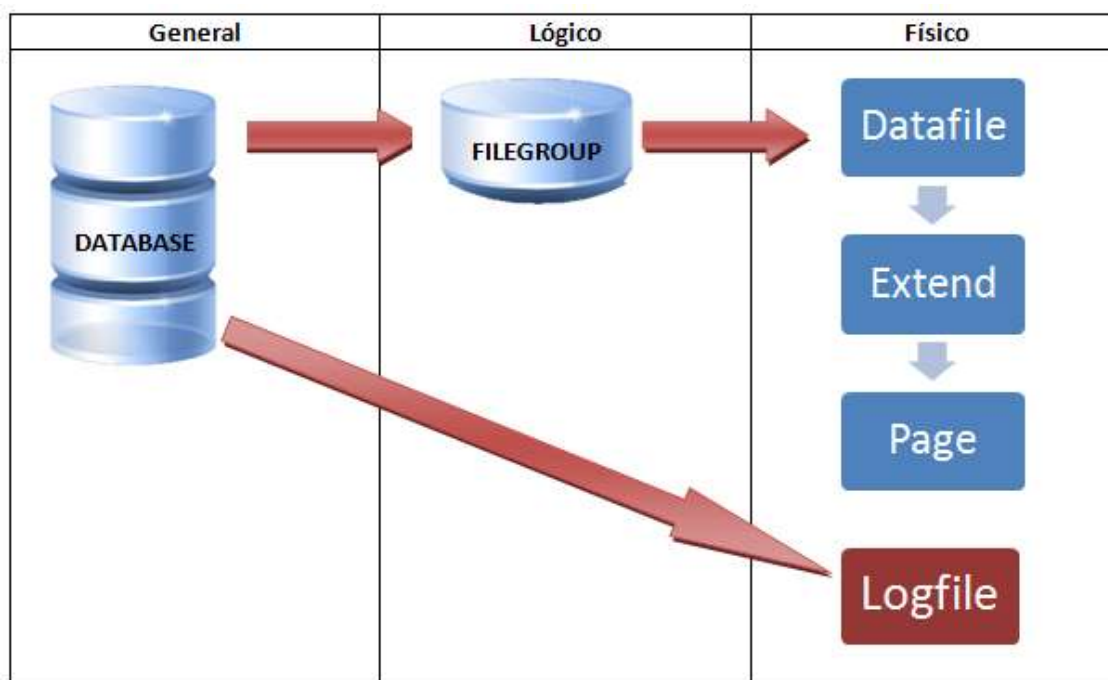
6. Acepte los valores predeterminados de los cuadros **Elementos protegibles** y **Estado** y haga clic en **Aceptar** para crear el inicio de sesión.



Capítulo 3

TRABAJANDO CON BASES DE DATOS

ARQUITECTURA DE BASES DE DATOS SQL SERVER



La arquitectura interna de las bases de datos en SQL Server está compuesta por 2 tipos de estructura, la estructura lógica y la estructura física. Es muy importante conocer cómo es que estas estructuras están compuestas y cuál es la relación que tienen los objetos de base de datos con cada una de estas estructuras.

Estructura Lógica

Desde el punto de vista lógico, la base de datos debe tener al menos 1 "FileGroup" el cual contiene a toda la metadata de la misma base de datos, es decir tablas y vistas de sistema, a este "FileGroup" inicial se le conoce como "Primario" y está presente en todas las bases de datos.

Todos los objetos de usuario que contengan data, ya sean tablas o índices, deben estar ligados a un "FileGroup", esto se puede definir al momento de ejecutar la sentencia DDL de creación del objeto, si no se indica a que "FileGroup" estará ligado

ese objeto, este pertenecerá al “FileGroup” por defecto definido en la base de datos. La base de datos solo puede tener definido 1 solo “FileGroup” por defecto.

Las bases de datos pueden tener hasta 32767 “FileGroups” definidos, según los límites establecidos. Uno de los propósitos de los “FileGroups” es poder distribuir la data a través de varios discos duros físicos, de esta manera se puede obtener mayor rendimiento en las operaciones de I/O debido a que más de un disco trabajara al mismo tiempo. Otro de los propósitos es poder esconder la ubicación física real de la información a los programadores, ya que para ellos la tabla “X” pertenece al “FileGroup” “A”, pero no saben en que data files físicamente se encuentra la información de la tabla “X”.

Nota

Para saber las capacidades máximas de SQL Sever, puedes consultar el siguiente enlace:

<https://docs.microsoft.com/en-us/sql/sql-server/maximum-capacity-specifications-for-sql-server>

Los “FileGroups” pueden contener 1 o más “Datafiles”, y cada uno de estos datafiles se puede encontrar en discos diferentes, lo cual también agilizará las consultas y los ingresos de información a las tablas que se encuentren asignadas a este “FileGroup”, debido a que SQL Server distribuirá la información uniformemente a través de todos los “DataFiles” del “FileGroup”.

Estructura Física

Desde el punto de vista físico, como ya hemos visto, tenemos los “DataFiles” que son en realidad los archivos de datos, es decir donde se guarda toda la información de la base de datos. Un “DataFile” solo puede pertenecer a 1 “FileGroup”.

Internamente los “DataFiles” están divididos en “Extends” y estos a su vez en “Pages”. Las “Pages” son la unidad mínima de almacenamiento dentro de la base de datos. Un “Page” tiene 8 Kb de tamaño en espacio de disco. Un “Extend” tiene 8 “Pages” contiguas que lo conforman, es decir, un “Extend” tiene como tamaño 64 Kb de espacio en disco.

En un “Page” solo puede haber información de 1 sola tabla, es decir el espacio de un “Page” no es compartido entre tablas o índices. En el caso de los “Extends”, estos pueden ser de dos tipos:

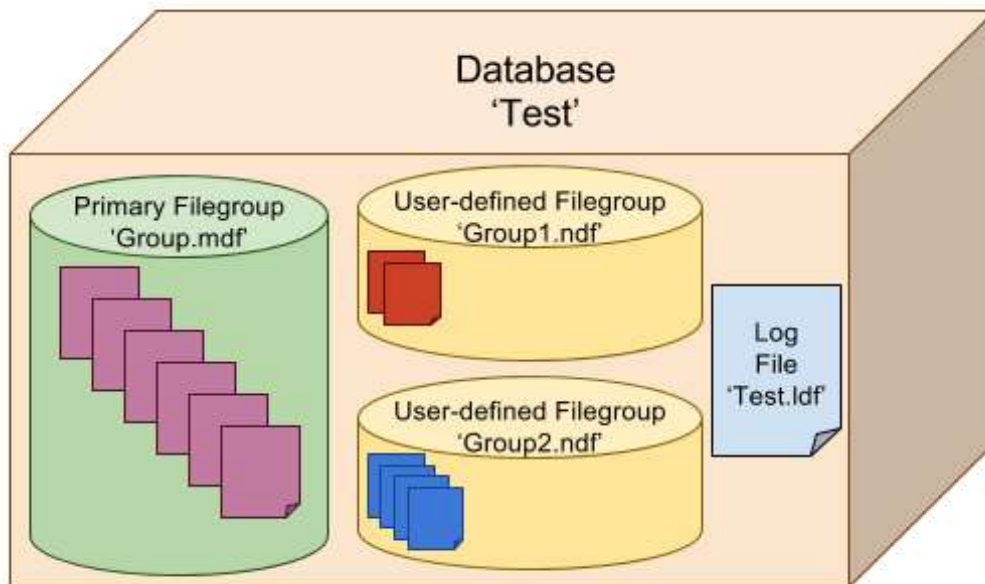
- **Mixed:** Los cuales son compartidos hasta por 8 objetos, uno por cada “Page”.
- **Uniform:** Los cuales solo pertenecen a un solo objeto, es decir que todos los “Pages” pertenecen a un solo objeto.

Normalmente cuando se crea una nueva tabla esta es asignada a un “Extend” de tipo “Mixed”, hasta alcanzar la utilización de hasta 8 “Pages”, a partir de ese momento se asignan “Extends” de tipo “Uniform” para optimizar el uso del espacio en la tabla.

Los “DataFiles” normalmente tienen 2 extensiones de archivo, las cuales son estándar mas no obligatorias, la extensión “**mdf**” que se utiliza para el primer “Datafile” perteneciente al “FileGroup” primario, y la extensión “**ndf**” que se utiliza para los demás datafiles que se agregan posteriormente a los demás “FileGroups” de la base de datos.

En el caso del “LogFile”, este no pertenece a un “FileGroup” en específico, en cambio archivo está ligado directamente a la base de datos. Las bases de datos de SQL Server solo pueden tener un solo “LogFile” activo al mismo tiempo, si bien se pueden crear múltiples “LogFiles” en la base de datos, solo uno podrá ser escrito, ya que solo uno puede estar activo, cuando este archivo se llene, la base de datos pasará a escribir al siguiente archivo de transacciones, y así sucesivamente. Por esta razón no es muy conveniente ni útil tener más de un “LogFile”.

CREACIÓN DE UNA BASE DE DATOS



Contexto

Ahora conocerás los fundamentos de la creación de una base de datos en SQL SERVER, así mismo realizaras una práctica sobre cómo crear una base de datos con la herramienta gráfica Management Studio y Transact-SQL.

Lógicamente es necesario que tengamos instalado el servidor de base de datos o dicho de otra manera: Microsoft SQL Server en cualquier versión, usualmente al instalar SQL SERVER ya viene incluida la herramienta Management Studio que te permite conectarte y manejar tu servidor SQL Server desde una interfaz gráfica en lugar de tener que usar la línea de comandos.

Concepto de base de datos

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso, en otros términos, una base de datos es un “ALMACEN” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrarla y utilizarla fácilmente.

Crear base de datos usando T-SQL

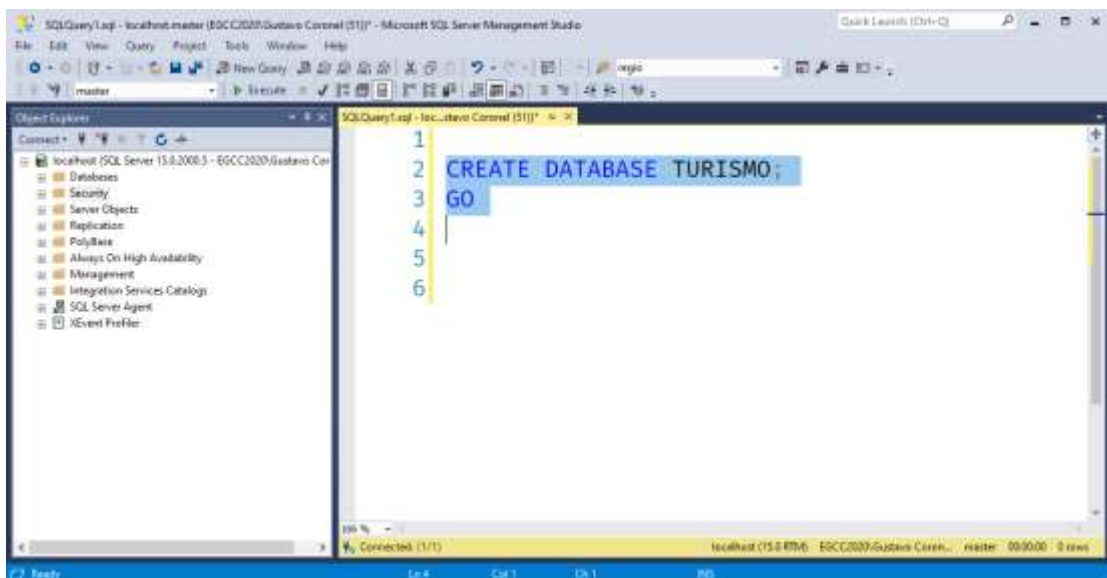
T-SQL es el lenguaje de manipulación de los datos en SQL Server, cada SGBD tiene un lenguaje propio, así, pues si utilizamos ORACLE el lenguaje de manipulación será el PL-SQL. Sin embargo, hay un lenguaje estándar llamado SQL y ese, aunque con limitantes en el sentido que no provee muchas funciones que los propios de las SGBD tienen, funciona en cualquier SGBD.

T-SQL o PL-SQL es el nombre dado a la modificación del lenguaje SQL por Microsoft y Oracle respectivamente, la esencia del lenguaje se mantiene, es por ello que existen muchas similitudes entre ambos. Ahora nos centraremos en el T-SQL.

Aprenderemos la forma de hacerlo mediante las Querys o consultas:

- Lo Primero que hacemos es dar click al botón **New Query** (o Nueva Consulta) y abrimos una nueva consulta.
- Debemos crear una base de datos y luego ponerla en uso entonces escribimos lo siguiente y lo ejecutamos:

```
CREATE DATABASE TURISMO;
```



Para ejecutar una sentencia, la seleccionamos con el mouse (toda la sentencia) y presionamos el botón **Ejecutar**.

```
-- Crea la base de datos

CREATE DATABASE TURISMO;
GO

-- Activar la BD TURISMO para poder hacer modificaciones

USE TURISMO;
GO
```

A continuación, tienes el script de creación de la base de datos TURISMO, pero indicando de manera explícita otros parámetros, como la ubicación de los archivos que la conforman:

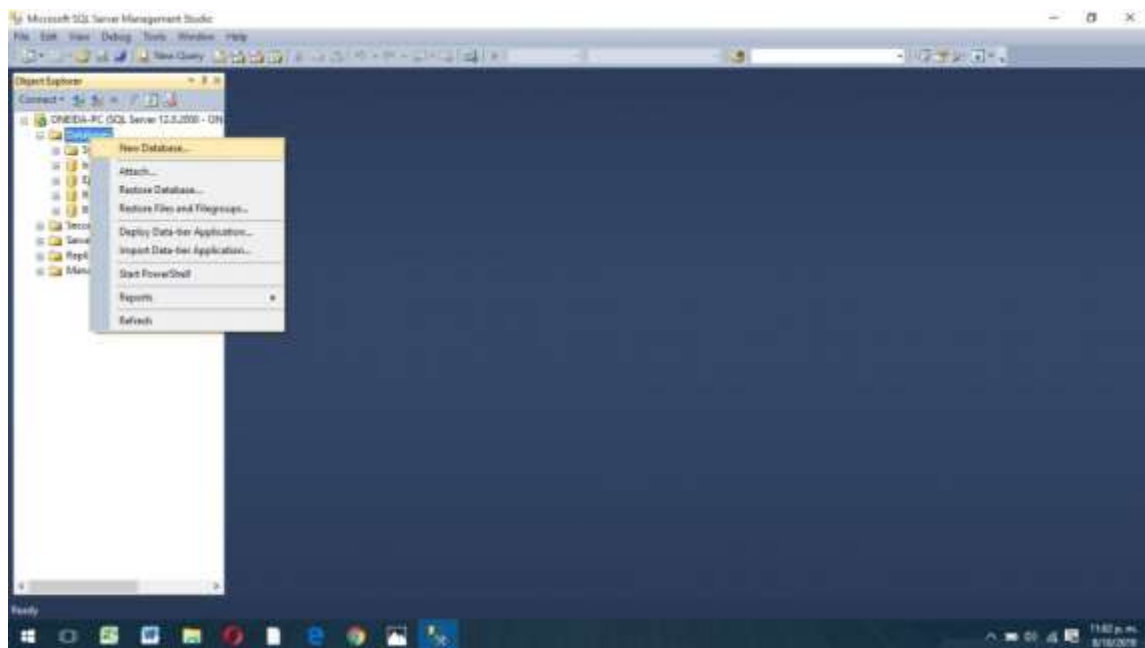
```
USE master;
GO

CREATE DATABASE TURISMO
ON
( NAME = TURIMO,
  FILENAME = 'E:\SQLServerDB\TURISMO.mdf',
  SIZE = 10,
  MAXSIZE = 50,
  FILEGROWTH = 5 )
LOG ON
( NAME = TURISMOLOG,
  FILENAME = 'E:\SQLServerDB\TURISMOLOG.ldf',
  SIZE = 5MB,
  MAXSIZE = 25MB,
  FILEGROWTH = 5MB );
GO
```

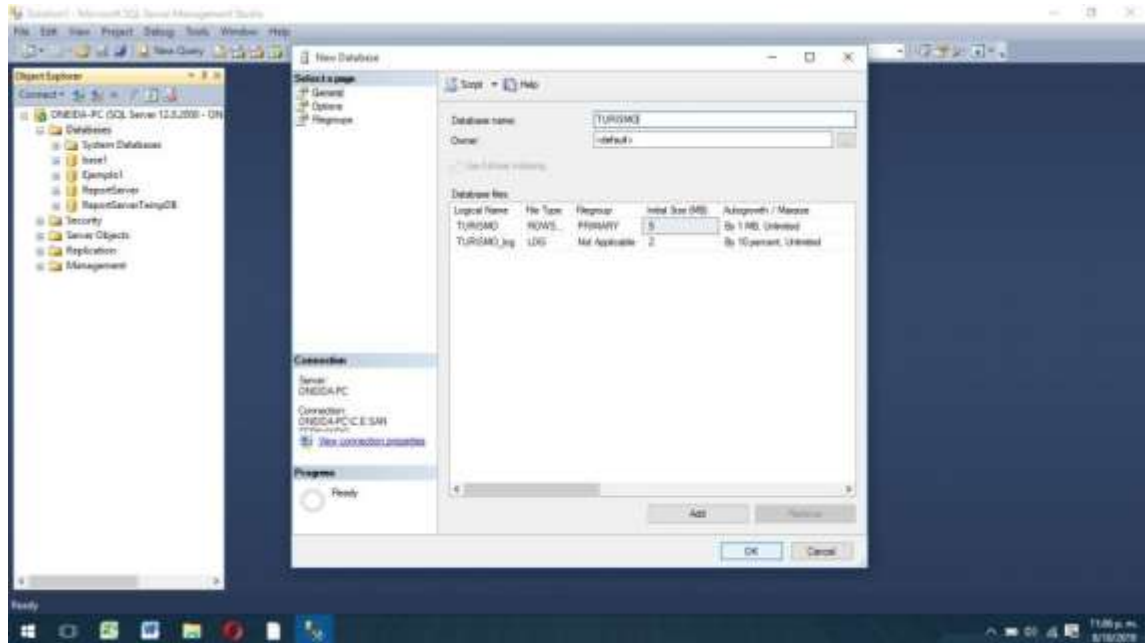
Creación una base de datos usando SSMS

Paso a seguir:

1. En el **Explorador de objetos**, conéctese a una instancia del motor de base de datos de SQL Server y expándala.
2. Hacer click con el botón derecho del Mouse en **Bases de datos** y luego hacer click en **Nueva base de datos** o **New database**.



3. En el diálogo **Nueva base de datos**, especifique un nombre para la base de datos.



4. Si desea crear la base de datos aceptando todos los valores predeterminados, hacer click en el botón **Aceptar**; de lo contrario, continúe con siguientes los pasos opcionales.

Básicamente esos son los pasos para crear una base de datos en SQL Server de forma gráfica.

Es importante mencionar que ya teniendo creada la base de datos vacía, lo que corresponde hacer es crear las tablas en las cuales se almacena y guarda la información de forma organizada.

GESTIÓN DE ARCHIVOS

Grupos de archivos

El grupo de archivos contiene el archivo de datos principal y cualquier otro archivo secundario que no se encuentre en otro grupo de archivos.

Se pueden crear grupos de archivos definidos por el usuario para agrupar archivos con fines administrativos y de asignación y ubicación de datos.

Por ejemplo, **Data1.ndf**, **Data2.ndf** y **Data3.ndf** pueden crearse en tres unidades de disco, respectivamente, y asignarse al grupo de archivos **fgroup1**. Se puede crear una tabla específicamente para el grupo de archivos **fgroup1**. Las consultas de datos de la tabla se distribuirán por los tres discos, con lo que mejorará el rendimiento. Puede obtenerse la misma mejora del rendimiento con un solo archivo creado en un conjunto de bandas RAID (matriz redundante de discos independientes). No obstante, los archivos y grupos de archivos permiten agregar fácilmente nuevos archivos a discos nuevos.

Todos los archivos de datos se almacenan en los grupos de archivos que se indican en la tabla siguiente.

GRUPO DE ARCHIVOS	DESCRIPCIÓN
Principal	Grupo de archivos que contiene el archivo principal. Todas las tablas del sistema forman parte del grupo de archivos principal.
Tabla optimizada para memoria	Un grupo de archivos optimizados para memoria está basado en un grupo de archivos de FILESTREAM.
Definidos por el usuario	Cualquier grupo de archivos creado por el usuario al crear la base de datos o al modificarla.

Grupo de archivos predeterminado

Cuando se crean objetos en la base de datos sin especificar a qué grupo de archivos pertenecen, se asignan al grupo de archivos predeterminado. Siempre existe un grupo de archivos designado como predeterminado. Los archivos del grupo de archivos predeterminado deben ser lo suficientemente grandes como para dar cabida a todos los objetos nuevos no asignados a otros grupos de archivos.

El grupo de archivos PRINCIPAL es el predeterminado, a menos que se cambie mediante la instrucción ALTER DATABASE. Los objetos y las tablas del sistema no se asignan al nuevo grupo de archivos predeterminado, sino que siguen asignados al grupo de archivos PRIMARY.

Ejemplo de archivos y grupos de archivos

En el siguiente ejemplo se crea una base de datos en una instancia de SQL Server. La base de datos tiene un archivo de datos principal, un grupo de archivos definido por el usuario y el archivo de registro.

El archivo de datos principal está en el grupo de archivos principal y el grupo de archivos definido por el usuario tiene dos archivos de datos secundarios. Una instrucción ALTER DATABASE hace que el grupo de archivos definido por el usuario sea el grupo predeterminado. A continuación, se crea una tabla que especifica el grupo de archivos definido por el usuario.

En este ejemplo se usa una ruta de acceso **E:\SQLServerDB** para los archivos.

Creación de la base de datos:

```
USE master;
GO

CREATE DATABASE TURISMO
ON PRIMARY
( NAME='TURISMO',
  FILENAME= 'E:\SQLServerDB\TURISMO.mdf',
  SIZE=4MB,
  MAXSIZE=10MB,
  FILEGROWTH=1MB),
FILEGROUP TURISMO_FG1
( NAME = 'TURISMO_FG1_Data1',
  FILENAME = 'E:\SQLServerDB\TURISMO_FG1_Data1.ndf',
  SIZE = 1MB,
  MAXSIZE=10MB,
  FILEGROWTH=1MB),
( NAME = 'TURISMO_FG1_Data2',
  FILENAME = 'E:\SQLServerDB\TURISMO_FG1_Data2.ndf',
  SIZE = 1MB,
```

```
MAXSIZE=10MB,  
FILEGROWTH=1MB)  
LOG ON  
( NAME='TURISMOLOG',  
  FILENAME = 'E:\SQLServerDB\TURISMOLOG.ldf',  
  SIZE=1MB,  
  MAXSIZE=10MB,  
  FILEGROWTH=1MB);  
GO
```

Cambiar el file group por defecto:

```
ALTER DATABASE TURISMO  
  MODIFY FILEGROUP TURISMO_FG1 DEFAULT;  
GO
```

Creando una tabla en el file group de usuario:

```
USE TURISMO;  
CREATE TABLE MyTable  
(  
  id int PRIMARY KEY,  
  dato varchar(100)  
)  
ON TURISMO_FG1;  
GO
```


PRACTICA

Ejercicio 1

Crear la base de datos **LibroTSQL** y, como su nombre indica, se creará con la ayuda de un script Transact SQL. Crearemos una segunda base de datos, llamada LibroSSMS, desde la interfaz gráfica del Management Studio.

Los archivos de datos se definirán en un directorio específico. Conviene crear el directorio **c:\datos**. Evidentemente, guardar los datos en el disco **c:** es producto del ejemplo pedagógico.

Parámetros de la base de datos LibroTSQL

Archivo de datos:

- Tamaño: 10 MB
- Nombre físico: c:\datos\LibroTSQL.mdf
- Nombre lógico: LibroTSQL

Archivo de log:

- Tamaño: 8 MB
- Nombre físico: c:\datos\LibroTSQL_log.ldf
- Nombre lógico: LibroTSQL_log

Parámetros de la base de datos LibroSSMS

Archivo de datos:

- Tamaño: 15 MB
- Nombre físico: c:\datos\LibroSSMS.mdf
- Nombre lógico: LibroSSMS

Archivo de log:

- Tamaño: 8 MB
- Nombre físico: c:\datos\LibroSSMS_log.ldf
- Nombre lógico: LibroSSMS_log

Ejercicio 2

En la base de datos LibroTSQL, añadir el grupo de archivos **Data**. Este grupo de archivos se compone de dos archivos: **data1.ndf** y **data2.ndf**.

El archivo **data1.ndf** posee un tamaño fijo de **50 MB** mientras que el archivo **data2.ndf** tiene un tamaño inicial de **10 MB** que puede después crecer hasta un tamaño de **50 MB** en pasos de **10 MB**.

