

# Capítulo 01



## Introducción

SQL Server 2014 es la última versión del software DBMS relacional lanzada por la empresa Microsoft, que está disponible desde el 1 de abril del 2014.

### Principales características

Mayor rendimiento, disponibilidad y capacidad de gestión para sus aplicaciones de misión crítica son algunos de los puntos en los que Microsoft ha hecho mayor énfasis a la hora de desarrollar la nueva versión de SQL Server.

En cuanto a novedades, Microsoft ha adelantado que SQL Server 2014 ofrecerá nuevas capacidades in-memory integradas en la base de datos central para OLTP y data warehousing, que complementan el almacenamiento de datos in-memory y capacidades de Business Intelligence.

En una época en la que las empresas tienen más información que analizar que nunca, SQL Server permitirá construir soluciones de Big Data y aplicaciones de misión crítica,

utilizando el alto rendimiento que ofrece la tecnología in-memory a través de OLTP, data warehousing, Business Intelligence, entornos de cloud híbrido y procesos analíticos, sin necesidad de invertir en costosas soluciones adicionales y aplicaciones de alta gama.

OLTP. Las nuevas capacidades in-memory para Proceso de Transacción Online in-memory que incluye SQL Server proporcionan un rendimiento 30 veces superior.

In-Memory Columnstore (Data Warehousing). Permite escribir y leer mayores volúmenes de datos y obtener repuestas a consultas de forma más rápida y con unos ratios de compresión de la información más elevados.

Nuevos escenarios híbridos para Windows Azure. Entre los que se incluyen backup para Windows Azure y la posibilidad de crear una Máquina Virtual de Windows Azure, que proporcionará un escenario de recuperación de desastres en el menor tiempo posible.

SQL Server 2014 ya está disponible en Windows Azure Virtual Machine, de modo que los clientes pueden beneficiarse desde la nube de la tecnología in-memory.

## **Versiones de SQL Server 2014**

### **Enterprise**

SQL Server Enterprise pone a su disposición funcionalidad de centro de datos completa y de alta gama para satisfacer los requisitos de base de datos y Business Intelligence más exigentes.

### **Standard**

SQL Server Standard proporciona funcionalidad de Business Intelligence y administración de datos básica para cargas de trabajo que no son críticas, con recursos de TI mínimos.

### **Business Intelligence**

SQL Server Business Intelligence permite a las organizaciones compilar e implementar soluciones BI corporativas autoservicio seguro, escalable y administrable.

### **Developer**

SQL Server 2014 Developer Edition permite a los desarrolladores compilar cualquier tipo de aplicación en SQL Server. Incluye toda la funcionalidad de la edición Enterprise, pero tiene licencias para usarse como sistema de prueba y desarrollo, no como un

servidor de producción. SQL Server Developer es una opción ideal para las personas que compilan y prueban aplicaciones.

## Web

SQL Server 2014 Web Edition es una opción con un costo total de propiedad bajo para los hosts de Web y los VAP de Web que proporciona capacidades asequibles de administración y escalabilidad para propiedades web, tanto de pequeña como de gran escala.

## Express

SQL Server Express es una edición gratuita de SQL Server, ideal para desarrollar e impulsar aplicaciones de escritorio y web, y pequeñas aplicaciones de servidor.

## Requisitos de hardware y software

Los siguientes requisitos se aplican a todas las instalaciones de SQL Server 2014:

### Componente

#### Requisito

##### .NET Framework

.NET 3.5 SP1 es un requisito para SQL Server 2014 cuando se selecciona Motor de base de datos, Reporting Services, Master Data Services, Data Quality Services, Replicación o SQL Server Management Studio, y no lo instala ya el programa de instalación de SQL Server.

Si ejecuta el programa de instalación y no tiene .NET 3.5 SP1, el programa de instalación de SQL Server requiere que descargue e instale .NET 3.5 SP1 para poder continuar con la instalación de SQL Server. (Instale .NET 3.5 SP1 desde Microsoft .NET Framework 3.5 Service Pack 1). El mensaje de error incluye un vínculo al centro de descarga o bien puede descargar .NET 3.5 SP1 desde Windows Update. Para evitar la interrupción durante la instalación de SQL Server, puede descargar e instalar .NET 3.5 SP1 antes de ejecutar el programa de instalación de SQL Server.

.NET 4.0 es un requisito para SQL Server 2014. SQL Server instala .NET 4.0 durante el paso de instalación de características.

#### Windows PowerShell

SQL Server 2014 no instala ni habilita Windows PowerShell 2.0; sin embargo, Windows PowerShell 2.0 es un requisito previo de instalación para los componentes del Motor de base de datos y SQL Server Management Studio. Si el programa de instalación notifica que Windows PowerShell 2.0 no está presente, puede instalarlo o habilitarlo siguiendo las instrucciones de la página Windows Management Framework.

#### Software de red

Los sistemas operativos admitidos para SQL Server 2014 tienen software de red integrado. Las instancias con nombre y predeterminadas de una instalación independiente admiten los siguientes protocolos de red: Memoria compartida, canalizaciones con nombre, TCP/IP y VIA.

#### Virtualización

SQL Server 2014 se admite en entornos de máquina virtual que se ejecutan en el rol Hyper-V de:

Windows Server 2008 SP2 Standard, Enterprise y Datacenter

Ediciones Windows Server 2008 R2 SP1 Standard, Enterprise y Datacenter.

Windows Server 2012 Datacenter y Standard.

#### Disco duro

SQL Server 2014 requiere un mínimo de 6 GB de espacio disponible en disco.

Las necesidades de espacio en disco variarán según los componentes de SQL Server 2014 que instale. Para obtener más información, vea Requisitos de espacio en disco duro (32 y 64 bits), más adelante en este tema. Para obtener información acerca de los tipos de almacenamiento admitidos para los archivos de datos, vea Tipos de almacenamiento para los archivos de datos

## Unidad

Para la instalación desde disco se necesita una unidad de DVD.

## Monitor

SQL Server 2014 requiere Super VGA (800x600) o un monitor de una resolución mayor.

## Internet

La funcionalidad de Internet necesita acceso a Internet (no necesariamente de carácter gratuito).

1La ejecución de SQL Server 2014 en una máquina virtual será más lenta que la ejecución nativa debido a la sobrecarga de virtualización.

## Requisitos de procesador, memoria y sistema operativo

Los siguientes requisitos de memoria y procesador se aplican a todas las ediciones de SQL Server 2014:

### Componente

#### Requisito

#### Memoria[1]

##### Mínimo:

Ediciones Express: 512 MB

Todas las demás ediciones: 1 GB

##### Se recomienda:

Ediciones Express: 1 GB

Todas las demás ediciones: Al menos 4 GB y debe aumentar a medida que el tamaño de la base de datos aumente para asegurar un rendimiento óptimo.

#### Velocidad del procesador

##### Mínimo:

Procesador x86: 1,0 GHz

Procesador x64: 1,4 GHz

Se recomienda: 2 GHz o superior

Tipo de procesador

Procesador x64: AMD Opteron, AMD Athlon 64, Intel Xeon compatible con Intel EM64T, Intel Pentium IV compatible con EM64T

Procesador x86: Procesador compatible con Pentium III o superior

La tabla siguiente se muestran los requisitos del sistema operativo para las ediciones principales de SQL Server 2014:

Edición de SQL Server

SQL Server Enterprise

Windows Server 2012 R2 Datacenter de 64 bits

Windows Server 2012 R2 Standard de 64 bits

Windows Server 2008 R2 SP1 Datacenter de 64 bits

Windows Server 2008 R2 SP1 Enterprise de 64 bits

Windows Server 2008 R2 SP1 Standard de 64 bits

Windows Server 2008 R2 SP1 Web de 64 bits

Windows Server 2008 SP2 Datacenter de 64 bits

Windows Server 2008 SP2 Enterprise de 64 bits

Windows Server 2008 SP2 Standard de 64 bits

Característica

Requisito de espacio en disco

Motor de base de datos y archivos de datos, replicación, búsqueda de texto completo y Data Quality Services. 811 MB

Analysis Services y archivos de datos. 345 MB

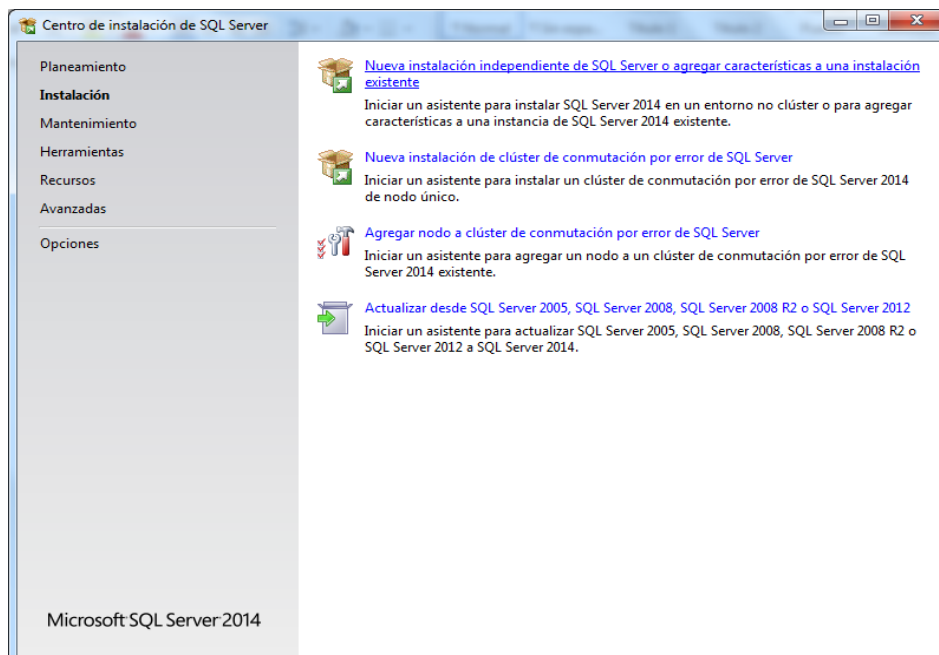
Reporting Services y Administrador de informes. 304 MB

## Instalación de SQL Server 2014

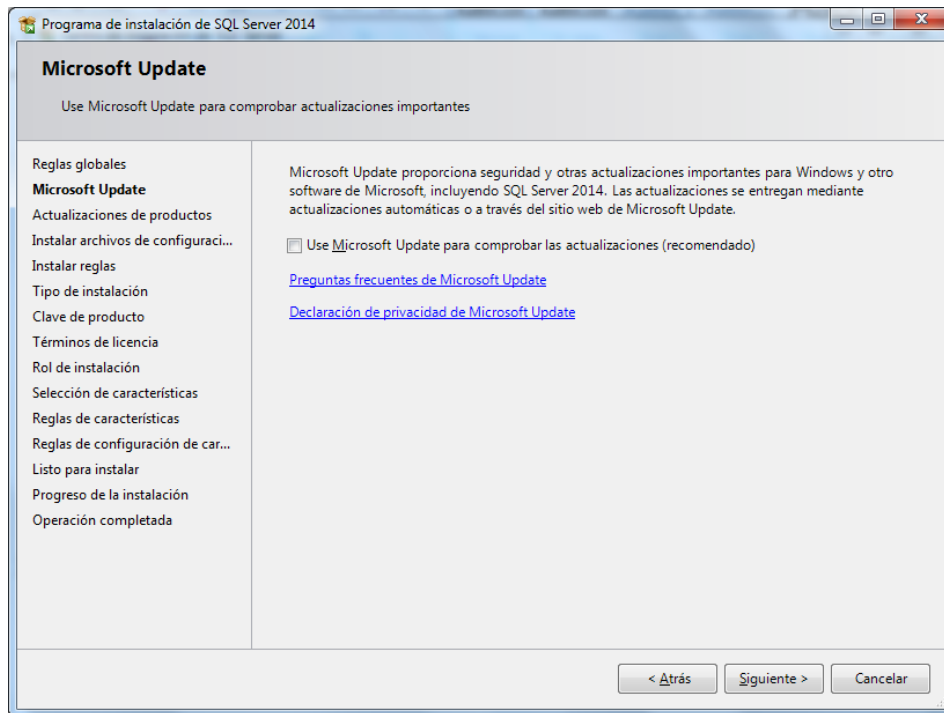
Para iniciar la instalación del producto de la unidad de dvd selecciones setup.exe , se presentara la siguiente ventana. Seleccione la opción Instalación



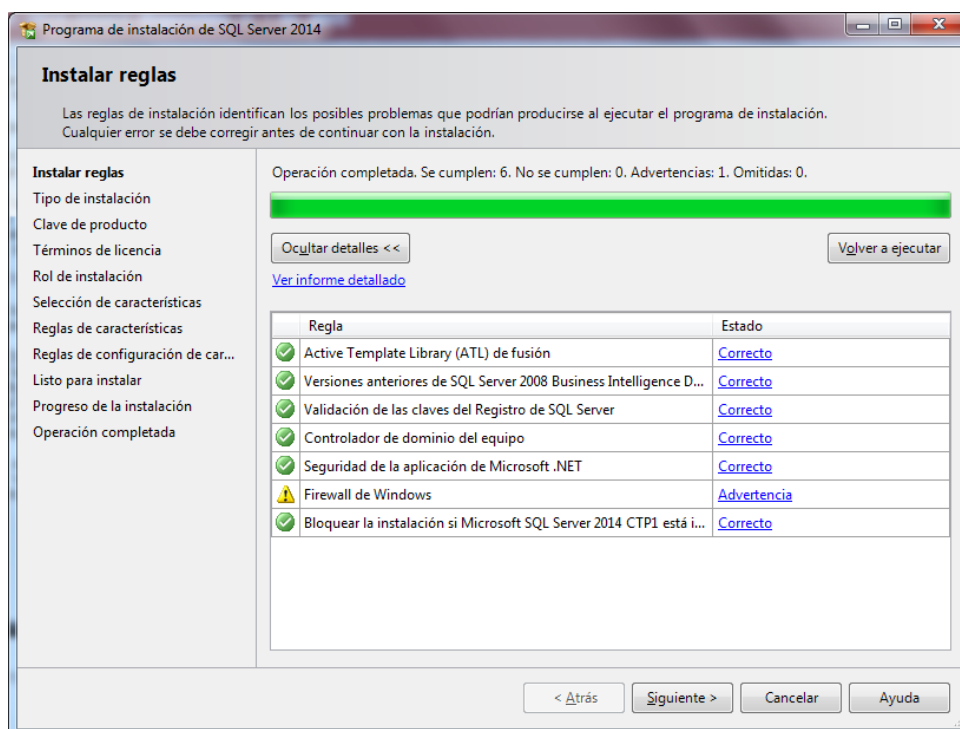
Ahora seleccione: Nueva instalación independiente de SQL server o agregar característica a una instalación existente.



En la ventana mostrada dar un clic en siguiente:

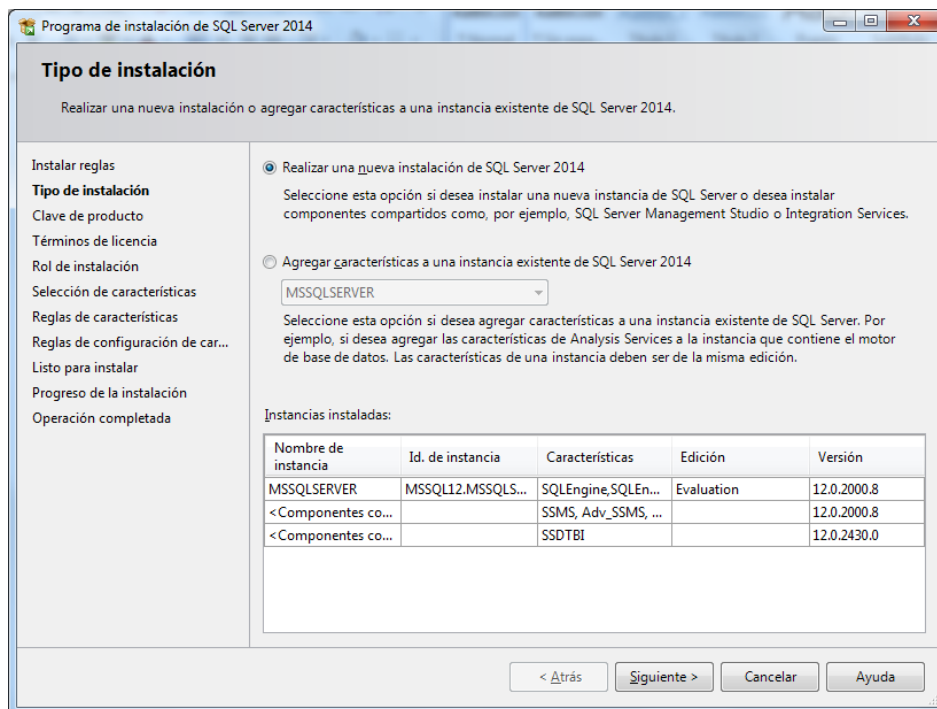


En la ventana instalar reglas, dar un clic en Siguiete

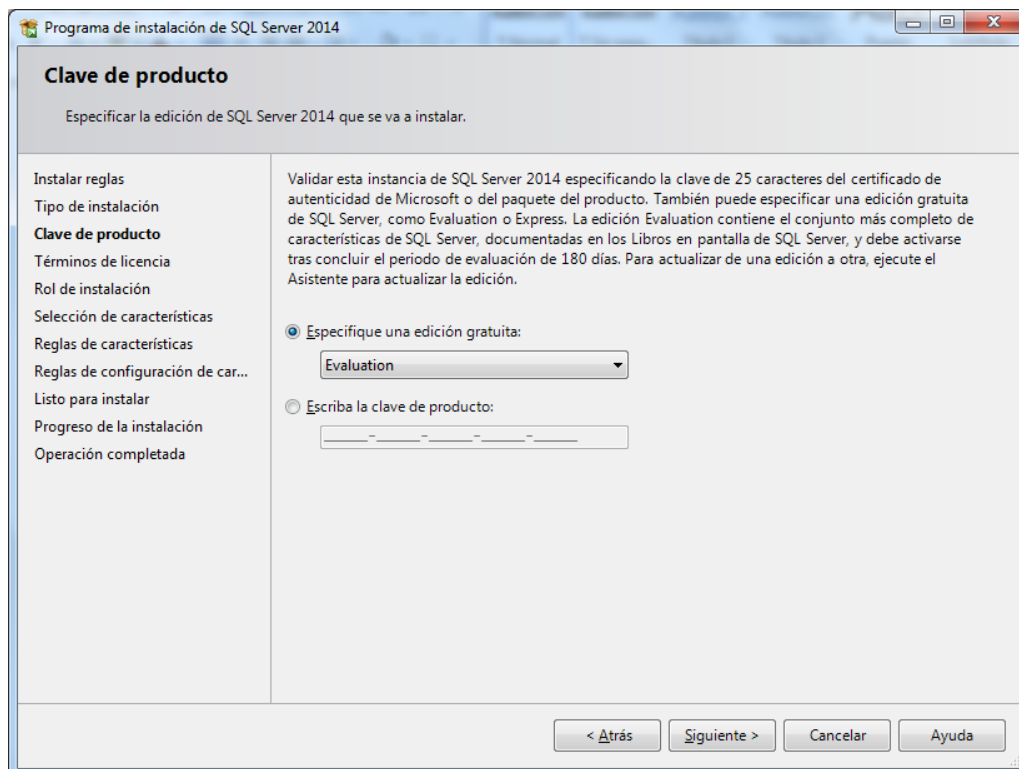




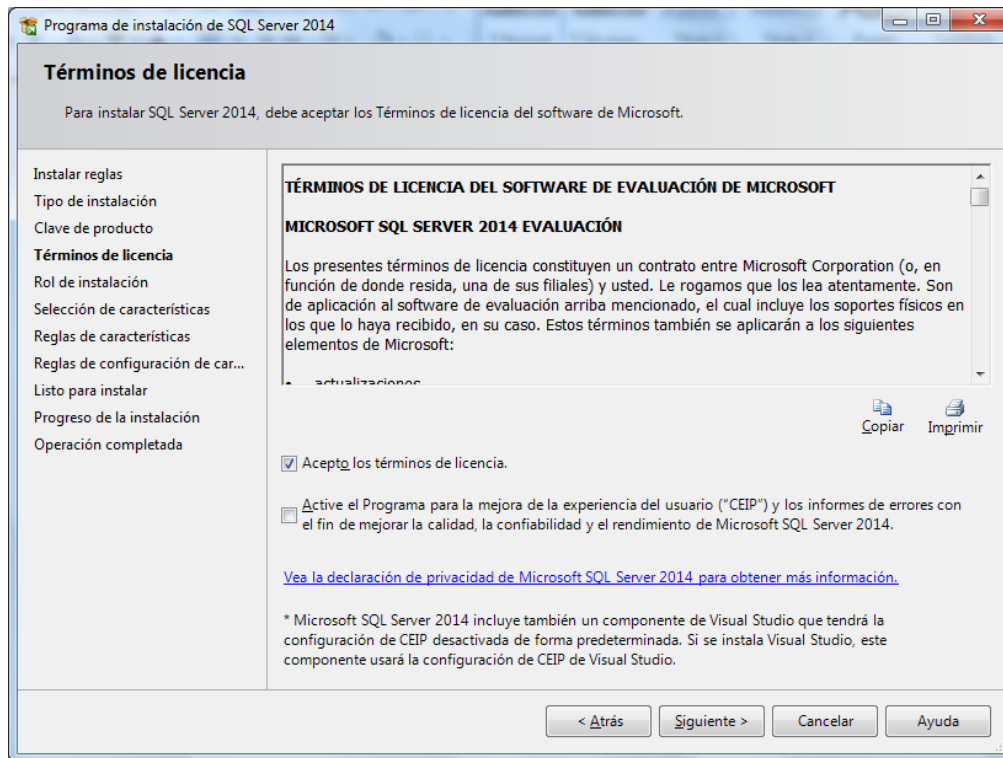
En la ventana Tipo de instalación, seleccione Realizar una nueva instalación de SQL Server 2014.



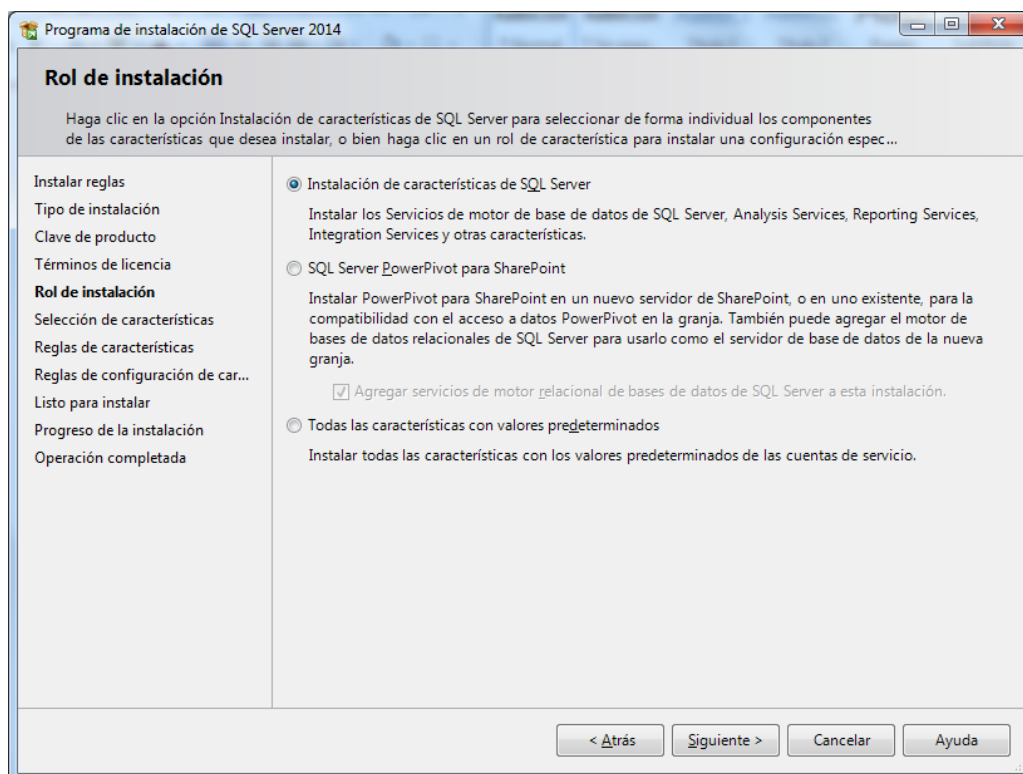
Ingrese la clave del producto o seleccione Evaluation



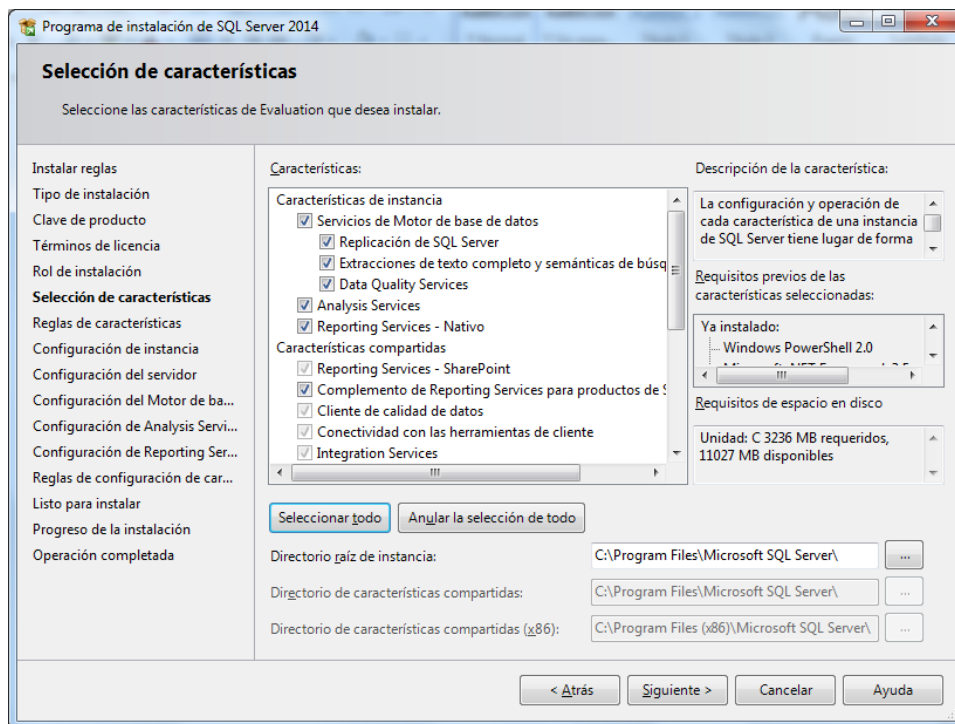
## Acepte los términos de licencia



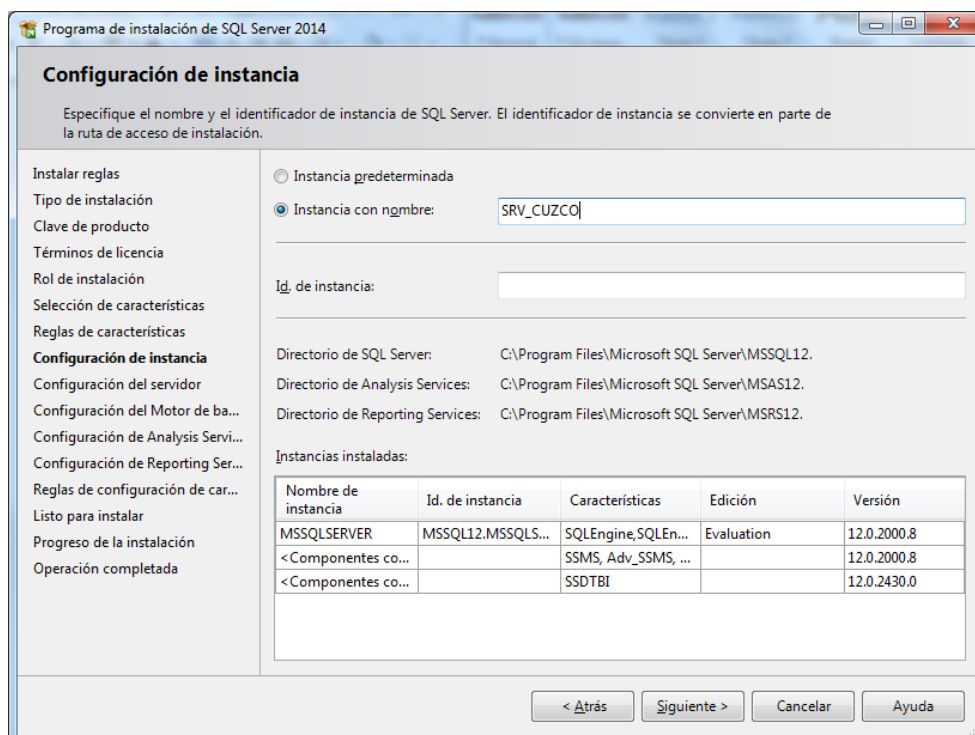
En la ventana Rol de instalación, seleccione Instalación de características de SQL Server



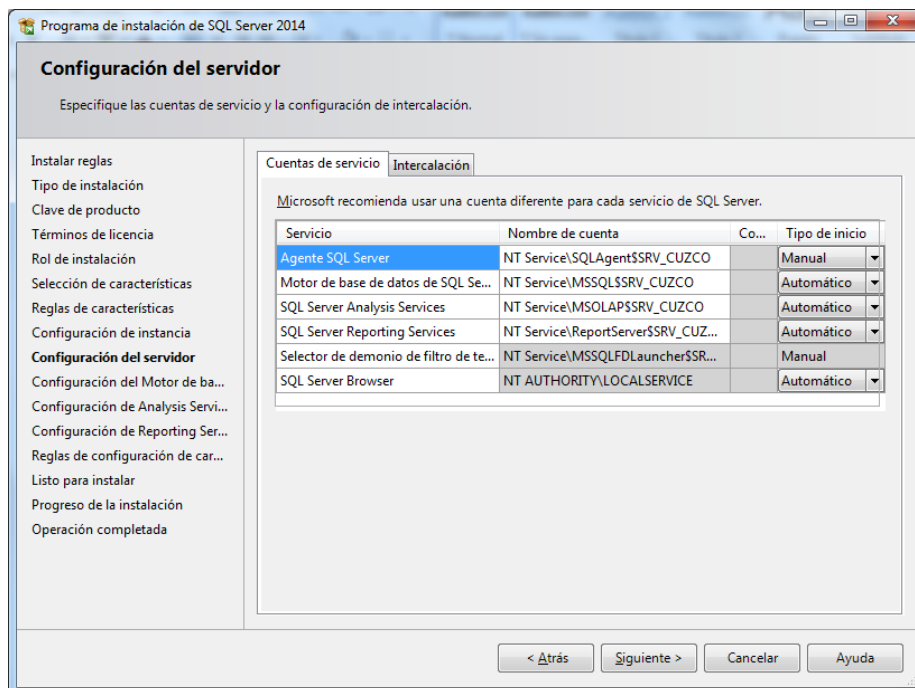
En la ventana Selección de características, dar un clic en el botón Seleccionar todo luego clic en Siguiente



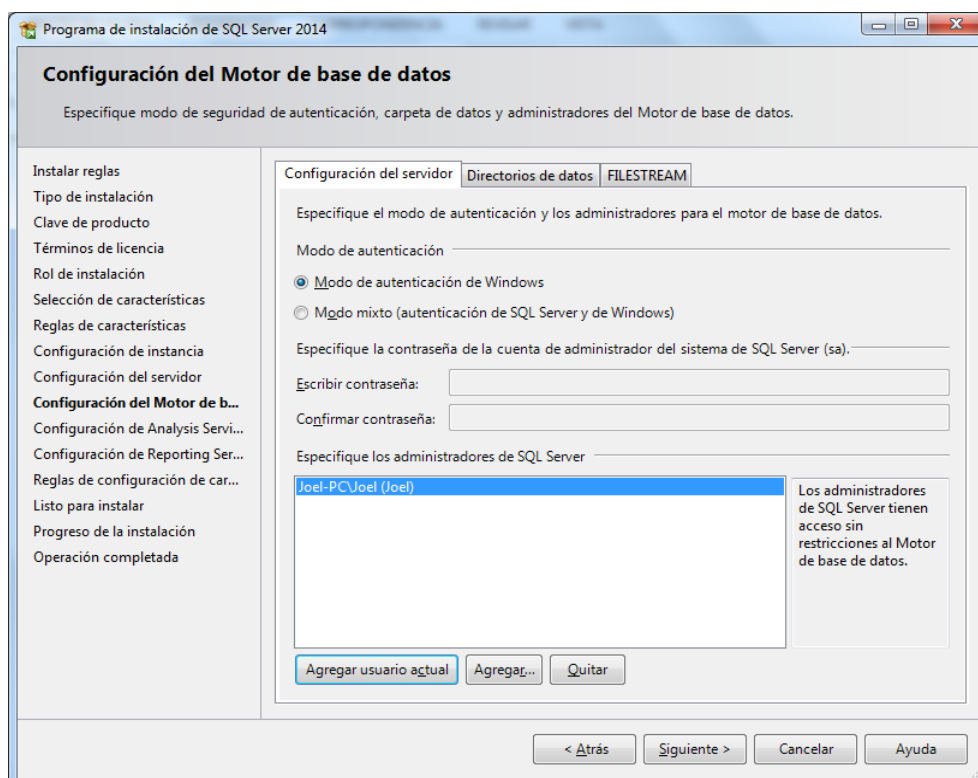
En la ventana Configuración de instancia, cuando instala por primera vez, debe estar seleccionada la opción Instancia predeterminada.



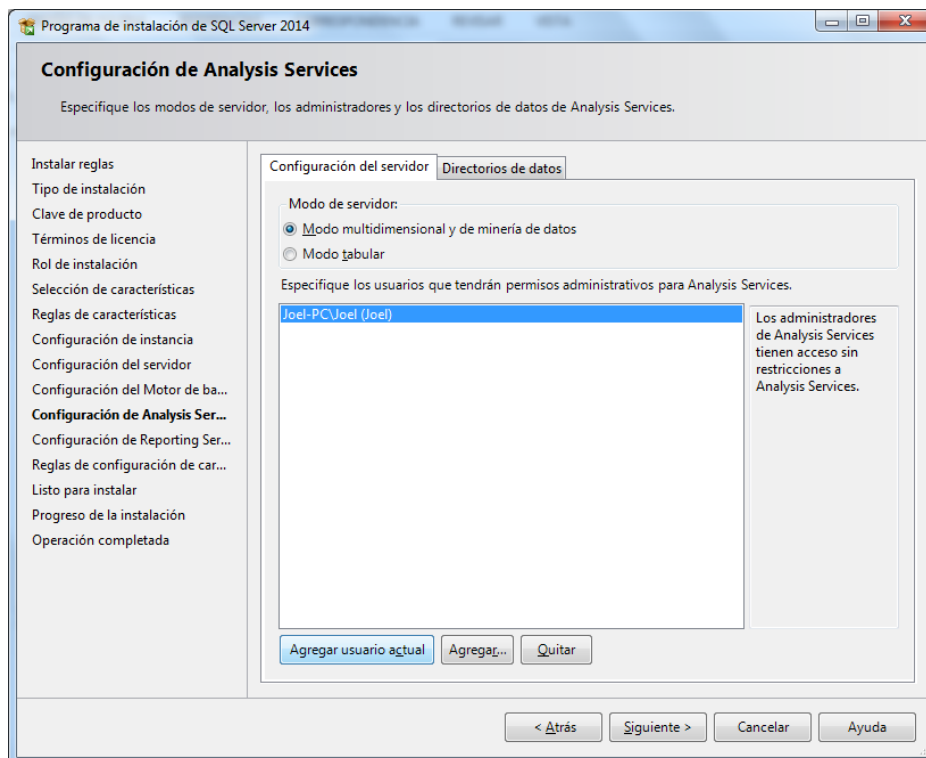
En la ventana Configuración del servidor, se debe establecer un nombre de cuenta para cada servicio y el tipo de inicio.



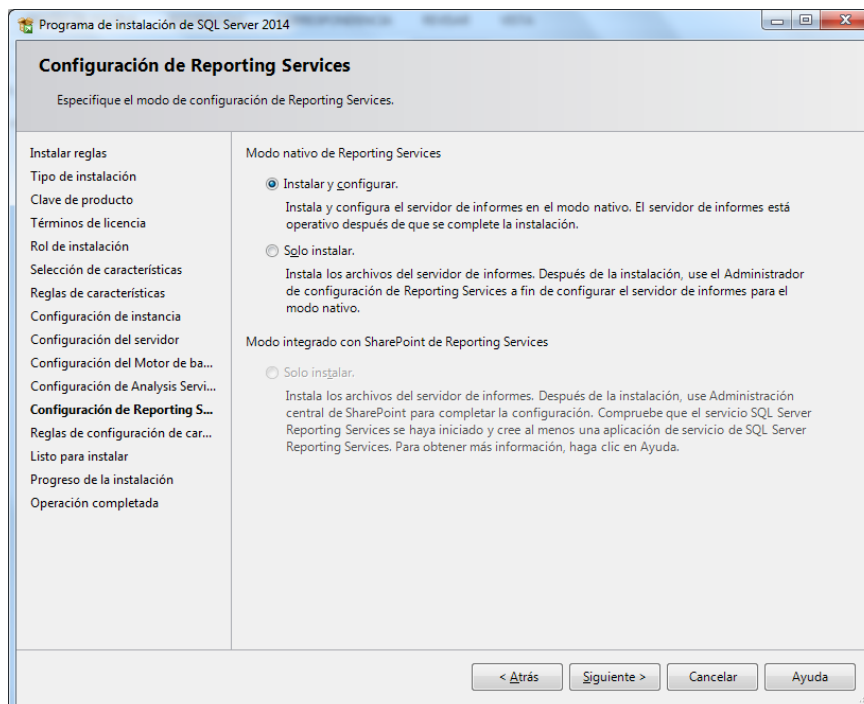
En la ventana Configuración del motor de base de datos, se debe especificar el administrador de base de datos. Puede dar un clic en el botón Agregar usuario actual.



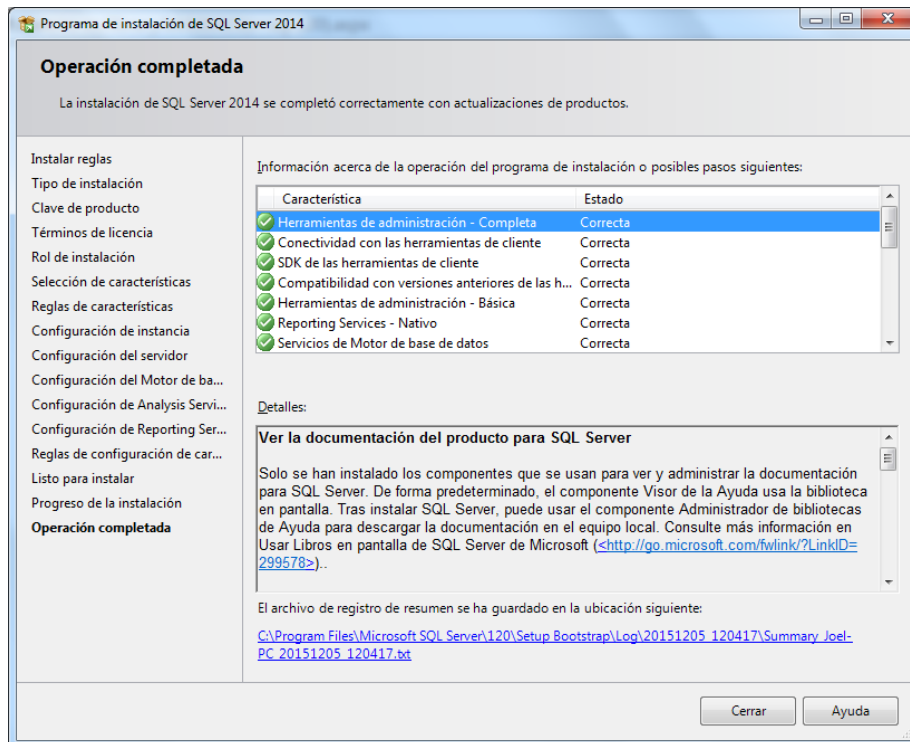
En la ventana Configuración de Analysis Services, dar un clic en el botón Agregar usuario actual.



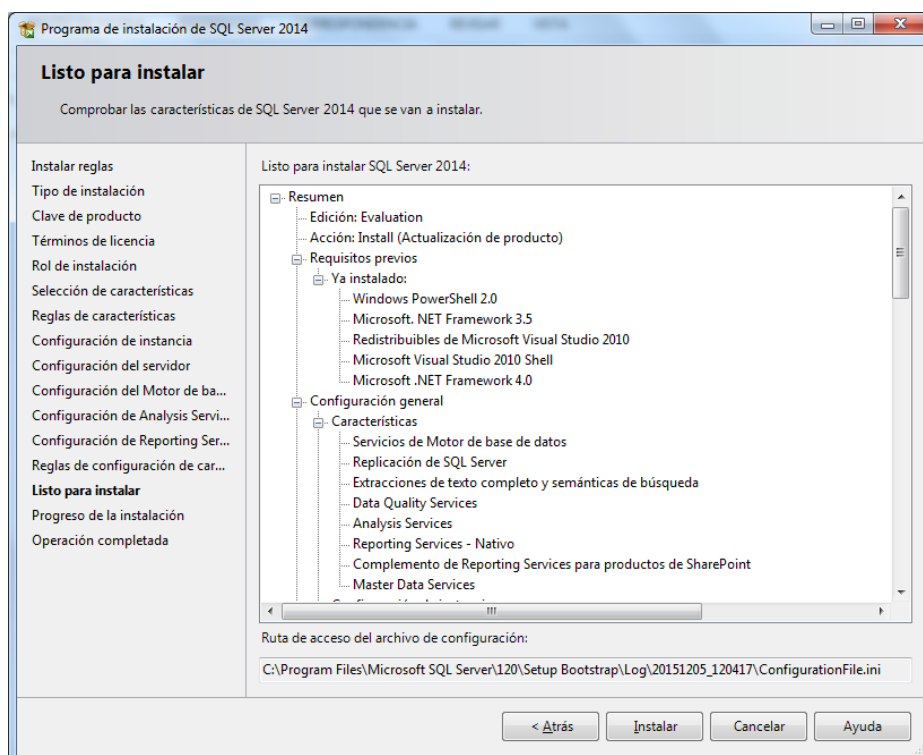
En la ventana Configuración de Reporting Services, dar un clic en el botón Instalar y Configurar.



En la ventana Operación completada dar un clic en el botón Cerrar.



En la ventana Listo para instalar dar un clic en el botón Instalar y finalizar la instalación.



# Conceptos básicos de Windows

## 1. Active Directory

El servicio de directorio basado en Windows. Active Directory almacena información acerca de los objetos de una red y la pone a disposición de los usuarios y administradores de la red. Active Directory da a los usuarios de red acceso a los recursos permitidos en cualquier punto de la red mediante un único proceso de inicio de sesión. Proporciona a los administradores de red una vista jerárquica intuitiva de la red y un punto de administración único para todos sus objetos.

## 2. Servicio de directorio

Tanto el origen de información del directorio como el servicio que hace que dicha información esté disponible para los usuarios. Un servicio de directorio permite a los usuarios encontrar un objeto con cualquiera de sus atributos.

## 3. Dominio

En Active Directory, un conjunto de objetos de equipo, usuario y grupo definido por el administrador. Estos objetos comparten una base de datos común de directorios, directivas de seguridad y relaciones de seguridad con otros dominios.

En DNS, cualquier árbol o subárbol del espacio de nombres DNS. Aunque los nombres de los dominios DNS suelen corresponder a los dominios de Active Directory, no deben confundirse con los dominios de Active Directory.

## 4. Objeto

Entidad, como un archivo, carpeta, carpeta compartida, impresora u objeto de Active Directory, descrita por un conjunto de atributos diferenciado y con nombre. Por ejemplo, algunos atributos de un objeto Archivo incluyen su nombre, ubicación y tamaño; entre los atributos de un objeto Usuario de Active Directory se cuentan el nombre del usuario, su apellido y su dirección de correo electrónico.

En OLE y ActiveX, un objeto puede ser también cualquier fragmento de información que se pueda vincular o incrustar en otro objeto.

## **5. Cuenta de usuario**

En Active Directory, objeto formado por toda la información que define un usuario del dominio, que incluye el nombre de usuario, la contraseña y los grupos a los que pertenece la cuenta de usuario. Las cuentas de usuario se pueden almacenar en Active Directory o en el equipo local.

En equipos donde se ejecuta Windows XP Professional y servidores miembros donde se ejecuta Windows Server 2003, utilice Usuarios y grupos locales para administrar cuentas de usuario locales. En controladores de dominio donde se ejecuta Windows Server 2003, utilice Usuarios y equipos de Active Directory para administrar cuentas de usuario de dominio.

## **6. Servicio**

Programa, rutina o proceso que realiza una determinada función del sistema para ofrecer compatibilidad con otros programas, especialmente a bajo nivel (cerca del hardware). Cuando se ofrecen servicios a través de una red, se pueden publicar en Active Directory, facilitando así la administración y el uso centralizados de los mismos. Algunos ejemplos son el servicio Administrador de cuentas de seguridad, Replicación de archivos y el Servicio de enrutamiento y acceso remoto.

## **7. Cuenta Administrador**

En un equipo local, la primera cuenta que se crea al instalar un sistema operativo en una nueva estación de trabajo, servidor independiente o servidor miembro. De forma predeterminada, esta cuenta tiene el nivel máximo de acceso administrativo al equipo local y es miembro del grupo Administradores.

En un dominio de Active Directory, la primera cuenta que se crea al configurar un nuevo dominio utilizando el Asistente para instalación de Active Directory. De forma predeterminada, esta cuenta tiene el nivel máximo de acceso administrativo en un



dominio y es miembro de los grupos Administradores, Administradores de dominio, Usuarios de dominio, Administradores de empresa, Propietarios del creador de directivas de grupo y Administradores de esquema.

## **8. Control de acceso**

Mecanismo de seguridad que determina qué operaciones está autorizado a realizar un usuario, grupo, servicio o equipo en un equipo o en un objeto determinado, como un archivo, una impresora, una subclave del Registro o un objeto del servicio de directorio.

## **9. Credenciales administrativas**

Información de inicio de sesión que se utiliza para identificar un miembro de un grupo administrativo. Entre los grupos que utilizan credenciales administrativas se incluyen Administradores, Administradores de dominio y Administradores DNS. La mayoría de las tareas para todo el sistema o todo el dominio requieren credenciales administrativas

## **10. Cuenta Invitado**

Cuenta integrada que se utiliza para iniciar una sesión en un equipo donde se ejecuta Windows, cuando un usuario no dispone de una cuenta en el equipo o dominio, o en cualquiera de los dominios de confianza del dominio del equipo.

## **11. Cuenta de usuario**

En Active Directory, objeto formado por toda la información que define un usuario del dominio, que incluye el nombre de usuario, la contraseña y los grupos a los que pertenece la cuenta de usuario. Las cuentas de usuario se pueden almacenar en Active Directory o en el equipo local. En equipos donde se ejecuta Windows XP Professional y servidores miembros donde se ejecuta Windows Server 2003, utilice Usuarios y grupos locales para administrar cuentas de usuario locales. En controladores de dominio donde se ejecuta Windows Server 2003, utilice Usuarios y equipos de Active Directory para administrar cuentas de usuario de dominio.

## **12. Contraseña**

Medida de seguridad para restringir los nombres de inicio de sesión a cuentas de usuario y el acceso a los sistemas y recursos.

Una contraseña es una cadena de caracteres que hay que suministrar para obtener la autorización para un acceso o un nombre de inicio de sesión. Las contraseñas pueden incluir letras, números y símbolos, y distinguen mayúsculas y minúsculas.

## **13. Nombre de usuario**

Nombre único que identifica una cuenta de usuario en Windows. El nombre de usuario de una cuenta debe ser único entre los demás nombres de grupo y nombres de usuario de su propio dominio o grupo de trabajo.

## **14. Control de acceso**

Mecanismo de seguridad que determina qué operaciones está autorizado a realizar un usuario, grupo, servicio o equipo en un equipo o en un objeto determinado, como un archivo, una impresora, una subclave del Registro o un objeto del servicio de directorio.

## **15. Grupo**

Conjunto de usuarios, equipos, contactos y otros grupos. Los grupos se pueden utilizar como conjuntos de distribución de correo electrónico o de seguridad. Los grupos de distribución sólo se utilizan para correo electrónico. Los grupos de seguridad se utilizan como listas de distribución de correo electrónico y para permitir el acceso a los recursos.

## **16. Autenticación**

Proceso que se utiliza para comprobar que una entidad o un objeto es quien dice ser. Algunos ejemplos son la confirmación del origen y la integridad de la información, como la comprobación de una firma digital o de la identidad de un usuario o equipo.

## **17. Relación de confianza**

Relación lógica establecida entre dominios para permitir la autenticación transferida, en la que un dominio que confía admite las autenticaciones de inicios de sesión del dominio de confianza. Las cuentas de usuario y grupos globales definidos en un dominio de confianza pueden recibir derechos y permisos en un dominio que confía, aunque las cuentas de usuario o los grupos no existan en el directorio de ese dominio.

## **18. Grupo global**

Grupo de seguridad o distribución que puede tener como miembros a usuarios, grupos y equipos de su propio dominio. Los grupos globales de seguridad pueden tener derechos y permisos sobre recursos de cualquier dominio del bosque.

## **19. Servidor miembro**

Un servidor que se ha unido a un dominio pero que no es un controlador de dominio. Los servidores miembros funcionan normalmente como servidores de archivos, servidores de aplicaciones, servidores de base de datos, servidores Web, servidores de certificados, servidores de seguridad o servidores de acceso remoto.

## **20. Grupo local**

Grupo de seguridad que puede tener derechos y permisos sólo sobre recursos del equipo en el que se haya creado el grupo. Los grupos locales pueden tener como miembros cualquier cuenta de usuario que sea local en el equipo, así como usuarios, grupos y equipos de un dominio al que pertenezca el equipo.

## **21. Privilegio**

Derecho de un usuario a realizar una tarea específica que suele afectar a un sistema completo en lugar de un objeto determinado. Los privilegios los asignan los administradores a usuarios individuales o grupos de usuarios como parte de la configuración de seguridad del equipo.

## **22. Derechos de inicio de sesión**

Derechos asignados a un usuario que especifican el modo en el que éste puede iniciar una sesión en un sistema. Un ejemplo de derecho de inicio de sesión es el de iniciar una sesión en un sistema de forma remota.

## **23. Controlador de dominio**

Los controladores de dominio que ejecutan Windows Server 2003 almacenan datos del directorio y administran las interacciones entre el usuario y el dominio, incluidos los procesos de inicio de sesión de los usuarios, la autenticación y las búsquedas en directorios. Los controladores de dominio se crean con el Asistente para instalación de Active Directory.

## **24. Grupo Administradores**

Los miembros del grupo Administradores tienen el máximo nivel de permisos predeterminados y pueden cambiar sus propios permisos

## **25. Servidores miembro**

Son servidores que se hacen miembros de un dominio con la finalidad de proporcionar servicios al dominio sin la obligación de mantener la lista de usuarios. Es recomendable que Exchange Server y SQL Server se configuren como servidores miembros.

## **26. Clientes**

Son equipos con algunos de los siguientes sistemas operativos: DOS, Windows 95, Windows xp, que requieren los servicios y recursos del dominio.

## Capítulo 02

# Correo electrónico de base de datos

Correo electrónico de base de datos es una solución empresarial para enviar mensajes de correo electrónico desde el SQL Server 2008 Database Engine (Motor de base de datos de SQL Server 2008) de Microsoft. El Correo electrónico de base de datos permite a las aplicaciones de base de datos enviar mensajes de correo electrónico a los usuarios. Los mensajes enviados pueden incluir resultados de consultas y archivos de cualquier recurso de la red. El Correo electrónico de base de datos está diseñado para proporcionar confiabilidad, escalabilidad, seguridad y compatibilidad.

### Confiabilidad

No se requiere Microsoft Outlook ni MAPI (Interfaz de programación de aplicaciones de mensajería) extendida. El Correo electrónico de base de datos utiliza el protocolo estándar SMTP (Protocolo simple de transferencia de correo) para enviar correo electrónico. Puede utilizar el Correo electrónico de base de datos sin necesidad de instalar un cliente con MAPI extendida en el equipo que ejecuta SQL Server.

Aislamiento de procesos. Para minimizar el impacto en SQL Server, el componente que proporciona el correo electrónico se ejecuta fuera de SQL Server, en un proceso separado. SQL Server continuará poniendo en cola los mensajes de correo electrónico aunque el proceso externo se detenga o se produzca un error. Los mensajes en cola se enviarán cuando el proceso externo o el servidor SMTP recuperen la conexión.

### Escalabilidad

Entrega en segundo plano. El Correo electrónico de base de datos permite realizar entregas en segundo plano o asincrónicas. Cuando se llama a `sp_send_dbmail` para enviar un mensaje, el Correo electrónico de base de datos agrega una solicitud a una

cola de Service Broker. El procedimiento almacenado se devuelve inmediatamente. El componente de correo electrónico externo recibe la solicitud y entrega el mensaje.

Varios perfiles. El Correo electrónico de base de datos permite crear varios perfiles en una instancia de SQL Server. También se puede seleccionar el perfil del Correo electrónico de base de datos para enviar el mensaje.

## **Seguridad**

Desactivado de forma predeterminada. Para reducir la superficie de SQL Server, los procedimientos almacenados del Correo electrónico de base de datos están desactivados de forma predeterminada. Utilice la herramienta Configuración de superficie de SQL Server para habilitar el procedimiento almacenado del Correo electrónico de base de datos. Para enviar correo electrónico de base de datos, debe pertenecer a la función de base de datos DatabaseMailUserRole en la base de datos msdb.

## **Compatibilidad**

Configuración integrada. El Correo electrónico de base de datos mantiene la información para las cuentas de correo electrónico del SQL Server Database Engine (Motor de base de datos de SQL Server). No es necesario administrar un perfil de correo en una aplicación cliente externa. El Asistente para configuración del Correo electrónico de base de datos proporciona una interfaz adecuada para configurar el Correo electrónico de base de datos. También se pueden crear y mantener configuraciones del Correo electrónico de base de datos mediante Transact-SQL.

**Registro.** El Correo electrónico de base de datos registra la actividad de correo electrónico en SQL Server, en el registro de sucesos de aplicación de Microsoft Windows y en la base de datos msdb.

**Auditoría.** El Correo electrónico de base de datos conserva copias de los mensajes y datos adjuntos enviados en la base de datos msdb. Puede auditar fácilmente la utilización del Correo electrónico de base de datos y revisar los mensajes conservados.

**Compatibilidad con HTML.** El Correo electrónico de base de datos permite enviar mensajes de correo electrónico con el formato HTML.

El Correo electrónico de base de datos consta de cuatro componentes principales:

1. Componentes de seguridad y configuración
2. Componentes de mensajería
3. Ejecutable del Correo electrónico de base de datos
4. Componentes de registro y auditoría

El Correo electrónico de base de datos almacena información de configuración y seguridad en la base de datos msdb. La base de datos msdb actúa como la base de datos host de correo que contiene los objetos de mensajería que utiliza el Correo electrónico de base de datos para enviar correo electrónico. Esto incluye el procedimiento almacenado **sp\_send\_dbmail** y las estructuras de datos que contienen información acerca de los mensajes. El ejecutable del Correo electrónico de base de datos es un programa externo que lee en una cola de la base de datos msdb y envía mensajes a servidores de correo electrónico. El Correo electrónico de base de datos registra información de registro en la base de datos msdb y el registro de sucesos de aplicación de Microsoft Windows.

Los objetos de configuración y seguridad crean perfiles y cuentas usadas por el Correo electrónico de base de datos. Los objetos de mensajería se utilizan para enviar mensajes de correo electrónico. Los componentes adicionales registran y supervisan el estado del Correo electrónico de base de datos.

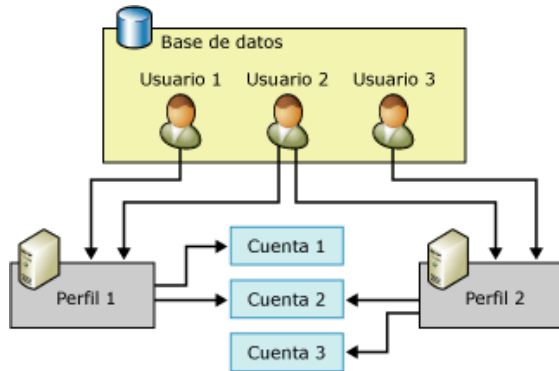
Correo electrónico de base de datos proporciona dos tipos de objetos de configuración:

Los perfiles de Correo electrónico de base de datos son conjuntos de cuentas.

Las cuentas de Correo electrónico de base de datos contienen información para los servidores de correo electrónico.

El Correo electrónico de base de datos incluye procedimientos almacenados de configuración en la base de datos msdb. Para utilizar estos procedimientos directamente se pueden escribir secuencias de comandos Transact-SQL.

En la siguiente ilustración se muestra la relación entre las cuentas, los perfiles y los usuarios. Observe que un usuario puede tener permiso para utilizar más de un perfil, pero sólo se puede definir un perfil como predeterminado. Cada perfil puede utilizar más de una cuenta. Varios perfiles pueden utilizar una misma cuenta.



En la siguiente ilustración se muestran dos perfiles, tres cuentas y tres usuarios. El usuario 1 tiene acceso al perfil 1, que utiliza la cuenta 1 y la cuenta 2. El usuario 3 tiene acceso al perfil 2, que utiliza la cuenta 2 y la cuenta 3. El usuario 2 tiene acceso al perfil 1 y al perfil 2.

## Perfil de correo

Un perfil del Correo electrónico de base de datos es una colección ordenada de cuentas relacionadas del Correo electrónico de base de datos. Para enviar correo, debe ser un miembro de la función DatabaseMailUserRole en la base de datos msdb y tener acceso como mínimo a un perfil de Correo electrónico de base de datos. Los perfiles permiten a los administradores de bases de datos volver a configurar procedimientos almacenados y aplicaciones de bases de datos que utilizan el correo electrónico sin modificar el código de la aplicación.

## Cuenta de correo

Una cuenta del Correo electrónico de base de datos contiene la información que Microsoft SQL Server utiliza para enviar mensajes de correo electrónico a un servidor SMTP. Cada cuenta contiene información para un servidor de correo electrónico.



El Correo electrónico de base de datos permite el uso de tres métodos de autenticación para comunicarse con un servidor SMTP:

- Autenticación de Windows: el Correo electrónico de base de datos utiliza las mismas credenciales de la cuenta del servicio SQL Server Database Engine (Motor de base de datos de SQL Server) Windows para la autenticación en el servidor SMTP.
- Autenticación básica: el Correo electrónico de base de datos utiliza el nombre de usuario y la contraseña especificados para la autenticación en el servidor SMTP.
- Autenticación anónima: el servidor SMTP no requiere ninguna autenticación. El Correo electrónico de base de datos no utilizará ninguna credencial para la autenticación en el servidor SMTP.

La información de las cuentas se almacena en la base de datos msdb. Cada cuenta contiene la siguiente información:

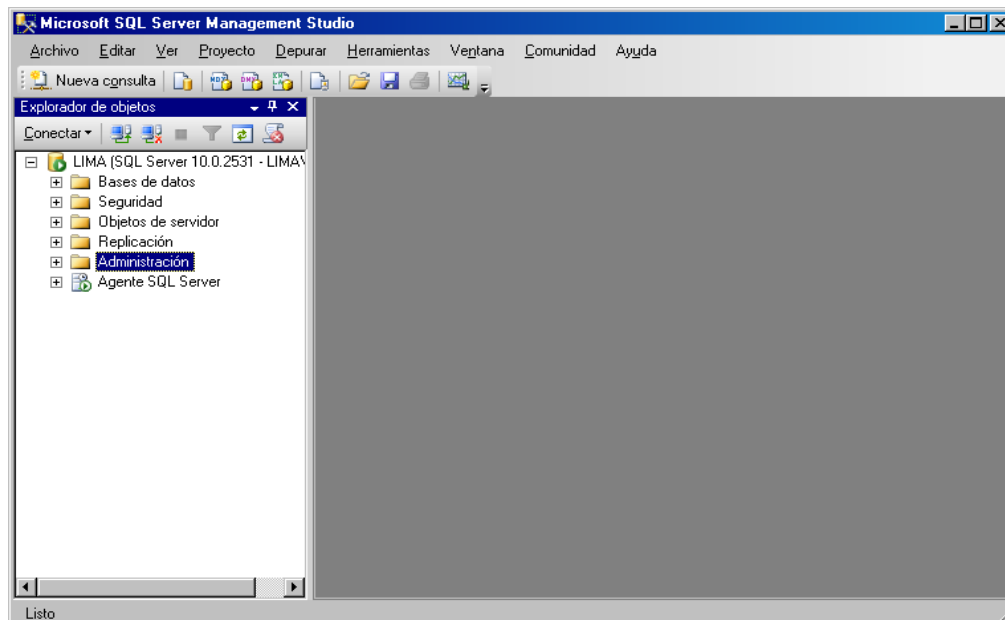
- El nombre de la cuenta.
- La descripción de la cuenta.
- La dirección de correo electrónico de la cuenta.
- El nombre para mostrar de la cuenta.
- La dirección de correo electrónico que se utilizará como información de "Responder a" para la cuenta.
- El nombre del servidor de correo electrónico.
- El tipo de servidor de correo electrónico. Para Microsoft SQL Server 2008, siempre es el Protocolo simple de transferencia de correo (SMTP).
- El número de puerto del servidor de correo electrónico.

- Una columna de tipo bit que indica si se ha realizado la conexión al servidor de correo SMTP con SSL (Capa de sockets seguros).
- Una columna de tipo bit que indica si se ha realizado la conexión al servidor SMTP con las credenciales configuradas para el SQL Server Database Engine (Motor de base de datos de SQL Server).
- El nombre de usuario que se utiliza para la autenticación en el servidor de correo electrónico, si éste requiere autenticación.
- La contraseña que se utiliza para la autenticación en el servidor de correo electrónico, si éste requiere autenticación.

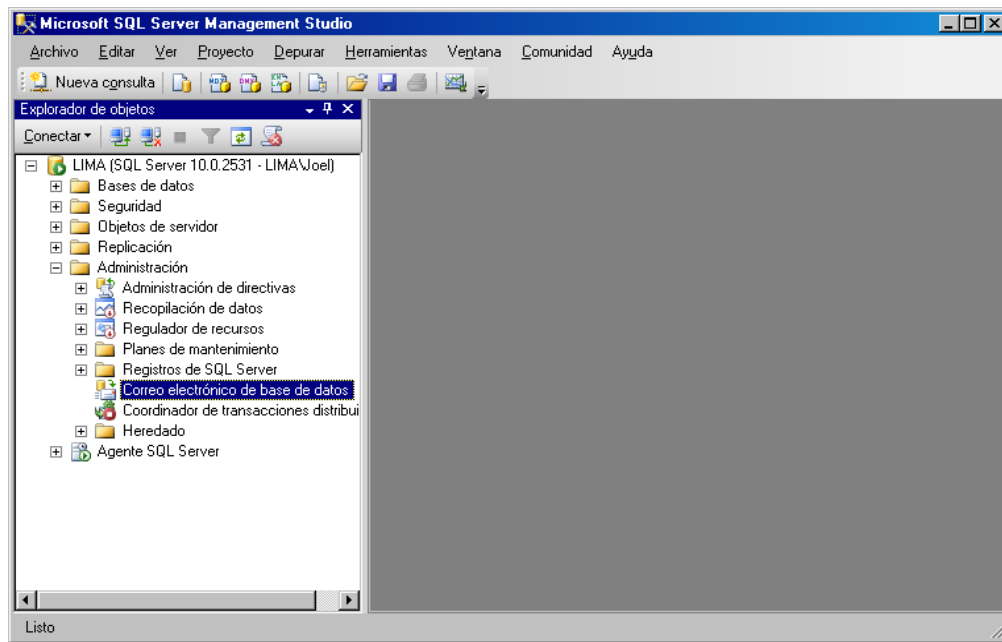
## Crear perfiles y cuentas de correo electrónico utilizando el asistente

El Asistente para configuración de Correo electrónico de base de datos proporciona una forma adecuada de crear y administrar cuentas.

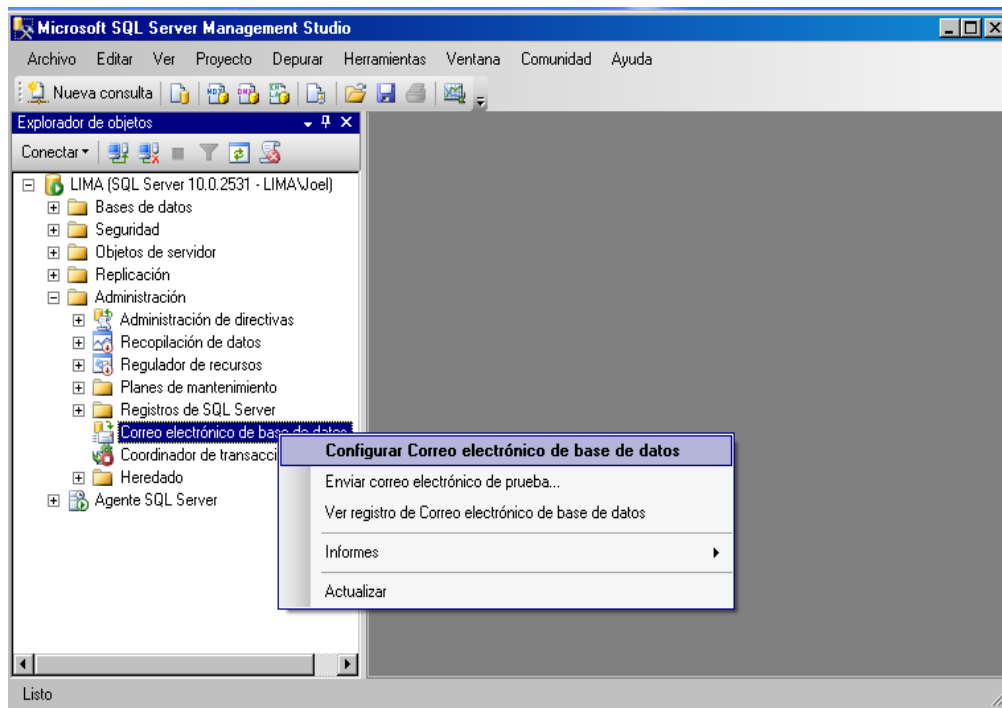
1. En el explorador de objetos ampliar la carpeta Administración



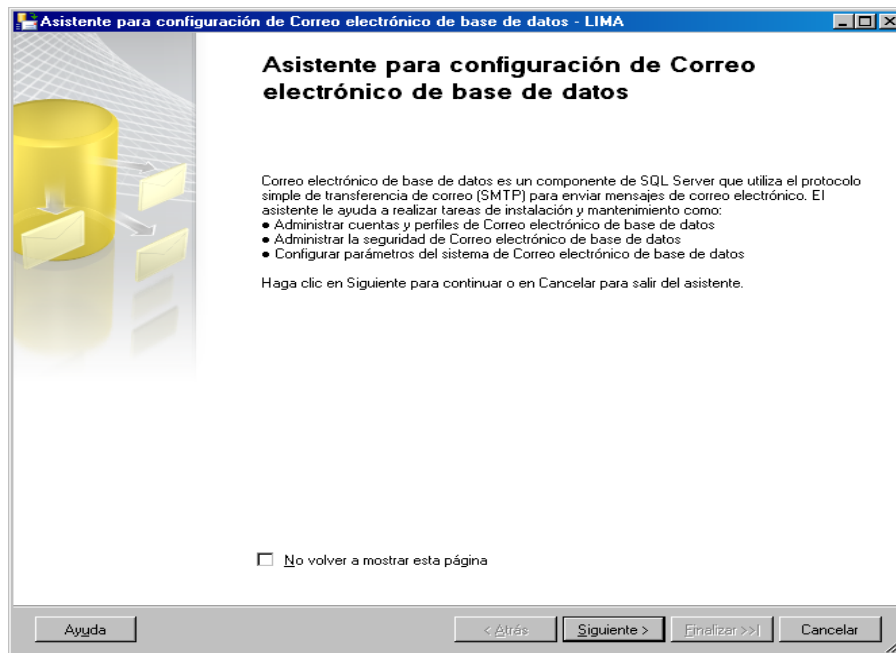
2. Seleccionar la opción Correo electrónico de base de datos



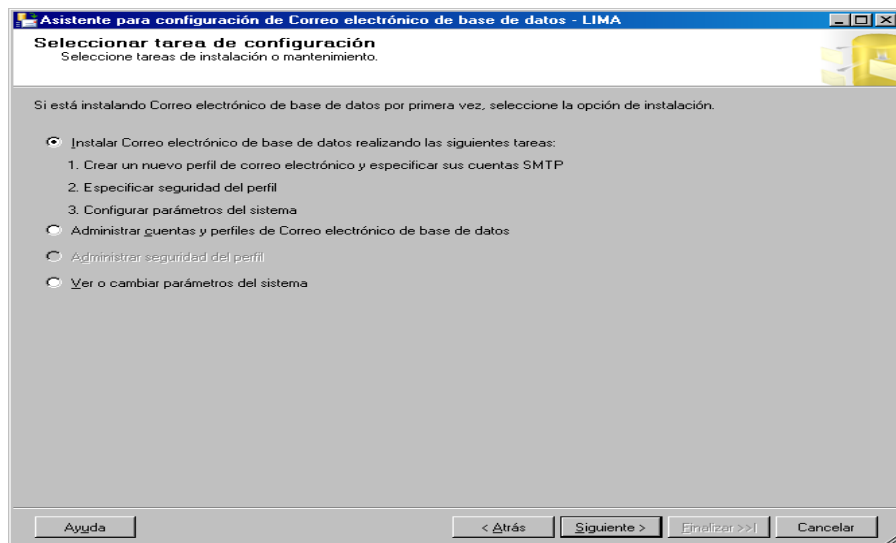
3. Hacer clic con el botón secundario del mouse y seleccionar la opción Configurar Correo electrónico de base de datos.



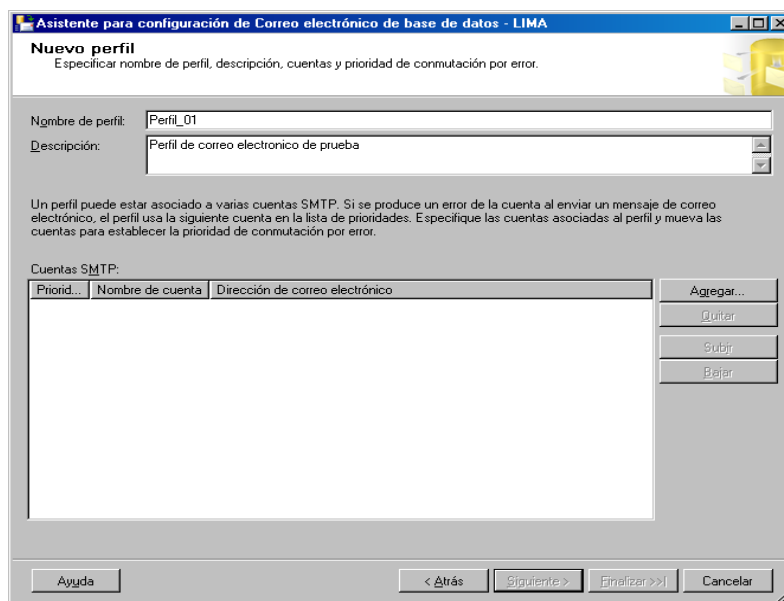
4. Se activa el asistente para configurar el correo electrónico de base de datos: clic en el botón siguiente.



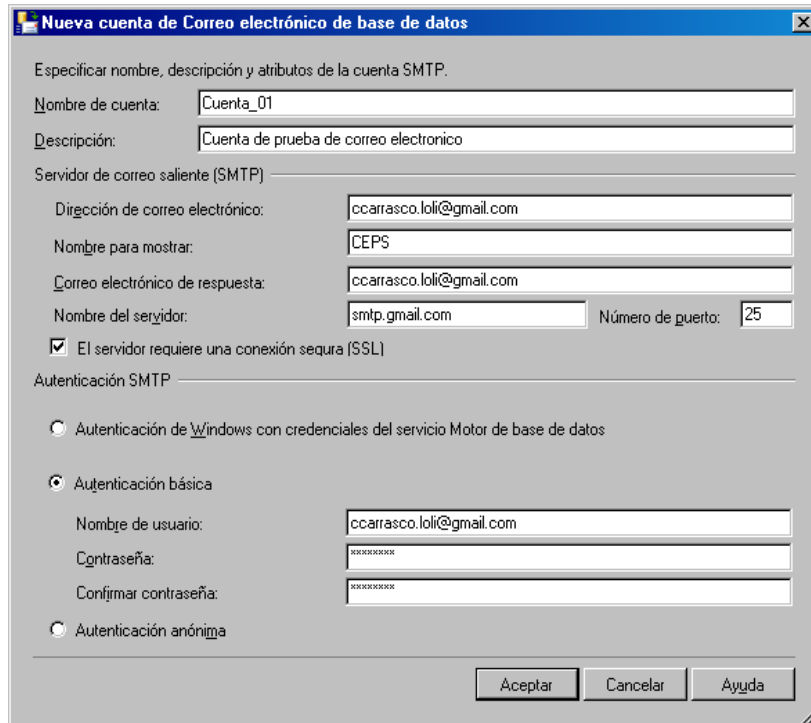
5. En la siguiente ventana seleccionar la primera opción y luego dar un clic en el botón Siguiente



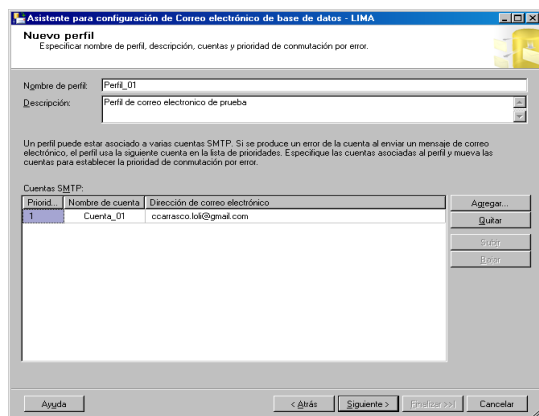
6. En la ventana presente ingresar el nombre del nuevo perfil y de forma opcional una breve descripción. Para el ejemplo propuesto el nombre del perfil es Perfil\_01.



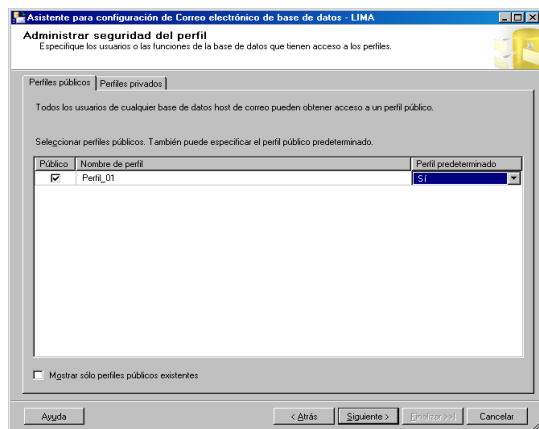
7. Para agregar una nueva cuenta al perfil creado, dar un clic en el botón Agregar, se presentara la siguiente ventana. Para el ejemplo se ha creado previamente una cuenta de correo en gmail. Si desea utilizar otro servicio de correo debe averiguar el nombre del servidor y el número de puerto para ese servicio.



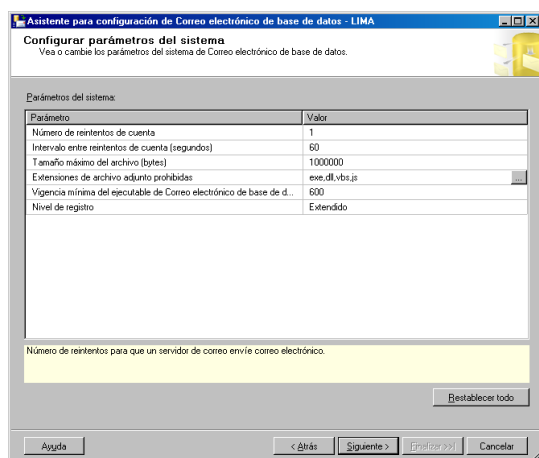
8. En la ventana presente continuar dar un clic en el botón Siguiente.



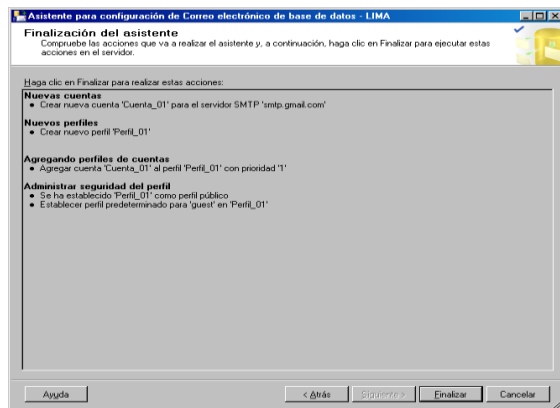
9. A continuación seleccionar el check Público y seleccionar como perfil predeterminado.



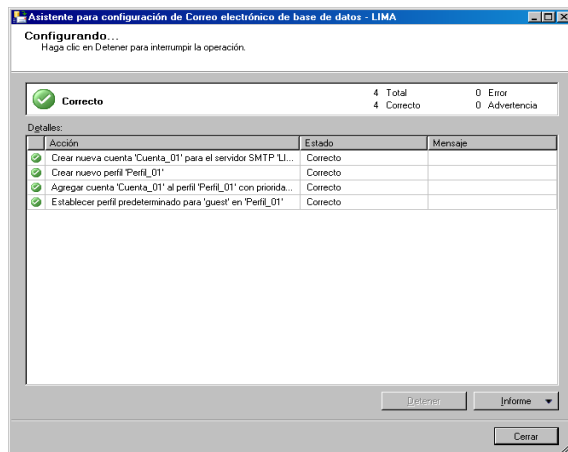
10. La ventana siguiente muestra los parámetros del sistema, para continuar dar un clic en el botón siguiente.



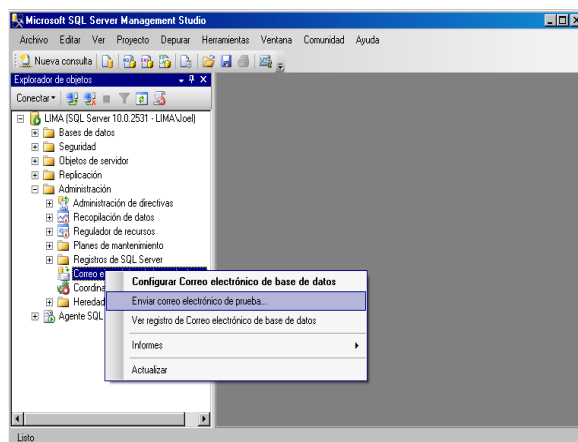
11. Dar un clic en el botón Finalizar.



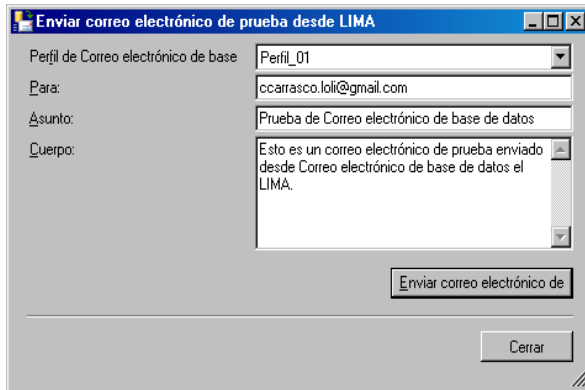
12. Para terminar dar un clic en el botón Cerrar.



13. Para probar seleccione Correo electrónico de base de datos, luego clic con el botón derecho, del menú contextual seleccionar la opción Enviar correo electrónico de prueba.



14. En la siguiente ventana seleccionar el perfil de correo a utilizar, ingresar la dirección de correo a enviar y finalmente un clic en el botón Enviar correo electrónico de. Revisar su cuenta de correo y verificar que se envió el correo.



## Utilizando procedimientos almacenados para crear un perfil de correo electrónico

--01 Crear un perfil de correo de base de datos

```
EXECUTE msdb.dbo.sysmail_add_profile_sp
```

```
@profile_name = 'Perfil de Prueba 2',
```

```
@description = 'Perfil utilizado para administrar correo.' ;
```

## Utilizando procedimientos almacenados para crear una cuenta de correo electrónico

--02 Crear una cuenta de correo de base de datos

```
EXECUTE msdb.dbo.sysmail_add_account_sp
```

```
@account_name = 'Cuenta administrador 2',
```

```
@description = 'Cuenta de correo para administrar correo.',
```

```
@email_address = 'ccarrasco.loli@gmail.com',
```

```
@replyto_address = 'ccarrasco.loli@gmail.com',
```

```
@port = 587,-- en la UNI utilizar 25
```

```
@username = 'ccarrasco.loli@gmail.com',
```

```
@password = 'P@ssw0rd',
```



```
@use_default_credentials =0,  
  
@enable_ssl = 1,  
  
@display_name = 'CEPS UNI',  
  
@mailserver_name = 'smtp.gmail.com' ;
```

### **Utilizando procedimientos almacenados para agregar una cuenta a un perfil.**

--03 Agregar la cuenta al perfil

```
EXECUTE msdb.dbo.sysmail_add_profileaccount_sp  
  
@profile_name = 'Perfil de Prueba 2',  
  
@account_name = 'Cuenta administrador 2',  
  
@sequence_number =1 ;
```

### **Utilizando procedimientos almacenados para otorgar permiso que permita utilizar un perfil de correo electrónico**

--04 Concede permiso a un usuario para utilizar el perfil de correo de base de datos

```
EXECUTE msdb.dbo.sysmail_add_principalprofile_sp  
  
@profile_name = 'Perfil de Prueba 2',  
  
@principal_name = 'guest',  
  
@is_default = 1 ;
```

### **Utilizando procedimiento almacenado para eliminar un perfil de correo electrónico**

-- 05 Eliminar un perfil de base de datos

```
EXECUTE msdb.dbo.sysmail_delete_profile_sp  
  
@profile_name = 'Perfil de prueba 1' ;
```

## **Utilizando procedimiento almacenado para eliminar una cuenta de correo electrónico.**

-- 06 Eliminar una cuenta

```
EXECUTE msdb.dbo.sysmail_delete_account_sp
```

```
@account_name = 'Cuenta administrador 1' ;
```

## **Utilizando procedimiento almacenado para todos los eventos del registro de correo electrónico**

-- 07 Elimina todos los eventos del registro del correo electrónico

```
EXECUTE msdb.dbo.sysmail_delete_log_sp ;
```

```
GO
```

## **El procedimiento almacenado sp\_send\_dbmail**

Envía un mensaje de correo electrónico a los destinatarios especificados. El mensaje puede incluir un conjunto de resultados de una consulta, archivos adjuntos o ambos elementos. Si el mensaje se coloca correctamente en la cola del Correo electrónico de base de datos, sp\_send\_dbmail devuelve el valor de mailitem\_id del mensaje. Este procedimiento almacenado se encuentra en la base de datos msdb.

Sintaxis:

```
[ @profile_name = ] 'profile_name'
```

Nombre del perfil desde el que se va a enviar el mensaje. profile\_name es de tipo sysname y su valor predeterminado es NULL. profile\_name debe ser el nombre de un perfil existente de Correo electrónico de base de datos. Si no se especifica profile\_name, sp\_send\_dbmail utiliza el perfil privado predeterminado del usuario actual. Si el usuario no tiene un perfil privado predeterminado, sp\_send\_dbmail utiliza el perfil público predeterminado de la base de datos msdb. Si el usuario no tiene un perfil privado

predeterminado y no hay ningún perfil público predeterminado para la base de datos, debe especificarse @profile\_name.

[ @recipients = ] 'recipients'

Lista de direcciones de correo electrónico, separadas por punto y coma, a las que se va a enviar el mensaje. La lista de destinatarios es de tipo varchar(max). Aunque este parámetro es opcional, debe especificarse como mínimo uno de los valores @recipients, @copy\_recipients o @blind\_copy\_recipients, de lo contrario sp\_send\_dbmail devuelve un error.

[ @copy\_recipients = ] 'copy\_recipients'

Lista de direcciones de correo electrónico, separadas por punto y coma, que van a recibir copia del mensaje. La lista de destinatarios que recibirán copia es de tipo varchar(max). Aunque este parámetro es opcional, debe especificarse como mínimo uno de los valores @recipients, @copy\_recipients o @blind\_copy\_recipients, de lo contrario sp\_send\_dbmail devuelve un error.

[ @blind\_copy\_recipients = ] 'blind\_copy\_recipients'

Lista de direcciones de correo electrónico, separadas por punto y coma, que van a recibir copia oculta del mensaje. La lista de destinatarios de la copia oculta es de tipo varchar(max). Aunque este parámetro es opcional, debe especificarse como mínimo uno de los valores @recipients, @copy\_recipients o @blind\_copy\_recipients, de lo contrario sp\_send\_dbmail devuelve un error.

[ @subject = ] 'subject'

Asunto del mensaje de correo electrónico. El asunto es de tipo nvarchar(255). Si no se especifica ningún asunto, el valor predeterminado es 'Mensaje de SQL Server'.

[ @body = ] 'body'

Cuerpo del mensaje de correo electrónico. El cuerpo del mensaje es de tipo nvarchar(max) y su valor predeterminado es NULL.

[ @body\_format = ] 'body\_format'

Formato del cuerpo del mensaje. El parámetro es de tipo varchar(20) y su valor predeterminado es NULL. Si se especifica, los encabezados del mensaje saliente indican que el cuerpo del mensaje tiene el formato especificado. El parámetro puede contener uno de los siguientes valores:

- \* TEXT

- \* HTML

El valor predeterminado es TEXT.

@importance = ] 'importance'

Hace referencia a la importancia del mensaje. El parámetro es de tipo varchar(6). El parámetro puede contener uno de los siguientes valores:

- \* Low

- \* Normal

- \* High

El valor predeterminado es Normal.

[ @query = ] 'query'

Consulta que se va a ejecutar. Los resultados de la consulta pueden adjuntarse como archivo o incluirse en el cuerpo del mensaje de correo electrónico. La consulta es de tipo nvarchar(max) y puede contener cualquier instrucción de Transact-SQL válida. Tenga en cuenta que la consulta se ejecuta en una sesión independiente, por lo que las variables locales de la secuencia de comandos que llama a sp\_send\_dbmail no están disponibles para la consulta.

[ @attach\_query\_result\_as\_file = ] attach\_query\_result\_as\_file

Especifica si el conjunto de resultados de la consulta se devuelve como un archivo adjunto. attach\_query\_result\_as\_file es de tipo bit y su valor predeterminado es 0.

Si el valor es 0, los resultados de la consulta se incluyen en el cuerpo del mensaje de correo electrónico, después del contenido del parámetro @body. Si el valor es 1, los resultados se devuelven como archivo adjunto. Este parámetro sólo se puede aplicar si se especifica @query.

## Ejemplos

### --01 Uso del correo electrónico enviando un mensaje sencillo.

```
EXEC msdb.dbo.sp_send_dbmail
```

```
@profile_name = 'Perfil de Prueba 2',  
@recipients = 'ccarrasco.loli@gmail.com',  
@body = 'El procedimiento se ejecuto con éxito',  
@subject = 'Mensaje de éxito automatizado ' ;
```

### --02 Se envía el resultado de una consulta

```
EXEC msdb.dbo.sp_send_dbmail
```

```
@profile_name = 'Perfil de Prueba 2',  
@recipients = 'ccarrasco.loli@gmail.com',  
@query = 'SELECT COUNT(*) FROM Northwind.dbo.Customers',  
@subject = 'Numero de clientes',  
@attach_query_result_as_file = 1 ;
```

### --03 Se utiliza un Trigger

```
USE pubs
```

```
GO
```

```
IF EXISTS (SELECT name FROM sysobjects
```

```
    WHERE name = 'reminder' AND type = 'TR')
```

```
    DROP TRIGGER reminder
```

```
GO
```

CREATE TRIGGER reminder

ON titles

FOR INSERT, UPDATE, DELETE

AS

EXEC msdb.dbo.sp\_send\_dbmail

    @profile\_name = 'Perfil de Prueba 2',

    @recipients = 'ccarrasco.loli@gmail.com',

    @body = 'Se realizaron cambios en la base de datos',

    @subject = 'Prueba de correo' ;

GO

**--04 Se actualiza la tabla TITLES, entonces debe llegar un correo electrónico al destinatario.**

UPDATE TITLES

SET price=50

WHERE title\_id='BU1032'

## **Solucionar problemas**

**--01 Determinar si los usuarios están correctamente configurados para enviar mensajes del Correo electrónico de base de datos**

EXECUTE msdb.sys.sp\_helprolemember 'DatabaseMailUserRole'

**--02 Para agregar usuarios a la función DatabaseMailUserRole, utilice la instrucción siguiente**

USE msdb

GO

EXECUTE sp\_addrolemember @rolename = 'DatabaseMailUserRole',

@membername = 'guest'

**---03 Para enviar mensajes del Correo electrónico de base de datos, los usuarios deben tener acceso al menos a un perfil del Correo electrónico de base de datos. Para obtener una lista de los usuarios (entidades de seguridad) y los perfiles a los que tienen acceso, ejecute la instrucción que se indica a continuación.**

EXECUTE msdb.dbo.sysmail\_help\_principalprofile\_sp ;

**--04 El Programa externo Correo electrónico de base de datos se activa cuando hay mensajes de correo electrónico que se deben procesar. El programa termina cuando ya no hay mensajes para enviar durante el período de tiempo de espera especificado. Para confirmar que se ha iniciado la activación del Correo electrónico de base de datos, ejecute la instrucción que se indica a continuación.**

EXECUTE msdb.dbo.sysmail\_help\_status\_sp ;

**--05 Si no se ha iniciado la activación del Correo electrónico de base de datos, ejecute la instrucción que se indica a continuación:**

EXECUTE msdb.dbo.sysmail\_start\_sp

**--06 Si se ha iniciado el programa externo del Correo electrónico de base de datos, compruebe el estado de la cola de correo electrónico con la instrucción siguiente:**

EXECUTE msdb.dbo.sysmail\_help\_queue\_sp @queue\_type = 'mail' ;

La cola debe tener el estado RECEIVES\_OCCURRING. Su estado puede variar según el momento. Si el estado de la cola de correo no es RECEIVES\_OCCURRING, intente detenerla mediante sysmail\_stop\_sp e iniciarla mediante sysmail\_start\_sp.

**--07 Si ha determinado que sólo algunos perfiles pueden enviar correo, puede que tenga problemas con las cuentas del Correo electrónico de base de datos utilizadas por los perfiles problemáticos. Para determinar qué cuentas envían correo correctamente, ejecute la instrucción siguiente:**

SELECT sent\_account\_id, sent\_date FROM msdb.dbo.sysmail\_sentitems ;

**--08 Para ver los mensajes de error devueltos por el Correo electrónico de base de datos, ejecute la instrucción siguiente:**

SELECT \* FROM msdb.dbo.sysmail\_event\_log ;

**--09 Para eliminar los mensajes de error**

DELETE FROM msdb.dbo.sysmail\_event\_log;

## Capítulo 03

# Administración de la seguridad

Una de las tareas más importantes de un administrador de base de datos es la administración de la seguridad. Si la seguridad del sistema es violada los datos se pueden perder o corromper.

Para garantizar que sólo los usuarios autorizados obtienen acceso a los datos y objetos almacenados en Microsoft® SQL Server, la seguridad se debe configurar correctamente. Entre los elementos de seguridad que puede ser necesario configurar están los modos de autenticación, los inicios de sesión, los usuarios, las funciones, otorgar, revocar y denegar permisos de objetos e instrucciones Transact-SQL, y el cifrado de datos.

### Entidades de seguridad

Las entidades de seguridad son entidades que pueden solicitar recursos de SQL Server. Igual que otros componentes del modelo de autorización de SQL Server, las entidades de seguridad se pueden organizar en jerarquías. El ámbito de influencia de una entidad de seguridad depende del ámbito de su definición: Windows, servidor o base de datos; y de si la entidad de seguridad es indivisible o es una colección. Un Inicio de sesión de Windows es un ejemplo de entidad de seguridad indivisible y un Grupo de Windows es un ejemplo de una del tipo colección. Toda entidad de seguridad tiene un identificador de seguridad (SID).

### Entidades de seguridad a nivel de Windows

- Inicio de sesión del dominio de Windows
- Inicio de sesión local de Windows



## Entidad de seguridad de SQL Server

- Inicio de sesión de SQL Server

## Entidades de seguridad a nivel de bases de datos

- Usuario de base de datos
- Función de base de datos
- Función de aplicación

## Asegurables

Los asegurables son los recursos cuyo acceso es regulado por el sistema de autorización del SQL Server Database Engine (Motor de base de datos de SQL Server). Algunos asegurables pueden estar incluidos en otros, con lo que se crean jerarquías anidadas denominadas "ámbitos" que a su vez se pueden asegurar. Los ámbitos asegurables son servidor, base de datos y esquema. A los asegurables se les pueden asignar permisos.

## Esquema

Un esquema es un contenedor que contiene tablas, vistas, procedimientos, etc. Se encuentra dentro de una base de datos, que a su vez está dentro de un servidor. Estas entidades se acomodan como cajas anidadas. El servidor es la caja más externa y el esquema la más interna. La creación de esquemas facilita la administración de la seguridad, ya que se pueden administrar permisos sobre los esquemas.

## La instrucción CREATE SCHEMA

Permite crear un esquema en la base de datos actual.

```
CREATE SCHEMA schema_name_clause [ <schema_element> [ ...n ] ]
```

```
<schema_name_clause> ::=  
    {  
        schema_name  
        | AUTHORIZATION owner_name  
        | schema_name AUTHORIZATION owner_name  
    }
```

```
<schema_element> ::=  
    {  
        table_definition | view_definition | grant_statement |  
        revoke_statement | deny_statement  
    }
```

Ejemplo:

Crear la base de datos Mercado

Crear el esquema Produccion y luego en dicho esquema crear la tabla Producto

```
--01 Activar la base de datos del sistema master  
USE master  
GO  
--02 Crear la base de datos de usuario Mercado  
CREATE DATABASE Mercado  
GO  
--03 Activar la base de datos de usuario Mercado  
USE Mercado  
GO  
  
--04 Crear el esquema Produccion  
CREATE SCHEMA Produccion  
GO  
--05 Crear la tabla Producto en el esquema Produccion  
CREATE TABLE Produccion.Producto(  
    IDProducto      CHAR(5) NOT NULL,  
    Nombre          VARCHAR(30) NOT NULL,  
    Precio          DECIMAL(8,2) NOT NULL  
)  
GO  
--06 insertar una fila a la tabla producto  
INSERT INTO Produccion.Producto(IDProducto,Nombre,Precio)  
VALUES('P0001','Monitor LCD',400)  
GO
```

## Modelo de seguridad de SQL Server

En el modelo de seguridad de SQL Server existen entidades de seguridad en tres niveles:

### Entidades de seguridad a nivel de Windows

- Inicio de sesión del dominio de Windows
- Inicio de sesión de grupo de Windows.

- Inicio de sesión local de Windows

### **Entidad de seguridad de SQL Server**

- Inicio de sesión de SQL Server
- Funciones del servidor.

### **Entidades de seguridad a nivel de bases de datos**

- Usuario de base de datos
- Función de base de datos
- Función de aplicación

## **Modo de autenticación**

Es la forma en que SQL Server procesa los nombres de usuario y las contraseñas. SQL Server ofrece dos modos de autenticación:

### **Modo de autenticación de Windows (Autenticación de Windows)**

El modo de autenticación de Windows permite a un usuario conectarse mediante una cuenta de usuario de Windows® Server 2012.

Permite a los usuarios de Windows local o de un dominio acceder al servidor sin necesitar un identificador y contraseñas separadas de SQL Server.

Ventajas:

- Evita recordar más contraseñas.
- La administración de la seguridad se simplifica.
- Por defecto la cuenta del usuario del administrador local y la cuenta del grupo Administradores de Windows están configurados con permisos totales a SQL Server.

### Modo mixto (autenticación de Windows y autenticación de SQL Server)

El modo mixto permite a los usuarios conectarse a una instancia de SQL Server utilizando identificadores de SQL Server o cuentas de Windows. Si un usuario tiene un identificador valido de SQL Server se acepta la conexión.

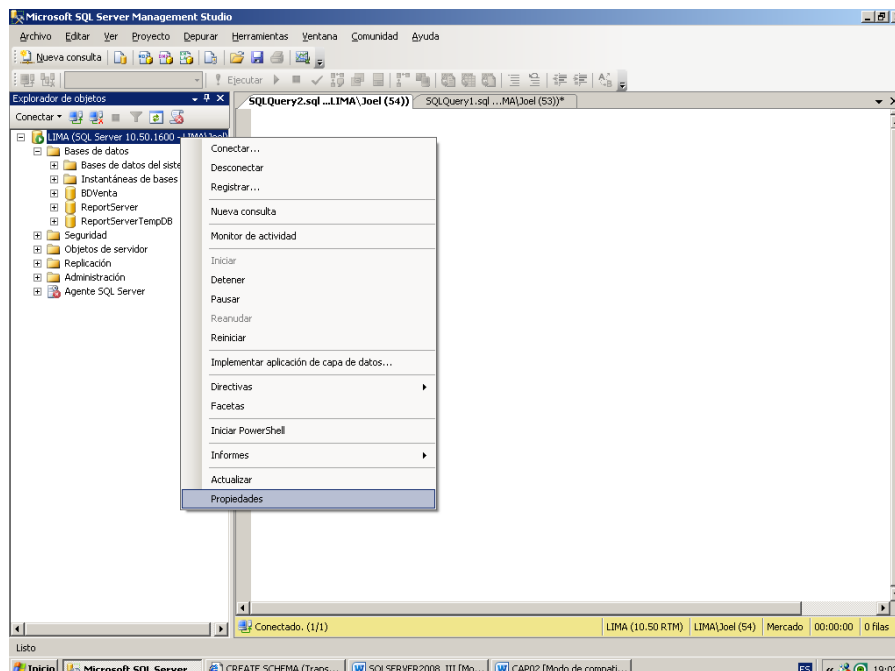
Si no tiene un identificador valido de SQL Server verifica la información de la cuenta de Windows si tiene permiso para conectarse acepta la conexión.

### Configuración del modo de autenticación

El modo de autenticación se configura al instalar SQL Server, sin embargo se puede cambiar cuando se efectúa un cambio en la red, agregar usuarios que no tengan una cuenta de Windows se debe cambiar al modo de autenticación mixto.

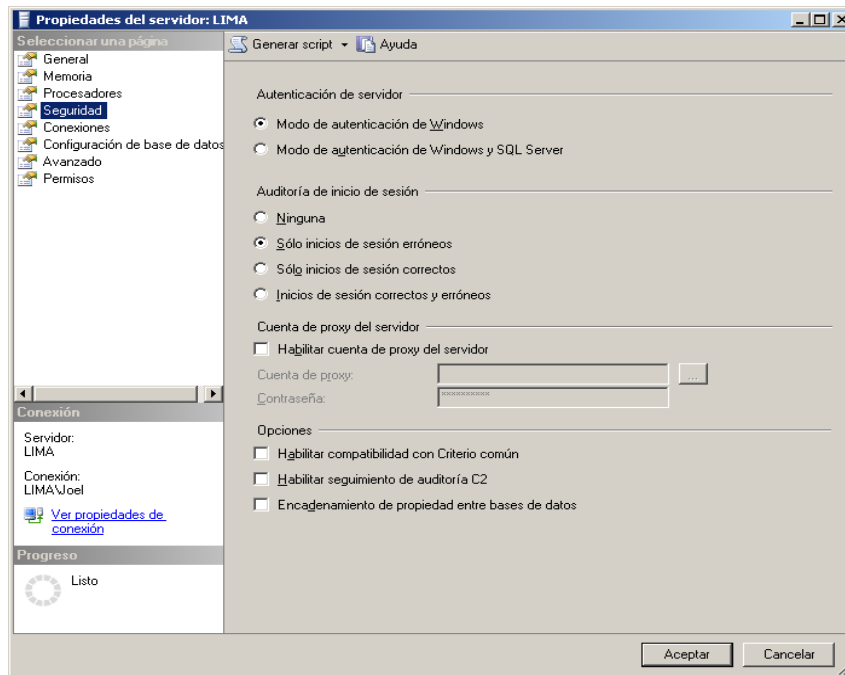
### Para cambiar el modo de autenticación

En la ventana Explorador de objetos dar un clic con el botón secundario, del menú contextual seleccionar la opción propiedades tal como se muestra en el siguiente gráfico:



En la ventana Propiedades del servidor seleccionar la página Seguridad. A continuación seleccionar el botón de opción Modo de autenticación de Windows y SQL Server.

Finalmente dar un clic en le botón Aceptar. Debe reiniciar el servicio SQL Server para poder utilizar esta nueva configuración.



## Inicio de sesión

Los inicios de sesión proporcionan a los usuarios acceso a SQL Server pero no a los recursos.

### Funciones de nivel de servidor

Para administrar con facilidad los permisos en el servidor, SQL Server proporciona varias funciones, que son las entidades de seguridad que agrupan a otras entidades de seguridad. Las funciones son como los grupos del sistema operativo Microsoft Windows.

Las funciones de nivel de servidor también se denominan funciones fijas de servidor porque no se pueden crear nuevas funciones de nivel de servidor. Las funciones de nivel de servidor se aplican a todo el servidor en lo que respecta a su ámbito de permisos.

Puede agregar inicios de sesión de SQL Server, cuentas de Windows y grupos de Windows a las funciones de nivel de servidor. Cada miembro de una función fija de servidor puede agregar otros inicios de sesión a esa misma función.

En la tabla siguiente se muestran las funciones de nivel de servidor y sus capacidades.

Nombre de la función de nivel de servidor	Descripción
sysadmin	Los miembros de la función fija de servidor sysadmin pueden realizar cualquier actividad en el servidor.
serveradmin	Los miembros de la función fija de servidor serveradmin pueden cambiar las opciones de configuración en el servidor y cerrar el servidor.
securityadmin	Los miembros de la función fija de servidor securityadmin administran los inicios de sesión y sus propiedades. Administran los permisos de servidor GRANT, DENY y REVOKE. También administran los permisos de base de datos GRANT, DENY y REVOKE. Asimismo, pueden restablecer las contraseñas para los inicios de sesión de SQL Server.
processadmin	Los miembros de la función fija de servidor processadmin pueden finalizar los procesos que se ejecutan en una instancia de SQL Server.
setupadmin	Los miembros de la función fija de servidor setupadmin pueden agregar y quitar los servidores vinculados.
bulkadmin	Los miembros de la función fija de servidor bulkadmin pueden ejecutar la instrucción BULK INSERT.
diskadmin	La función fija de servidor diskadmin se utiliza para administrar archivos de disco.
dbcreator	Los miembros de la función fija de servidor dbcreator pueden crear, modificar, quitar y restaurar cualquier base de datos.
public	Cada inicio de sesión de SQL Server pertenece a la función pública de servidor. Cuando a una entidad de seguridad de servidor no se le han concedido ni denegado permisos específicos para un objeto protegible, el usuario hereda los permisos concedidos a la función pública para ese elemento. Solo asigne los permisos públicos en cualquier objeto cuando desee que el objeto esté disponible para todos los usuarios.

## Trabajar con funciones de nivel de servidor

En la tabla siguiente se explican los comandos, las vistas y las funciones que se utilizan para trabajar con funciones de nivel de servidor.

Característica	Tipo	Descripción
sp_helpsrvrole (Transact-SQL)	Metadatos	Devuelve una lista de funciones de nivel de servidor.
sp_helpsrvrolemember (Transact-SQL)	Metadatos	Devuelve información acerca de los miembros de una función de nivel de servidor.
sp_srvrolepermission (Transact-SQL)	Metadatos	Muestra los permisos de una función de nivel de servidor.
IS_SRVROLEMEMBER (Transact-SQL)	Metadatos	Indica si un inicio de sesión de SQL Server es miembro de la función de nivel de servidor especificada.
sys.server_role_members (Transact-SQL)	Metadatos	Devuelve una fila por cada miembro de cada función de nivel de servidor.
sp_addsrvrolemember (Transact-SQL)	Comando	Agrega un inicio de sesión como miembro de una función de nivel de servidor.
sp_dropsrvrolemember (Transact-SQL)	Comando	Quita un inicio de sesión de SQL Server o un usuario o grupo de Windows de una función de nivel de servidor.

## Funciones en el nivel de base de datos

Para administrar con facilidad los permisos en las bases de datos, SQL Server proporciona varias funciones, que son las entidades de seguridad que agrupan a otras entidades de seguridad. Son como los grupos del sistema operativo Microsoft Windows. Las funciones del nivel de base de datos se aplican a toda la base de datos en lo que respecta a su ámbito de permisos.

Existen dos tipos de funciones de nivel de base de datos en SQL Server: las funciones fijas de base de datos, que están predefinidas en la base de datos, y las funciones flexibles de base de datos, que pueden crearse.

Las funciones de base de datos fijas se definen en el nivel de base de datos y existen en cada una de ellas. Los miembros de las funciones de base de datos db\_owner y db\_securityadmin pueden administrar los miembros de las funciones de base de datos fijas. Sin embargo, sólo los miembros de la función de base de datos db\_owner pueden agregar miembros a la función de base de datos fija db\_owner. Hay también algunas funciones fijas de base de datos con fines especiales en la base de datos msdb.

En la tabla siguiente se muestran las funciones fijas del nivel de base de datos y sus capacidades. Estas funciones existen en todas las bases de datos.

Nombre de función de nivel de base de datos	Descripción
db_owner	Los miembros de la función de base de datos fija db_owner pueden realizar todas las actividades de configuración y mantenimiento en la base de datos y también pueden quitar la base de datos.
db_securityadmin	Los miembros de la función de base de datos fija db_securityadmin pueden modificar la pertenencia a funciones y administrar permisos. Si se agregan entidades de seguridad a esta función, podría habilitarse un aumento de privilegios no deseado.
db_accessadmin	Los miembros de la función de base de datos fija db_accessadmin pueden agregar o quitar el acceso a la base de datos para inicios de sesión de Windows, grupos de Windows e inicios de sesión de SQL Server.
db_backupoperator	Los miembros de la función de base de datos fija db_backupoperator pueden crear copias de seguridad de la base de datos.
db_ddladmin	Los miembros de la función de base de datos fija db_ddladmin pueden ejecutar cualquier comando del lenguaje de definición de datos (DDL) en una base de datos.
db_datawriter	Los miembros de la función de base de datos fija db_datawriter pueden agregar, eliminar o cambiar datos en



	todas las tablas de usuario.
db_datareader	Los miembros de la función de base de datos fija db_datareader pueden leer todos los datos de todas las tablas de usuario.
db_denydatawriter	Los miembros de la función de base de datos fija db_denydatawriter no pueden agregar, modificar ni eliminar datos de tablas de usuario de una base de datos.
db_denydatareader	Los miembros de la función de base de datos fija db_denydatareader no pueden leer datos de las tablas de usuario dentro de una base de datos.

### **Función pública de la base de datos**

Todos los usuarios de una base de datos pertenecen a la función pública de la base de datos. Cuando a un usuario no se le han concedido ni denegado permisos específicos para un objeto que puede protegerse, el usuario hereda los permisos concedidos a la función pública para ese objeto.

### **Ejemplos de creación de un Inicio de sesión**

--01 Crear el inicio de sesión mtorres, mtorres pertenece al dominio CEPS01

```
USE [master]
```

```
GO
```

```
CREATE LOGIN [CEPS01\mtorres] FROM WINDOWS
```

```
WITH
```

```
DEFAULT_DATABASE=[Northwind]
```

```
GO
```

--02 Crear el usuario de base de datos de Northwind

```
USE [Northwind]
```

```
GO
```

```
CREATE USER [mtorres] FOR LOGIN [CEPS01\mtorres]
```

```
GO
```

--03 Otorgar el permiso SELECT al usuario mtorres

use [Northwind]

GO

GRANT SELECT ON [dbo].[Employees] TO [mtorres]

GO

--04 Crear un inicio de sesión dlopez para un usuario que NO tiene una cuenta en el dominio.

USE [master]

GO

CREATE LOGIN [dlopez] WITH

PASSWORD=N'ceps'

, DEFAULT\_DATABASE=[Northwind]

, CHECK\_EXPIRATION=OFF

, CHECK\_POLICY=OFF

GO

--05 Crear el usuario de base de datos de Northwind

USE [Northwind]

GO

CREATE USER [dlopez] FOR LOGIN [dlopez]

GO

--06 Otorgar el permiso SELECT al usuario dlopez

use [Northwind]

GO

GRANT SELECT ON [dbo].[Products] TO [dlopez]

GO

--07 Cuando el usuario tiene una cuenta en el dominio el encargado de autenticar es el SO WINDOWS

USE [master]

GO

CREATE LOGIN [CEPS01\psuarez] FROM WINDOWS

WITH

DEFAULT\_DATABASE=[Northwind]

GO

--08 Cuando el usuario no tiene una cuenta en el dominio el encargado de autenticar es el SQL Server. El modo de autenticación debe ser: Windows y SQL Server

USE [master]

GO

CREATE LOGIN [ana]

WITH

PASSWORD=N'ceps'

, DEFAULT\_DATABASE=[Northwind]

, CHECK\_EXPIRATION=OFF

, CHECK\_POLICY=OFF

GO

--09 Ejemplo de funciones fijas del servidor. rmarcelo debe pertenecer a la función fija del servidor sysadmin.

EXEC master..sp\_addsrvrolemember

@loginame = N'CEPS01\rmarcelo'

, @rolename = N'sysadmin'

GO

--10 El usuario jacarrasco1 debe pertenecer a la función fija de datos db\_ddladmin

USE [Northwind]

GO

EXEC sp\_addrolemember

```
N'db_ddladmin'
, N'jcarrasco1'
GO
--11 Activar la base de datos master
USE [master]
GO
--12 Crear los inicios de sesión
--12.1 Crear el inicio de sesion [CEPS01\dzegarra]
CREATE LOGIN [CEPS01\dzegarra] FROM WINDOWS
WITH DEFAULT_DATABASE=[Northwind]
GO
--12.2 Crear el inicio de sesion [CEPS01\jguerra]
CREATE LOGIN [CEPS01\jguerra] FROM WINDOWS
WITH DEFAULT_DATABASE=[Northwind]
GO
--12.3 Crear el inicio de sesion [CEPS01\taldoradin]
CREATE LOGIN [CEPS01\taldoradin] FROM WINDOWS
WITH DEFAULT_DATABASE=[Northwind]
GO
--13 Crear los usuarios de base de datos
--13.1
USE [Northwind]
GO
CREATE USER [dzegarra] FOR LOGIN [CEPS01\dzegarra]
GO
--13.2
USE [Northwind]
GO
```

```
CREATE USER [jguerra] FOR LOGIN [CEPS01\jguerra]
```

```
GO
```

```
--13.3
```

```
USE [Northwind]
```

```
GO
```

```
CREATE USER [taldoradin] FOR LOGIN [CEPS01\taldoradin]
```

```
GO
```

```
--14 El usuario dzegarra debe pertenecer a la función fija de servidor sysadmin.
```

```
EXEC master..sp_addsrvrolemember
```

```
@loginame = N'CEPS01\dzegarra'
```

```
, @rolename = N'sysadmin'
```

```
GO
```

```
--15 jguerra debe ser el propietario de la base de datos Northwind, debe pertenecer a la  
función fija de base de datos db_owner
```

```
USE [Northwind]
```

```
GO
```

```
EXEC sp_addrolemember
```

```
  N'db_owner'
```

```
, N'jguerra'
```

```
GO
```

```
--16 Crear la función de base de datos personalizada SEINUP
```

```
--16.1 Activar la base de datos del sistema master
```

```
USE master;
```

```
GO
```

```
--16.2 Crear la base de datos de usuario DELMONT
```

```
CREATE DATABASE DELMONT ;
```

```
GO
```

```
--16.3 Activar la base de datos de usuario DELMONT;
```

```
USE DELMONT;
```

GO

--16.4 Crear la función personalizada SEINUP

USE [DELMONT]

GO

CREATE ROLE [SEINUP]

GO

--16.5 Otorgar permisos Seleccionar, Actualizar Insertar a la función [SEINUP]

use [DELMONT]

GO

GRANT UPDATE TO [SEINUP]

GO

use [DELMONT]

GO

GRANT INSERT TO [SEINUP]

GO

use [DELMONT]

GO

GRANT SELECT TO [SEINUP]

GO

--16.6 Agregar usuarios a la función [SEINUP]

USE [DELMONT]

GO

EXEC sp\_addrolemember N'SEINUP', N'ana'

GO

--17 Ejemplo de función de aplicación

--17.1 Crear el inicio de sesión [AplicacionWEB]

USE [master]

GO

CREATE LOGIN [AplicacionWEB]

WITH

PASSWORD=N'ceps'

, DEFAULT\_DATABASE=[Northwind]

, CHECK\_EXPIRATION=OFF

, CHECK\_POLICY=OFF

GO

--17.2 Crear el usuario de base de datos

USE [Northwind]

GO

CREATE USER [AplicacionWEB] FOR LOGIN [AplicacionWEB]

GO

--17.3 Crear la función de aplicación [PruebaWEB]

USE [Northwind]

GO

CREATE APPLICATION ROLE [PruebaWEB] WITH PASSWORD = N'P@ssw0rd'

GO

--17.4 Otorgar permisos SELECT, en la tabla Products a la función de aplicación [PruebaWEB]

use [Northwind]

GRANT SELECT ON [dbo].[Products] TO [PruebaWEB]

--17.5 Desconectar el servidor

--17.6 Conectar como [AplicacionWEB], contraseña ceps.

--17.8 Activar la función de aplicación 'PruebaWeb'

EXECUTE sp\_setapprole 'PruebaWeb', 'P@ssw0rd';

GO

--17.9 Consultar la tabla Products.

SELECT \* FROM Products

# Capítulo 04

## Copia de seguridad de base de datos

### Introducción

El componente para la realización de copias de seguridad y restauración de Microsoft® SQL Server™ 2008 proporciona una importante protección para los datos decisivos almacenados en bases de datos de SQL Server. Con una planificación adecuada, se pueden recuperar muchos errores, incluidos: Errores de medios, errores de usuarios y pérdida permanente de un servidor.

Además, la realización de copias de seguridad y la restauración de bases de datos resulta útil para otros fines, como copiar una base de datos de un servidor a otro. Se puede hacer una copia de una base de datos rápida y fácilmente realizando una copia de seguridad de la base de datos de un equipo y restaurándolo en otro.

### Diseñar una estrategia de copia de seguridad y restauración

Debe identificar los requisitos de disponibilidad de sus datos para poder elegir la estrategia de copia de seguridad y restauración apropiada. La estrategia general de copia de seguridad define el tipo y frecuencia de copias de seguridad y la naturaleza y velocidad del hardware que requieren.

Se recomienda que pruebe cuidadosamente los procedimientos de copia de seguridad y recuperación. Las pruebas permiten asegurarse de que dispone de las copias de seguridad necesarias para recuperar diferentes errores y que los procedimientos se pueden ejecutar rápidamente y sin problemas cuando se produce un error real.

### Analizar los requisitos de disponibilidad y recuperación

Para desarrollar un plan de copia de seguridad y restauración correcto, debe comprender cuándo los datos deben ser accesibles y el posible impacto de la pérdida de datos en su



negocio. Responda a las siguientes preguntas para determinar sus requisitos de disponibilidad de datos y cómo le afectaría la pérdida de datos. A continuación, podrá elegir el modelo correcto de recuperación de Microsoft® SQL Server™ 2008 para las bases de datos y realizar las operaciones técnicas y financieras necesarias.

Estas son algunas preguntas básicas que le ayudarán a analizar sus requisitos de disponibilidad y recuperación:

¿Cuáles son sus requisitos de disponibilidad? ¿Qué parte de cada día debe estar en línea la base de datos?

¿Qué costo financiero supone para su negocio el tiempo de inactividad?

En caso de error de uno de los medios, por ejemplo, una unidad de disco, ¿cuál sería el tiempo de inactividad aceptable?

En caso de desastre, como la pérdida de un servidor en un incendio, ¿cuál sería el tiempo de inactividad aceptable?

¿Qué importancia tiene no perder nunca cambios?

¿Con qué facilidad podrían volverse a crear los datos perdidos?

¿Su organización dispone de administradores de sistema o de bases de datos?

¿Quién será el responsable de realizar las operaciones de copia de seguridad y recuperación y cómo se les formará?

Estas son algunas preguntas que le ayudarán a elegir las herramientas, técnicas y hardware apropiados para el sitio.

¿Qué tamaño tiene cada base de datos?

¿Con qué frecuencia se modifican los datos de cada base de datos?

¿Se modifican algunas tablas más a menudo que otras?

¿Cuáles son los periodos de producción de la base de datos más importantes?

¿Cuándo se utiliza mucho la base de datos y se obtienen como resultado inserciones y actualizaciones frecuentes?

¿Podría suponer un problema que el registro de transacciones consuma espacio debido a una gran actividad de actualización?

¿La base de datos se somete periódicamente a cargas masivas de datos?

¿La base de datos está sujeta a errores de aplicación o actualizaciones arriesgadas que podrían no detectarse inmediatamente?

¿El servidor de base de datos forma parte de un clúster de conmutación por error de SQL Server 2008 para aumentar la disponibilidad?

¿La base de datos está en un entorno multiservidor con administración centralizada?

## **Administrar medios**

Para realizar una copia de seguridad y restauración de una base de datos, es necesario realizar la copia de seguridad en un medio, por ejemplo, cintas o discos. Se recomienda que el plan de copias de seguridad incluya provisiones para administrar los medios, por ejemplo:

Un plan de seguimiento y administración para almacenar y reciclar conjuntos de copias de seguridad.

Una programación para sobrescribir el medio de copia de seguridad.

En un entorno multiservidor, la decisión de utilizar copias de seguridad centralizadas o distribuidas.

## **Planear la recuperación de un desastre**

Es necesario crear un plan de recuperación de desastres para garantizar que todos los sistemas y datos puedan recuperar rápidamente su funcionamiento normal en caso de un desastre natural (por ejemplo, un incendio) o un desastre técnico (por ejemplo, errores en dos discos de una matriz RAID-5). Para crear un plan de recuperación de desastres, es necesario preparar todas las acciones que deben ocurrir en respuesta a un suceso catastrófico. Se recomienda que compruebe el plan de recuperación de desastres mediante la simulación de un suceso catastrófico. Es necesario realizar el diseño del plan de recuperación de desastres en su propio entorno y según las necesidades de su negocio. Por ejemplo, supongamos que se produce un incendio y destruye el centro de datos disponibles 24 horas al día. ¿Está seguro de que es posible la recuperación? ¿Cuánto tiempo se puede tardar en llevar a cabo la recuperación y tener disponible el sistema? ¿Cuál es la cantidad de datos perdidos que pueden tolerar los usuarios?

Lo ideal es que el plan de recuperación de desastres indique el tiempo que durará la recuperación y el estado final de las bases de datos que los usuarios pueden esperar. Por

ejemplo, puede determinar que, tras la adquisición del hardware especificado, la recuperación debe completarse en 48 horas y sólo se garantizarán los datos hasta finales de la semana anterior.

Un plan de recuperación de desastres se puede estructurar de diferentes maneras y puede contener muchos tipos de información, entre otra:

Un plan para adquirir el hardware.

Un plan de comunicación.

Una lista de las personas con las que ponerse en contacto en caso de desastre.

Instrucciones para ponerse en contacto con las personas implicadas en la respuesta al desastre.

Información acerca del propietario de la administración del plan.

## **Ejecutar una secuencia de comandos con la funcionalidad básica**

Normalmente, las secuencias de comandos con la funcionalidad básica se incluyen en los planes de recuperación de desastres para confirmar que todo funciona como se espera. La secuencia de comandos con la funcionalidad básica proporciona una herramienta segura para que los administradores de sistemas o de bases de datos puedan comprobar que se ha recuperado la base de datos con un estado viable, sin tener que depender de los usuarios finales para llevar a cabo la comprobación. Normalmente, se trata de un archivo .SQL con instrucciones SQL por lotes que se ejecutan en el servidor desde osql. Para otras aplicaciones, resulta más apropiado un archivo .BAT, ya que puede contener comandos de bcp y osql. Esta secuencia de comandos con la funcionalidad básica es muy específica de la aplicación y puede adoptar muchas formas diferentes. Por ejemplo, en un sistema de notificación o ayuda a la toma de decisiones, la secuencia de comandos puede ser simplemente una copia de varias de las consultas para informes más importantes. Para una aplicación de procesamiento de transacciones en línea (OLTP), la secuencia de comandos puede ejecutar un lote de procedimientos almacenados que contengan instrucciones INSERT, UPDATE y DELETE.

## Seleccionar un modelo de recuperación

Microsoft® SQL Server™ dispone de tres modelos de recuperación que son Simple, Completa y Registro masivo utilizados para:

- Simplificar el diseño del plan de recuperación.
- Simplificar los procedimientos de copia de seguridad y recuperación.
- Aclarar las contrapartidas de los requisitos operativos del sistema.

Cada uno de estos modelos está destinado a diferentes necesidades de rendimiento, espacio en disco y cinta, y protección contra pérdida de datos. Por ejemplo, al elegir un modelo de recuperación, debe tener en cuenta las contrapartidas entre los siguientes requisitos del negocio:

- Rendimiento de operaciones a gran escala, por ejemplo, creación de índices o cargas masivas.
- Riesgo de perder de datos, por ejemplo, la pérdida de transacciones confirmadas.
- Espacio que consume el registro de transacciones.
- Simplicidad de los procedimientos de copia de seguridad y recuperación.

Esta tabla ofrece información general acerca de las ventajas e implicaciones de los tres modelos de recuperación.

Modelo de recuperación	Ventajas	Riesgo de perder trabajo	¿Recuperación hasta un momento dado?
Simple	Permite operaciones de copia masiva de gran rendimiento.  Reduce el tamaño del registro para mantener bajos los requisitos de espacio.	Se pueden repetir los cambios desde la última copia de seguridad diferencial o de base de datos.	Se puede recuperar hasta el final de cualquier copia de seguridad. A continuación, se pueden repetir los cambios.

Completa	No se pierde trabajo si un archivo de datos se pierde o se daña.  Se puede recuperar hasta cualquier momento, por ejemplo, antes del error de aplicación o usuario.	Normalmente, ninguno.  Si el registro se daña, se pueden repetir los cambios desde la última copia de seguridad del registro.	Se puede recuperar hasta cualquier momento.
Registro masivo	Permite operaciones de copia masiva de gran rendimiento.  Las operaciones masivas utilizan un espacio de registro mínimo.	Si el registro se daña o se han realizado operaciones masivas desde la última copia de seguridad del registro, se pueden repetir los cambios desde esa última copia de seguridad.  En caso contrario, no se pierde el trabajo.	Se puede recuperar hasta el final de cualquier copia de seguridad. A continuación, se pueden repetir los cambios.

Cuando se crea una base de datos, tiene el mismo modelo de recuperación que la base de datos model. Para modificar el modelo de recuperación predeterminado, utilice ALTER DATABASE para modificar el modelo de recuperación de la base de datos model. El modelo de recuperación se establece con la cláusula RECOVERY de la instrucción ALTER DATABASE.

### **Recuperación simple**

La recuperación simple es la que menos administración requiere. En el modelo de recuperación simple, sólo se pueden recuperar los datos de la copia de seguridad completa o diferencial más reciente. No se utilizan copias de seguridad del registro de transacciones y el registro de transacciones ocupa un espacio mínimo. Cuando el espacio del registro ya no se necesita para recuperar un error del servidor, se vuelve a utilizar.

El modelo de recuperación simple es más sencillo de administrar que los modelos de recuperación completa o de registro masivo, pero el riesgo de perder datos es mayor si el archivo de datos resulta dañado.

Importante La recuperación simple no es apropiada para sistemas de producción en los que la pérdida de datos es inadmisibles.

Cuando se utiliza la recuperación simple, el intervalo de tiempo entre las copias de seguridad debe ser suficientemente grande para evitar que la sobrecarga por las copias de seguridad afecte al trabajo de producción; al mismo tiempo, debe ser suficientemente pequeño como para evitar la pérdida de una cantidad considerable de datos.

### **Recuperación completa y de registro masivo**

Los modelos de recuperación completa y de registro masivo ofrecen la mayor protección de los datos. Estos modelos utilizan el registro de transacciones para ofrecer una recuperación completa y para evitar la pérdida de trabajo en una amplia variedad de casos de error. El modelo de recuperación completa ofrece la máxima flexibilidad para recuperar la base de datos hasta un momento anterior. El modelo de registro masivo ofrece más rendimiento y menos consumo de espacio de registro para determinadas operaciones a gran escala, como la creación de índices o copias masivas. A cambio, es menos flexible en el momento hasta el que recupera.

Debido a que muchas bases de datos pasan por períodos de carga masiva o creación de índices, es posible que desee alternar entre los modelos de recuperación completa o de registro masivo.

### **Arquitectura de la copia de seguridad o restauración**

Los componentes de copia de seguridad y restauración de Microsoft® SQL Server™ 2008 le permiten crear una copia de una base de datos. Esta copia se almacena en una ubicación protegida ante posibles fallos del servidor que ejecuta la instancia de SQL Server. Si fallara el servidor que ejecuta la instancia de SQL Server o si la base de datos se daña, la copia de seguridad se puede utilizar para volver a crear, o restaurar, la base de datos.

SQL Server 2008 proporciona capacidades de copia de seguridad y restauración sofisticadas:

**Opciones de copia de seguridad y restauración de una base de datos:**

1. Una copia de seguridad completa de la base de datos es una copia completa de la base de datos.
2. Una copia de seguridad del registro de transacciones sólo copia el registro de transacciones.
3. Una copia de seguridad diferencial sólo copia las páginas de la base de datos modificadas después de la última copia de seguridad completa de la base de datos.

Una restauración de un archivo o un grupo de archivos permite recuperar sólo la parte de la base de datos que estaba en el disco dañado.

Estas opciones permiten personalizar los procesos de copia de seguridad y restauración de acuerdo con la importancia de los datos de la base de datos. Las bases de datos menos importantes que se pueden volver a crear fácilmente a partir de otro origen pueden no tener copias de seguridad, otras bases de datos pueden tener copias de seguridad sencillas que vuelven a crear la base de datos en el estado de la noche anterior, y las bases de datos importantes pueden tener copias de seguridad sofisticadas que restauran la base de datos justo hasta el momento del fallo.

**Control con las instrucciones BACKUP y RESTORE.**

Los usuarios pueden ejecutar las instrucciones BACKUP y RESTORE directamente desde aplicaciones, secuencias de comandos Transact-SQL, procedimientos almacenados y desencadenadores.

Los procesos de copia de seguridad y restauración se deben planear conjuntamente. Primero, los administradores tienen que determinar la importancia de los datos de la base de datos. Tienen que determinar si es aceptable sólo restaurar la base de datos hasta el punto en el que se encontraba la noche anterior al fallo o si la base de datos tiene que

ser restaurada hasta el punto lo más próximo posible a la hora del fallo. También tienen que determinar durante cuánto tiempo puede no estar disponible la base de datos, si tiene que ser puesta en funcionamiento lo más rápidamente posible o si no es necesario restaurarla inmediatamente.

Después de determinar los requisitos de la restauración, los administradores pueden diseñar el proceso de copia de seguridad que mantenga el conjunto de copias de seguridad que cumpla los requisitos. Los administradores pueden elegir los procesos de copia de seguridad que se pueden ejecutar con un impacto mínimo sobre el sistema, al mismo tiempo que se cumplen los requisitos para la restauración. Basándose en los requisitos de recursos, los administradores también eligen el modelo de recuperación de la base de datos.

## **El modelo de recuperación**

Mantiene el equilibrio entre la sobrecarga del registro y la importancia de la recuperación completa de los datos. Los modelos de recuperación son:

### **Completa**

Los datos son importantes y se deben recuperar hasta el momento del error. Se registran todas las modificaciones de datos. Están disponibles todas las opciones de recuperación de SQL Server 2008.

### **Registro masivo**

Si es necesario es posible reproducir algunas operaciones masivas (operaciones de copia masivas, SELECT INTO, procesamiento de texto), ya que estas operaciones no se han registrado completamente. Sólo es posible recuperar hasta el final de la última base de datos o registro de copia de seguridad.

### **Simple**

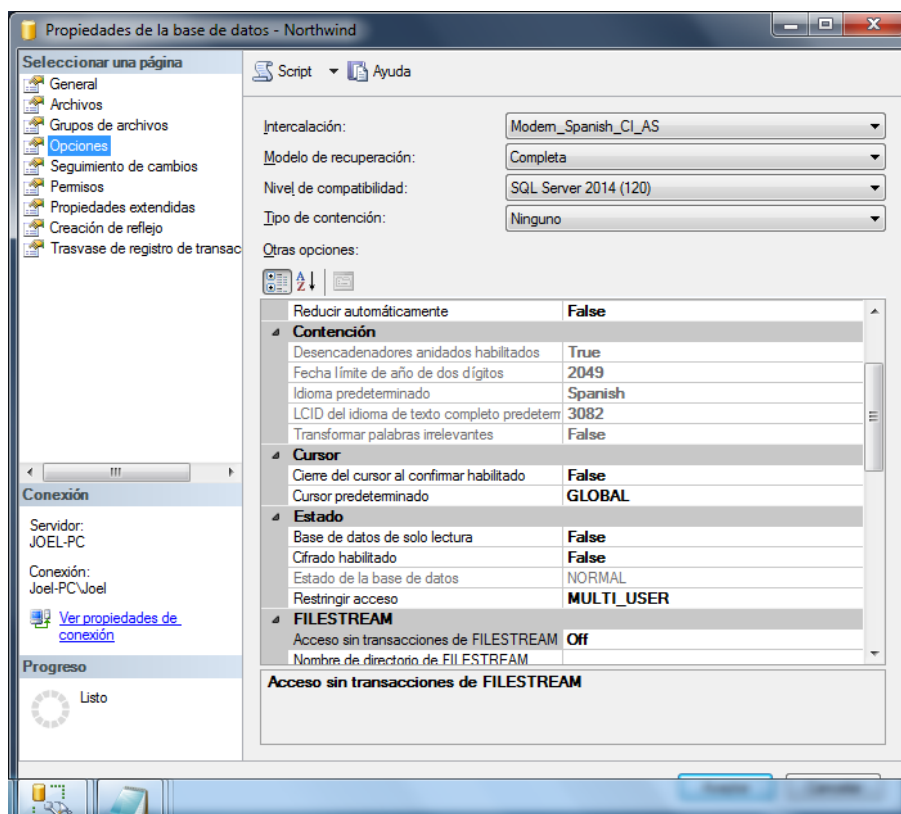
Todas las modificaciones de datos realizadas desde la última copia de seguridad son prescindibles, o es posible rehacerlas. Mínima sobrecarga del registro, pero no es posible recuperar más allá del final de la última copia de seguridad.



## Como cambiar el modelo de Recuperación de una base de datos

Para cambiar el modelo de recuperación de la base de datos Northwind, debe seguir los siguientes pasos: En el explorador de objetos seleccionar la base de datos Northwind clic derecho seleccionar del menú contextual Propiedades.

Seleccionar Opciones, Modelo de recuperación Completa.



## Dispositivos de copia de seguridad

Al crear copias de seguridad, debe seleccionar el dispositivo donde se creará la copia de seguridad de los datos. Microsoft® SQL Server™ realiza copias de seguridad de bases de datos, registros de transacciones y archivos en dispositivos de cinta y disco.

### Dispositivos de disco

Los dispositivos de disco para copias de seguridad son archivos ubicados en discos duros u otros medios de almacenamiento en disco y son como archivos normales del sistema operativo. La referencia a un dispositivo de disco para copias de seguridad se

realiza exactamente igual que la referencia a cualquier otro archivo del sistema operativo. Los dispositivos para copias de seguridad en disco se pueden definir en el disco local de un servidor o en un disco remoto de un recurso de red compartido, y pueden ser tan grandes o tan pequeños como sea necesario. El tamaño máximo de un archivo viene dado por el espacio libre disponible en el disco.

Si la copia de seguridad se va a realizar a través de la red en un disco de un equipo remoto, utilice el nombre UNC (Convención de nomenclatura universal) en la forma `\\nombreServidor\nombreCompartido\ruta\archivo` para especificar la ubicación del archivo. De la misma forma que al escribir archivos en el disco duro local, se debe conceder a la cuenta de usuario que utiliza SQL Server los permisos necesarios para leer o escribir archivos en el disco remoto.

Debido a que la realización de copias de seguridad de datos a través de una red está expuesta a errores, una vez completada, compruebe la operación de copia de seguridad..

Importante No se recomienda realizar una copia de seguridad en un archivo situado en el mismo disco que la base de datos. Si se produce un error en el dispositivo de disco que contiene la base de datos, no habrá manera de recuperar la base de datos porque la copia de seguridad está ubicada en el mismo disco.

### **Dispositivos de cinta**

Los dispositivos para copias de seguridad en cinta se utilizan exactamente de la misma manera que los dispositivos de disco, con las excepciones siguientes:

El dispositivo de cinta debe estar conectado físicamente al equipo donde se ejecuta una instancia de SQL Server.

No se admite la creación de copias de seguridad en dispositivos de cinta remotos.

Si un dispositivo para copias de seguridad en cinta se llena durante la operación de copia de seguridad, pero todavía falta por escribir la mayor parte de los datos, SQL Server le pedirá una nueva cinta y continuará con la operación de copia de seguridad.

## Dispositivos físicos y lógicos

SQL Server identifica dispositivos de copia de seguridad tanto con un nombre de dispositivo físico como lógico.

El nombre del dispositivo de copia de seguridad físico es el que utiliza el sistema operativo para identificar el dispositivo de copia de seguridad, por ejemplo, C:\BACKUPS\ACCOUNTING\FULL.BAK.

El dispositivo de copia de seguridad lógico es un alias, o nombre común, que se utiliza para identificar el dispositivo de copia de seguridad físico. El nombre del dispositivo lógico se almacena de manera permanente en las tablas del sistema de SQL Server. La ventaja de utilizar un dispositivo de copia de seguridad lógico es que se puede hacer referencia a él de forma más sencilla que a un nombre de dispositivo físico. Por ejemplo, un nombre de dispositivo lógico podría ser Accounting\_Backup, pero el dispositivo físico sería C:\BACKUPS\ACCOUNTING\FULL.BAK.

Cuando realice una copia de seguridad de una base de datos o la restaure, puede intercambiar el uso de nombres de dispositivo de copia de seguridad físicos y lógicos.

Por ejemplo, ejecute la instrucción BACKUP con el nombre de dispositivo lógico o físico:

-- Specify the logical backup device.

BACKUP DATABASE accounting

TO Accounting\_Backup

-- Or, specify the physical backup device.

BACKUP DATABASE accounting

TO DISK = 'C:\Backups\Accounting\Full.Bak'

## Copias de seguridad diferenciales de bases de datos

La copia de seguridad diferencial registra sólo los datos que han cambiado después de la última copia de seguridad de la base de datos. Puede realizar copias de seguridad más

frecuentes porque las copias de seguridad diferenciales son más pequeñas y más rápidas que las copias de seguridad de la base de datos. Realizar copias de seguridad frecuentes disminuye el riesgo de perder datos.

Nota Si ha creado copias de seguridad de archivos desde la última de copia de seguridad completa de base de datos, Microsoft® SQL Server™ 2008 examinará esos archivos al comenzar una copia de seguridad diferencial de base de datos. Esto puede afectar al rendimiento de la copia de seguridad diferencial de base de datos.

Las copias de seguridad diferenciales de una base de datos se utilizan para restaurar la base de datos hasta el momento en que se terminó la copia de seguridad diferencial. Para recuperar hasta el momento exacto del error, debe utilizar las copias de seguridad del registro de transacciones..

Utilice copias de seguridad diferenciales de la base de datos cuando:

1. Sólo ha cambiado una parte relativamente pequeña de los datos de la base de datos desde la última copia de seguridad de la base de datos. Las copias de seguridad diferenciales de base de datos son especialmente efectivas si se modifican los mismos datos muchas veces.
2. Esté utilizando el modelo de recuperación simple y desee realizar copias de seguridad más frecuentes, pero no completas.
3. Utilice el modelo de recuperación completa o de registro masivo y desee minimizar el tiempo que lleva la confirmación de las copias de seguridad del registro de transacciones al restaurar una base de datos.

El siguiente es un proceso recomendado para implementar copias de seguridad diferenciales de la base de datos:

Cree copias de seguridad de la base de datos con regularidad.

Entre las copias de seguridad de la base de datos, cree periódicamente copias de seguridad diferencial, por ejemplo, cada cuatro horas o más a menudo en sistemas con mucha actividad.

Si utiliza la recuperación completa o de registro masivo, cree copias de seguridad del registro de transacciones con más frecuencia que las copias de seguridad diferenciales de base de datos, por ejemplo, cada 30 minutos.

La secuencia de la restauración de las copias de seguridad diferenciales de base de datos es:

1. Restaure la última copia de seguridad de la base de datos.
2. Restaure la última copia de seguridad diferencial de la base de datos.
3. Si utiliza la recuperación completa o de registro masivo, aplique todas las copias de seguridad del registro de transacciones creadas después de la última copia de seguridad diferencial de la base de datos.

**Importante** No se puede crear una copia de seguridad diferencial de una base de datos si no se ha realizado antes una copia de seguridad de la base de datos.

Ejecute la instrucción `BACKUP DATABASE` para crear la copia diferencial de la base de datos especificando:

El nombre de la base de datos de la que se va a realizar una copia de seguridad.

El dispositivo de copia de seguridad en el que se escribirá la copia de seguridad de la base de datos.

La cláusula `DIFFERENTIAL`, para especificar que sólo se realice una copia de seguridad de algunas partes de la base de datos que han cambiado desde la última copia de seguridad.

Opcionalmente, especifique:

La cláusula `INIT` para sobrescribir el medio de copia de seguridad y escribir la copia de seguridad como el primer archivo del medio de copia de seguridad. Si no existe un encabezado de medio, se escribe uno automáticamente.

Las cláusulas `SKIP` e `INIT` para sobrescribir el medio de copia de seguridad, incluso si hay alguna copia de seguridad que aún no haya caducado en el medio de copia de

seguridad o si el nombre del medio no coincide con el nombre del medio de la copia de seguridad.

La cláusula **FORMAT** cuando se utiliza el medio por primera vez para inicializar completamente el medio de copia de seguridad y sobrescribir cualquier encabezado de medio existente.

Importante Tenga mucha precaución al utilizar las cláusulas **FORMAT** o **INIT** de la instrucción **BACKUP**, ya que éstas destruirán cualquier copia de seguridad almacenada previamente en el medio de copia de seguridad.

### Ejemplos

En este ejemplo se crea una copia de seguridad completa y una copia de seguridad diferencial de la base de datos MyNwind.

-- Crear primero la copia completa de la base de datos.

```
BACKUP DATABASE MyNwind
```

```
    TO MyNwind_1
```

```
    WITH INIT
```

```
GO
```

```
BACKUP DATABASE MyNwind
```

```
    TO MyNwind_1
```

```
    WITH DIFFERENTIAL
```

```
GO
```

### Copias de seguridad de registros de transacciones

El registro de transacciones es un registro en serie de todas las transacciones que se han realizado en la base de datos desde que se realizó la última copia de seguridad del registro de transacciones. Con las copias de seguridad del registro de transacciones, puede recuperar la base de datos hasta un momento determinado (por ejemplo, antes de escribir datos no deseados) o hasta el momento del error.

Al restaurar una copia de seguridad del registro de transacciones, Microsoft® SQL Server™ confirma todos los cambios registrados en el registro de transacciones. Cuando

SQL Server llega al final del registro de transacciones, ha creado de nuevo el estado exacto de la base de datos en el momento en el que se inició la operación de copia de seguridad. Si se recupera la base de datos, SQL Server deshace todas las transacciones que estaban incompletas cuando se inició la operación de copia de seguridad.

Normalmente, las copias de seguridad del registro de transacciones utilizan menos recursos que las copias de seguridad de la base de datos. Como resultado, puede crearlas con más frecuencia que las copias de seguridad de la base de datos. Si realiza copias de seguridad frecuentes disminuye el riesgo de perder datos.

**Nota** Algunas veces, una copia de seguridad del registro de transacciones es mayor que una copia de seguridad de la base de datos. Por ejemplo, una base de datos tiene una tasa de transacciones alta, lo que provoca que el registro de transacciones crezca rápidamente. En este caso, cree las copias de seguridad del registro de transacciones con más frecuencia.

Las copias de seguridad del registro de transacciones sólo se utilizan con los modelos de recuperación completa y de registro masivo.

Utilizar copias de seguridad del registro de transacciones con copias de seguridad de la base de datos

La restauración de una base de datos mediante la utilización de copias de seguridad de la base de datos y del registro de transacciones puede realizarse sólo si dispone de una secuencia ininterrumpida de copias de seguridad del registro de transacciones después de la última copia de seguridad diferencial o copia de seguridad de la base de datos. Si una copia de seguridad del registro se pierde o se daña, debe crear una copia de seguridad de base de datos o diferencial de base de datos y comenzar de nuevo la copia de seguridad de los registros de transacciones. Guarde las copias de seguridad anteriores de los registros de transacciones si desea restaurar la base de datos hasta un momento incluido en esas copias de seguridad.

El único momento en el que las copias de seguridad diferenciales o las copias de seguridad de la base de datos deben estar sincronizadas con las copias de seguridad del registro de transacciones es cuando se inicia una secuencia de copias de seguridad del

registro de transacciones. Cada secuencia de copias de seguridad del registro de transacciones se debe iniciar con una copia de seguridad diferencial o una copia de seguridad de la base de datos.

Normalmente, sólo se inicia una secuencia nueva de copias de seguridad cuando se realiza una copia de seguridad de la base de datos por primera vez o se cambia el modelo de recuperación de simple a completo o de registro masivo

### **Truncar el registro de transacciones**

Cuando SQL Server termina de realizar una copia de seguridad del registro de transacciones, trunca automáticamente la parte inactiva del registro. Esta parte inactiva contiene las transacciones terminadas, por lo que no se utilizará durante el proceso de recuperación. Por el contrario, la parte activa del registro de transacciones contiene transacciones que se están ejecutando y no se han completado todavía. SQL Server vuelve a usar este espacio truncado e inactivo del registro de transacciones en lugar de seguir aumentando el registro y utilizar más espacio.

Nota Aunque el registro de transacciones se puede truncar manualmente, se recomienda que no lo haga ya que interrumpe la cadena de copia de seguridad del registro. Hasta que se realiza una copia de seguridad completa de la base de datos, la base de datos no está protegida contra errores de los medios. Utilice el truncamiento manual del registro sólo en circunstancias especiales y realice una copia de seguridad completa de la base de datos tan pronto como sea posible.

El punto final de la parte inactiva del registro de transacciones y, por tanto, el punto de truncamiento, lo marca el suceso de entre los siguientes que se produzca primero:

- El último punto de comprobación.

- El inicio de la transacción activa más antigua, que es una transacción que no se ha confirmado o deshecho todavía.

Esto representa el primer punto hasta el que SQL Server deberá deshacer las transacciones durante la recuperación.



El inicio de la transacción más antigua que incluye objetos publicados para duplicación cuyos cambios no se han duplicado todavía.

Esto representa el primer punto que SQL Server todavía tiene que duplicar.

### **Condiciones para realizar copias de seguridad del registro de transacciones**

No se puede realizar una copia de seguridad del registro de transacciones durante una copia de seguridad completa o diferencial de base de datos. No obstante, se puede realizar una copia de seguridad del registro de transacciones mientras se realiza una copia de seguridad de archivos.

No realice una copia de seguridad del registro de transacciones:

Hasta que se haya creado una copia de seguridad de base de datos o de archivos, ya que el registro de transacciones contiene los cambios realizados en la base de datos después de la última copia de seguridad.

Si el registro de transacciones se ha truncado explícitamente, a menos que se haya creado una copia de seguridad de base de datos o diferencial de base de datos después de que se produzca el truncamiento del registro de transacciones.

Restaurar copias de seguridad de registros de transacciones

No es posible aplicar una copia de seguridad del registro de transacciones:

A menos que se restaure primero la copia de seguridad diferencial o la copia de seguridad de la base de datos que precede a la copia de seguridad del registro de transacciones.

A menos que se apliquen primero todas las copias de seguridad del registro de transacciones creadas antes de la copia de seguridad diferencial o la copia de seguridad de la base de datos.

Si se pierde o se daña una copia de seguridad anterior del registro de transacciones, puede restaurar sólo los registros de transacciones hasta la última copia de seguridad anterior al registro de transacciones perdido.

Si la base de datos ya está recuperada y todas las transacciones pendientes ya se han deshecho o confirmado.

Cuando se apliquen copias de seguridad del registro de transacciones, la base de datos no se debe recuperar hasta que se haya aplicado la copia de seguridad final del registro de transacciones. Si permite que tenga lugar la recuperación cuando se aplique una de las copias de seguridad intermedias del registro de transacciones, no podrá restaurar más allá de ese momento sin volver a iniciar toda la operación de restauración, comenzando por la copia de seguridad de la base de datos.

### **Crear una secuencia de copias de seguridad del registro de transacciones**

Para crear conjuntos de copias de seguridad, normalmente se realizan copias de seguridad de la base de datos a intervalos periódicos, por ejemplo, diariamente, y copias de seguridad del registro de transacciones a intervalos más cortos, por ejemplo, cada 10 minutos. Debe tener al menos una copia de seguridad de la base de datos, o un conjunto de copias de seguridad de archivos, para realizar copias de seguridad del registro útiles. El intervalo entre las copias de seguridad varía según la importancia de los datos y la carga de trabajo del servidor. Si el registro de transacciones resulta dañado, perderá el trabajo realizado desde la última copia de seguridad del registro. Por lo tanto, es recomendable realizar copias de seguridad frecuentes de los registros de datos fundamentales y situar los archivos del registro en medios de almacenamiento con tolerancia a errores.

La secuencia de las copias de seguridad del registro de transacciones es independiente de las copias de seguridad de la base de datos. Realice una secuencia de copias de seguridad del registro de transacciones y luego copias de seguridad periódicas de la base de datos, que se utilizan para iniciar una operación de restauración. Por ejemplo, supongamos la siguiente secuencia de sucesos.

Hora	Suceso
8:00 a.m.	Copia de seguridad de la base de datos

Mediodía	Copia de seguridad del registro de transacciones
04:00:00 p.m.	Copia de seguridad del registro de transacciones
06:00:00 p.m.	Copia de seguridad de la base de datos
08:00:00 p.m.	Copia de seguridad del registro de transacciones
10:00 p.m.	Se produce el error

La copia de seguridad del registro de transacciones creada a las 8:00 p.m. contiene registros realizados desde las 4:00 p.m. a las 8:00 p.m., que abarca el momento en que se creó la copia de seguridad de la base de datos (a las 6:00 p.m.). La secuencia de copias de seguridad del registro de transacciones es continua desde la primera copia de seguridad de la base de datos, creada a las 8:00 a.m., a la última copia de seguridad del registro de transacciones, creada a las 8:00 p.m. Se pueden utilizar los siguientes procedimientos para restaurar la base de datos a su estado a las 10:00 p.m. (momento del error):

Restaurar la base de datos con la última copia de seguridad de la base de datos que se haya creado.

Crear una copia de seguridad del registro de transacciones activo actualmente.

Restaurar la copia de seguridad de la base de datos creada a las 6:00 p.m. y, después, aplicar las copias de seguridad del registro de transacciones activo y de las 8:00 p.m.

El proceso de restauración detecta que la copia de seguridad del registro de transacciones de las 8:00 p.m. contiene transacciones que se han producido antes de la última copia de seguridad restaurada. Por tanto, en la operación de restauración se examina el registro de transacciones hasta el momento correspondiente a la finalización de la copia de seguridad de la base de datos a las 6:00 p.m. y se confirman sólo las transacciones completadas desde ese momento en adelante en la copia de seguridad del

registro de transacciones. Esto ocurre de nuevo con la copia de seguridad del registro de transacciones creada a las 10:00 p.m.

Restaure la base de datos utilizando una copia de seguridad anterior de la base de datos (con anterioridad a la última copia de seguridad creada de la base de datos).

Cree una copia de seguridad del registro de transacciones activo actualmente.

Restaure la copia de seguridad de la base de datos creada a las 8:00 a.m. y, después, restaure secuencialmente las cuatro copias de seguridad del registro de transacciones. No restaure la copia de seguridad de la base de datos creada a las 6:00 p.m. Esta acción confirma todas las transacciones completadas hasta las 10:00 p.m.

Este proceso llevará más tiempo que restaurar la copia de seguridad de la base de datos creada a las 6:00 p.m.

La segunda opción destaca la mayor seguridad que ofrece una cadena de copias de seguridad del registro de transacciones que se puede utilizar para restaurar una base de datos aunque se pierda la copia de seguridad de ésta. Puede restaurar una copia de seguridad anterior de la base de datos y, después, restaurar todas las copias de seguridad del registro de transacciones creadas después de esa copia de seguridad.

Nota Es importante no perder una copia de seguridad del registro de transacciones. Considere la posibilidad de crear varias copias de los conjuntos de copias de seguridad del registro. Para ello, realice una copia de seguridad del registro en un disco y, a continuación, copie el archivo del disco a otro dispositivo, por ejemplo, otro disco o una cinta.

### **Registros de transacciones y recuperación**

Cuando termina la operación de restauración y recuperación de la base de datos, se deshacen todas las transacciones incompletas. Esto es necesario para restaurar la integridad de la base de datos.

Una vez realizado esto, ya no se pueden aplicar más copias de seguridad del registro de transacciones a la base de datos. Por ejemplo, una serie de copias de seguridad del

registro de transacciones contiene una transacción de larga duración. El inicio de la transacción se registra en la primera copia de seguridad del registro de transacciones, pero el final de la transacción se registra en la segunda copia de seguridad. En la primera copia de seguridad del registro de transacciones no se registra ninguna operación de confirmación o deshacer. Por lo tanto, si se lleva a cabo una operación de recuperación al aplicar por primera vez la copia de seguridad del registro de transacciones, la transacción de larga duración se trata como incompleta. Las modificaciones de los datos registradas en la primera copia de seguridad del registro de transacciones se deshacen. SQL Server no permite aplicar la segunda copia de seguridad del registro de transacciones después de haberse realizado la recuperación.

Por tanto, cuando se restauren copias de seguridad del registro de transacciones, la base de datos no se deberá recuperar hasta que se haya aplicado el registro de transacciones final. Esto evita que se deshagan transacciones parcialmente. El único momento en el que es necesario deshacer transacciones pendientes es al final de la última operación de restauración.

## El comando BACKUP

Realiza la copia de seguridad de una base de datos completa, del registro de transacciones o de uno o más archivos o grupos de archivos.

### Sintaxis

#### Realizar la copia de seguridad de una base de datos completa:

```
BACKUP DATABASE { database_name | @database_name_var }  
TO < backup_device > [ ,...n ]  
[ WITH  
    [ BLOCKSIZE = { blocksize | @blocksize_variable } ]  
    [ [ , ] DESCRIPTION = { 'text' | @text_variable } ]  
    [ [ , ] DIFFERENTIAL ]  
    [ [ , ] EXPIREDATE = { date | @date_var }  
      | RETAINDAYS = { days | @days_var } ]  
    [ [ , ] PASSWORD = { password | @password_variable } ]  
    [ [ , ] FORMAT | NOFORMAT ]  
    [ [ , ] { INIT | NOINIT } ]  
    [ [ , ] MEDIADESCRIPTION = { 'text' | @text_variable } ]  
    [ [ , ] MEDIANAME = { media_name | @media_name_variable } ]  
    [ [ , ] MEDIAPASSWORD = { mediapassword | @mediapassword_variable } ]  
    [ [ , ] NAME = { backup_set_name | @backup_set_name_var } ]
```

```
[[,]{NOSKIP|SKIP}]
[[,]{NOREWIND|REWIND}]
[[,]{NOUNLOAD|UNLOAD}]
[[,]RESTART]
[[,]STATS [= percentage ]]
```

```
]
```

### Realizar la copia de seguridad de archivos o grupos de archivos específicos:

```
BACKUP DATABASE { database_name | @database_name_var }
    < file_or_filegroup > [ ,...n ]
TO < backup_device > [ ,...n ]
[ WITH
    [ BLOCKSIZE = { blocksize | @blocksize_variable } ]
    [[,] DESCRIPTION = { 'text' | @text_variable } ]
    [[,] EXPIREDATE = { date | @date_var }
        | RETAINDAYS = { days | @days_var } ]
    [[,] PASSWORD = { password | @password_variable } ]
    [[,] FORMAT | NOFORMAT ]
    [[,]{INIT|NOINIT}]
    [[,] MEDIADESCRIPTION = { 'text' | @text_variable } ]
    [[,] MEDIANAME = { media_name | @media_name_variable } ]
    [[,] MEDIAPASSWORD = { mediapassword | @mediapassword_variable } ]
    [[,] NAME = { backup_set_name | @backup_set_name_var } ]
    [[,]{NOSKIP|SKIP}]
    [[,]{NOREWIND|REWIND}]
    [[,]{NOUNLOAD|UNLOAD}]
    [[,]RESTART]
    [[,]STATS [= percentage ]]
```

```
]
```

### Realizar la copia de seguridad de un registro de transacciones:

```
BACKUP LOG { database_name | @database_name_var }
{
    TO < backup_device > [ ,...n ]
    [WITH
        [ BLOCKSIZE = { blocksize | @blocksize_variable } ]
        [[,] DESCR  PTION = { 'text' | @text_variable } ]
        [[,] EXPIREDATE = { date | @date_var }
            | RETAINDAYS = { days | @days_var } ]
        [[,] PASSWORD = { password | @password_variable } ]
        [[,] FORMAT | NOFORMAT ]
        [[,]{INIT|NOINIT}]
        [[,] MEDIADESCRIPTION = { 'text' | @text_variable } ]
        [[,] MEDIANAME = { media_name | @media_name_variable } ]
        [[,] MEDIAPASSWORD = { mediapassword | @mediapassword_variable } ]
        [[,] NAME = { backup_set_name | @backup_set_name_var } ]
        [[,] NO_TRUNCATE ]
        [[,]{NORECOVERY|STANDBY = undo_file_name} ]
        [[,]{NOREWIND|REWIND}]
        [[,]{NOSKIP|SKIP}]
        [[,]{NOUNLOAD|UNLOAD}]
        [[,]RESTART]
```

```
    [ [ , ] STATS [ = percentage ] ]  
  ]  
}  
  
< backup_device > ::=  
{  
    { logical_backup_device_name | @logical_backup_device_name_var }  
    |  
    { DISK | TAPE } =  
    { 'physical_backup_device_name' | @physical_backup_device_name_var }  
}  
  
< file_or_filegroup > ::=  
{  
    FILE = { logical_file_name | @logical_file_name_var }  
    |  
    FILEGROUP = { logical_filegroup_name | @logical_filegroup_name_var }  
}
```

### Truncar el registro de transacciones:

```
BACKUP LOG { database_name | @database_name_var }  
{  
    [WITH  
        { NO_LOG | TRUNCATE_ONLY } ]  
}
```

### Argumentos

#### DATABASE

Especifica una copia de seguridad completa de la base de datos. Si se especifica una lista de archivos y grupos de archivos, sólo se realiza la copia de seguridad de esos archivos o grupos de archivos.

Nota Durante una copia de seguridad completa o diferencial de una base de datos, Microsoft® SQL Server™ realiza la copia de seguridad de una parte del registro de transacciones suficiente para poder producir una base de datos coherente cuando se restaure la base de datos. Sólo se puede realizar una copia de seguridad completa de la base de datos master.

{ *database\_name* | @*database\_name\_var* }

Es la base de datos para la que se realiza la copia de seguridad del registro de transacciones, de una parte de la base de datos o de la base de datos completa. Si se suministra como variable (@*database\_name\_var*), este nombre se puede especificar como una constante de cadena (@*database\_name\_var* = nombre de la base de datos) o como una variable de un tipo de datos de cadena de caracteres, excepto los tipos de datos ntext o text.

<backup\_device>

Especifica el dispositivo de copia de seguridad físico o lógico que se va a utilizar para la operación de copia de seguridad. Puede ser uno o más de los siguientes:

`{ logical_backup_device_name } | { @logical_backup_device_name_var }`

`{ DISK | TAPE } =`

`'physical_backup_device_name' | @physical_backup_device_name_var`

Permite crear copias de seguridad en el dispositivo de disco o cinta especificado. No es necesario que exista el dispositivo físico antes de ejecutar la instrucción BACKUP. Si el dispositivo físico existe y no se especifica la opción INIT en la instrucción BACKUP, la copia de seguridad se anexa al dispositivo.

Si especifica TO DISK o TO TAPE, escriba la ruta de acceso y el nombre de archivo completos. Por ejemplo, DISK = 'C:\Archivos de programa\Microsoft SQL Server\MSSQL\BACKUP\MYBACKUP.DAT' o TAPE = '\\.\TAPE0'.

*n*

Es un marcador de posición que indica que se pueden especificar varios dispositivos de copia de seguridad. El número máximo de dispositivos de copia de seguridad es 64.

`BLOCKSIZE = { blocksize | @blocksize_variable }`

Especifica el tamaño de bloque físico, en bytes. En sistemas Windows NT, el tamaño predeterminado es el tamaño del bloque predeterminado del dispositivo. Normalmente, este parámetro no es necesario ya que SQL Server elige un tamaño de bloque correspondiente al dispositivo. En equipos con Windows 2008, el tamaño predeterminado es 65.536 (64 KB, que es el tamaño máximo que admite SQL Server).

`DESCRIPTION = { 'text' | @text_variable }`

Especifica el texto de formato libre que describe el conjunto de copias de seguridad. La cadena puede tener un máximo de 255 caracteres.

**DIFFERENTIAL**

Especifica que la copia de seguridad de base de datos o de archivo sólo debe constar de las partes de la base de datos o el archivo que hayan cambiado desde la última copia de seguridad completa. Una copia de seguridad diferencial suele ocupar menos espacio que una copia de seguridad completa. Utilice esta opción para que no sea necesario aplicar todas las copias de seguridad de registro individuales efectuadas desde que se realizó la última copia de seguridad completa.

`EXPIREDATE = { date | @date_var }`

Especifica la fecha en que el conjunto de copias de seguridad caduca y se puede sobrescribir. .

`RETAIN_DAYS = { days | @days_var }`

Especifica el número de días que deben transcurrir antes de que se pueda sobrescribir este conjunto de medios de copia de seguridad. Si se proporciona como una variable ( *@days\_var* ), se debe especificar como un valor de tipo integer (entero).



PASSWORD = { *password* | *@password\_variable* }

Establece la contraseña del conjunto de copias de seguridad. PASSWORD es una cadena de caracteres.

FORMAT

Especifica que el encabezado del medio se debe escribir en todos los volúmenes utilizados en esta operación de copia de seguridad. Se sobrescriben los encabezados de medio existentes. La opción FORMAT invalida todo el contenido del medio y omite el contenido existente.

NOFORMAT

Especifica que el encabezado del medio no se debe escribir en todos los volúmenes utilizados en esta operación de copia de seguridad y no vuelve a escribir el dispositivo de copia de seguridad a menos que se especifique INIT.

INIT

Especifica que se deben sobrescribir todos los conjuntos de copia de seguridad, pero conserva el encabezado de los medios. Si se especifica INIT, se sobrescriben los datos del conjunto de copias de seguridad existentes en el dispositivo.

NOINIT

Indica que el conjunto de copias de seguridad se anexa al dispositivo de disco o cinta especificado, conservando así los conjuntos de copia de seguridad existentes. NOINIT es el valor predeterminado.

La opción FILE del comando RESTORE se utiliza para seleccionar el conjunto de copias de seguridad apropiado en el momento de la restauración. Si se define una contraseña para un medio del conjunto, debe suministrarla.

MEDIADESCRIPTION = { *text* | *@text\_variable* }

Especifica la descripción de texto con formato libre, con un máximo de 255 caracteres, del conjunto de medios.

MEDIANAME = { *media\_name* | *@media\_name\_variable* }

Especifica el nombre del medio, con un máximo de 128 caracteres, de todo el conjunto de medios de copia de seguridad. MEDIAPASSWORD = { *mediapassword* | *@mediapassword\_variable* }

Establece la contraseña del conjunto de medios. MEDIAPASSWORD es una cadena de caracteres.

NAME = { *backup\_set\_name* | *@backup\_set\_var* }

Especifica el nombre del conjunto de copias de seguridad. Los nombres pueden tener un máximo de 128 caracteres. Si no se especifica NAME, está en blanco.

NORECOVERY

Se utiliza sólo con BACKUP LOG. Realiza una copia de seguridad del final del registro y deja la base de datos en estado de restauración. NORECOVERY resulta útil cuando, en caso de error, se conmuta a una base de datos secundaria y al guardar el final del registro antes de una operación RESTORE.

STANDBY = *undo\_file\_name*

Se utiliza sólo con BACKUP LOG. Realiza una copia de seguridad del final del registro y deja la base de datos en modo de sólo lectura y espera. El nombre del archivo para deshacer especifica dónde se almacenarán los cambios que se deben deshacer si se aplican operaciones RESTORE LOG posteriormente.

NOSKIP

Indica a la instrucción BACKUP que compruebe la fecha de caducidad de todos los conjuntos de copia de seguridad de los medios antes de permitir que se sobrescriban.

SKIP

Deshabilita la comprobación de la caducidad y el nombre del conjunto de copias de seguridad que suele realizar la instrucción BACKUP para impedir que se sobrescriban los conjuntos de copia de seguridad.

STATS [ = *percentage* ]

Muestra un mensaje cada vez que se completa otro porcentaje (*percentage*); se utiliza para indicar el progreso. Si se omite *percentage*, SQL Server muestra un mensaje después de completar cada 10 por ciento.

< *file\_or\_filegroup* >

Especifica los nombres lógicos de los archivos o grupos de archivos que se incluirán en la copia de seguridad de la base de datos. Se pueden especificar varios archivos o grupos de archivos.

FILE = { *logical\_file\_name* | @*logical\_file\_name\_var* }

Proporciona el nombre de uno o más archivos que se incluirán en la copia de seguridad de la base de datos.

FILEGROUP = { *logical\_filegroup\_name* | @*logical\_filegroup\_name\_var* }

Proporciona el nombre de uno o más grupos de archivos que se incluirán en la copia de seguridad de la base de datos.

*n*

Se trata de un marcador de posición que indica que se pueden especificar varios archivos y grupos de archivos. No hay un número máximo de archivos o grupos de archivos.

LOG

Especifica que sólo se realizará la copia de seguridad del registro de transacciones. Se realiza la copia de seguridad del registro desde la última copia de seguridad LOG ejecutada correctamente hasta el final actual del registro. Después de realizar la copia de seguridad del registro, el espacio se puede truncar cuando ya no sea necesario para duplicación o transacciones activas.

#### NO\_LOG | TRUNCATE\_ONLY

Elimina la parte inactiva del registro sin realizar una copia de seguridad del mismo y trunca el registro. Esta opción libera espacio. No es necesario especificar un dispositivo de copia de seguridad debido a que la copia de seguridad del registro no se guarda. NO\_LOG y TRUNCATE\_ONLY son sinónimos.

Después de realizar la copia de seguridad del registro con NO\_LOG o TRUNCATE\_ONLY, no se pueden recuperar los cambios grabados en el registro. Para propósitos de recuperación, ejecute inmediatamente BACKUP DATABASE.

#### NO\_TRUNCATE

Permite realizar copias de seguridad del registro cuando la base de datos está dañada.

Ejemplos de copia de seguridad.

#### --COPIAS DE SEGURIDAD

--01 En primer lugar establecer el modo de recuperación de la base de datos en COMPLETA

--02 Crear un dispositivo de copia de seguridad: NW1

USE master

GO

EXECUTE sp\_addumpdevice

    @devtype='DISK'

    ,@logicalname='NW1'

    ,@physicalname='C:\BACKUP\NW1.bak'

GO

--03 Crear la copia completa de Northwind.

BACKUP DATABASE Northwind TO NW1

WITH

DESCRIPTION ='Primera copia completa de Northwind'

,NAME='CopiaCompletaNorthwind'

,NOINIT --- Anexar la copia

,STATS = 10

GO

--04 Crear la copia completa de Northwind.

BACKUP DATABASE Northwind TO NW1

WITH

DESCRIPTION ='Segunda copia completa de Northwind'

,NAME='SegundaCopiaCompletaNorthwind'

,NOINIT --- Anexar la copia

,STATS = 10

GO

--05 Crear un dispositivo de copia de seguridad: NWCHANGE

USE master

GO

EXECUTE sp\_addumpdevice

    @devtype='DISK'

    ,@logicalname='NWCHANGE'

    ,@physicalname ='C:\BACKUP\NWCHANGE.bak'

GO

--06 Crear la copia del log de transacciones de la base de datos Northwind

```
BACKUP LOG Northwind TO NWCHANGE
```

```
WITH
```

```
DESCRIPTION ='2da Copia del log de transacciones de Northwind'
```

```
,NAME='SegundaCopiaLogNorthwind'
```

```
,NOINIT --- Anexar la copia
```

```
,STATS = 10
```

```
GO
```

--07 Crear la copia Diferencial de Northwind.

```
BACKUP DATABASE Northwind TO NW1
```

```
WITH
```

```
DIFFERENTIAL
```

```
,DESCRIPTION ='copia Diferencial de Northwind'
```

```
,NAME='CopiaDiferencialNorthwind'
```

```
,NOINIT --- Anexar la copia
```

```
,STATS = 10
```

```
GO
```

# Capítulo 05

## Restauración de seguridad de base de datos

### Actividades de SQL Server durante el proceso de restauración

Al restaurar bases de datos, SQL Server realiza automáticamente determinadas acciones para asegurar que la restauración se realiza rápidamente y tiene un efecto mínimo en las actividades de producción.

### Comprobación de seguridad

SQL Server hace una comprobación de seguridad al ejecutar la instrucción RESTORE DATABASE. Este mecanismo interno impide que se sobrescriba accidentalmente una base de datos existente con una copia de seguridad de una base de datos distinta o con información incompleta. SQL Server no restaura la base de datos si:

- \_ La base de datos indicada en la instrucción RESTORE DATABASE ya existe en el servidor y el nombre de base de datos registrado en el archivo de copia de seguridad es distinto del de la base de datos especificada en la instrucción RESTORE DATABASE.
- \_ El conjunto de archivos de base de datos del servidor es distinto del contenido en el conjunto de copia de seguridad.
- \_ No se proporcionan todos los archivos necesarios para restaurar la base de datos o grupo de archivos. SQL Server genera un mensaje de error en el que especifica los archivos que deben restaurarse como una unidad (en una misma operación de restauración).

Por ejemplo, si intenta restaurar una copia de seguridad de la base de datos Northwind en la base de datos Accounting, que ya existe en el servidor, SQL Server impedirá la restauración. Si desea restaurar una copia de seguridad de Northwind y sobrescribir los datos de Accounting, deberá anular la comprobación de seguridad.

## **Regeneración de la base de datos y todos los archivos asociados**

Al restaurar una base de datos a partir de una copia de seguridad completa, SQL Server vuelve a crear los archivos originales de la base de datos y los coloca en las ubicaciones que se registraron al hacer la copia de seguridad.

Todos los objetos de base de datos se vuelven a crear automáticamente. No es necesario volver a generar el esquema de la base de datos antes de restaurarla.

## **Preparación de la restauración de una base de datos**

Es conveniente comprobar las copias de seguridad para confirmar que se van a restaurar los datos y objetos que se desea, y que la copia de seguridad contiene información válida. Antes de restaurar una copia de seguridad, debe realizar tareas específicas que le permitirán comenzar el proceso de restauración.

## **Comprobación de las copias de seguridad**

Antes de restaurar un archivo de copia de seguridad, debe asegurarse de que es válido y de que contiene las copias de seguridad esperadas. Puede utilizar el Administrador corporativo de SQL Server para ver la hoja de propiedades de cada dispositivo de copia de seguridad. Para obtener información más detallada acerca de las copias de seguridad, puede ejecutar las instrucciones de Transact-SQL siguientes:

## **Instrucción RESTORE HEADERONLY**

Puede usar esta instrucción para obtener la información de encabezado de un archivo o conjunto de copia de seguridad determinado. Si en un mismo archivo residen varias copias de seguridad, SQL Server obtendrá información de encabezado de todas ellas.

Al ejecutar la instrucción RESTORE HEADERONLY, recibirá la información siguiente:

- \_ Nombre y descripción del archivo o conjunto de copia de seguridad
- \_ Tipo del medio de copia de seguridad empleado, por ejemplo, cinta o disco duro
- \_ Método de copia de seguridad, por ejemplo, completa, diferencial, registro de transacciones o archivo
- \_ Fecha y hora de realización de la copia de seguridad

- \_ Tamaño de la copia de seguridad
- \_ Número de secuencia de una copia de seguridad específica en una cadena de archivos de copia de seguridad.

### **Instrucción RESTORE FILELISTONLY**

Puede usar esta instrucción para obtener información acerca de los archivos originales de base de datos o de registro de transacciones contenidos en un archivo de copia de seguridad. La ejecución de esta instrucción puede ayudarle a evitar que se restauren archivos de copia de seguridad equivocados.

Al ejecutar la instrucción RESTORE FILELISTONLY, SQL Server devuelve la información siguiente:

- \_ Nombres lógicos de los archivos de base de datos y de registro de transacciones
- \_ Nombres físicos de los archivos de base de datos y de registro de transacciones
- \_ Tipo de archivo, por ejemplo, de base de datos o de registro de transacciones
- \_ Pertenencia a grupos de archivos
- \_ Tamaño del conjunto de copia de seguridad, en megabytes (MB)
- \_ Tamaño de archivo máximo permitido, en MB.

### **Instrucción RESTORE LABELONLY**

Puede usar esta instrucción para obtener información acerca del medio de copia de seguridad que almacena un archivo de copia de seguridad.

### **Instrucción RESTORE VERIFYONLY**

Puede usar esta instrucción para comprobar que los archivos individuales que forman el conjunto de copia de seguridad están completos y que todas las copias de seguridad son legibles. SQL Server no comprueba la estructura de los datos contenidos en la copia de seguridad.

### **Realización de tareas específicas antes de restaurar copias de seguridad**

Antes de restaurar copias de seguridad, debe restringir el acceso a la base de datos y hacer una copia de seguridad del registro de transacciones.



## **Restringir el acceso a la base de datos**

Es importante restringir el acceso a una base de datos antes de restaurarla. Establezca la opción de base de datos **Miembros de db\_owner, dbcreator** o **sysadmin** a true.

## **Hacer una copia de seguridad del registro de transacciones**

La coherencia de la base de datos quedará asegurada si hace una copia de seguridad del registro de transacciones antes de cualquier operación de restauración:

\_ La copia de seguridad del registro de transacciones se utiliza para recuperar la base de datos en el último paso del proceso de restauración.

\_ Si no hace una copia de seguridad del registro de transacciones antes de restaurar las copias de seguridad, se perderán las modificaciones de datos que se hayan producido entre la última copia de seguridad del registro de transacciones y el momento en que se desconectó la base de datos.

Si utiliza el Analizador de consultas de SQL Server o ejecuta una secuencia de comandos desde el símbolo del sistema para restaurar copias de seguridad, debe utilizar la base de datos **master**.

## **Restauración de copias de seguridad**

Puede utilizar el Administrador corporativo de SQL Server o la instrucción **RESTORE**, e indicar las opciones específicas del tipo de copia de seguridad que va a restaurar. También puede determinar si se debe iniciar el proceso de recuperación después de cada operación de restauración.

## **Uso de la instrucción RESTORE**

Para realizar operaciones de restauración, puede utilizar la instrucción **RESTORE** o el Administrador corporativo de SQL Server.

Las opciones de restauración permiten especificar detalles acerca de cómo restaurar las copias de seguridad.

Sintaxis:

```
RESTORE DATABASE {nombreBaseDatos | @varNombreBaseDatos}
[FROM <dispositivoCopiaSeguridad> [, ... n]]
[WITH
[FILE = númeroArchivo]
[[,] MOVE 'nombreLógico' TO 'nombreEnSistemaOperativo']
[[,] REPLACE]
[[,] {NORECOVERY | RECOVERY | STANDBY =
nombreArchivoParaDeshacer}]]
[[,] RESTART
Donde <dispositivoCopiaSeguridad> es
{{nombreDispositivoCopia | @varNombreDispositivoCopia} |
{DISK | TAPE | PIPE} = {'dispositivoCopiaTemporal' |
@varDispositivoCopiaTemporal}
}
```

En este ejemplo se restaura la base de datos **Northwind** a partir de un archivo de copia de seguridad permanente.

```
USE master
```

```
RESTORE DATABASE Northwind
```

```
FROM NwindBac
```

### Restauración de bases de datos de usuario dañadas

Si una base de datos resulta dañada, puede usar la instrucción RESTORE o el Administrador corporativo de SQL Server para restaurarla sobre la base de datos existente. Al restaurar sobre una base de datos:

- \_ No es necesario quitar la base de datos dañada.
- \_ SQL Server vuelve a crear automáticamente los archivos y objetos de la base de datos.

## **Inicio del proceso de recuperación**

Siembre debe especificar la opción RECOVERY o NORECOVERY para evitar errores administrativos durante el proceso de restauración y para hacer más legible la instrucción. La opción RECOVERY es la predeterminada en SQL Server.

### **Especificar la opción RECOVERY**

Utilice esta opción con el último registro de transacciones que se deba restaurar o con una restauración de la base de datos completa, para devolver la base de datos a un estado coherente:

- \_ SQL Server deshace todas las transacciones sin confirmar en el registro de transacciones y aplica todas las transacciones confirmadas.
- \_ La base de datos queda disponible para utilizarse una vez completado el proceso de recuperación.

No utilice esta opción si tiene otros registros de transacciones o copias de seguridad diferenciales que deba restaurar.

### **Especificar la opción NORECOVERY**

Utilice esta opción cuando deba restaurar varias copias de seguridad. Al utilizar la opción NORECOVERY, tenga en cuenta los hechos siguientes:

- \_ Especifique la opción NORECOVERY para todas las copias de seguridad, *excepto* la última que haya que restaurar.
- \_ SQL Server no deshará las transacciones sin confirmar en el registro de transacciones ni aplicará las confirmadas.
- \_ La base de datos no estará disponible para utilizarse hasta que se recupere.

## **Restauración de una copia de seguridad completa de la base de datos**

Al restaurar una base de datos a partir de una copia de seguridad completa, SQL Server vuelve a crear la base de datos y todos sus archivos asociados, y los coloca en su ubicación original. Todos los objetos de base de datos se vuelven a crear

automáticamente. No es necesario volver a generar el esquema de la base de datos antes de restaurarla.

### **Situaciones de uso**

Normalmente, la restauración de una copia de seguridad completa se hará en las situaciones siguientes:

- \_ El disco físico de la base de datos está dañado.
- \_ Toda la base de datos está dañada o se ha eliminado.
- \_ Se va a restaurar una copia idéntica de la base de datos en otro servidor SQL Server.

### **Especificación de las opciones de recuperación**

La opción RECOVERY inicia el proceso de recuperación para que la base de datos recobre un estado coherente:

- \_ Si adopta una estrategia de copia de seguridad completa y no dispone de copias de seguridad diferenciales ni del registro de transacciones, especifique la opción RECOVERY.
- \_ Si existe alguna copia de seguridad diferencial o del registro de transacciones, especifique la opción NORECOVERY para posponer el proceso de recuperación hasta que se haya restaurado la última copia de seguridad.

### **Ejemplo:**

En este ejemplo se supone que existe una copia de seguridad completa en el archivo de copia de seguridad permanente **NwindBac** y que se agregan dos copias de seguridad a ese archivo. La base de datos **Northwind** se reemplaza completamente por la segunda copia de seguridad del archivo permanente **NwindBac**. Por último, el proceso de recuperación devuelve la base de datos a un estado coherente (aplica los cambios confirmados y deshace las actividades sin confirmar).

USE master

RESTORE DATABASE Northwind

FROM NwindBac

WITH FILE = 2, RECOVERY

### **Restauración de una copia de seguridad diferencial**

Al restaurar una base de datos a partir de una copia de seguridad diferencial, SQL Server:

- \_ Sólo restaura las partes de la base de datos que han cambiado desde la última copia de seguridad completa.
- \_ Devuelve la base de datos al estado exacto en que se encontraba en el momento de realizar la copia de seguridad diferencial.
- \_ A menudo, lleva menos tiempo restaurar las copias de seguridad diferenciales que aplicar un conjunto de registros de transacciones que representan la misma actividad de la base de datos.

### **Consideraciones para restaurar copias de seguridad diferenciales**

Al restaurar a partir de una copia de seguridad diferencial, tenga en cuenta los siguientes hechos e instrucciones:

- \_ Restaure la copia de seguridad completa *antes* de restaurar una copia de seguridad diferencial.
- \_ La sintaxis para restaurar una copia de seguridad diferencial es la misma que para restaurar una copia de seguridad completa. En lugar de especificar la copia de seguridad completa en la cláusula FROM, especifique el archivo que contiene la copia de seguridad diferencial.
- \_ Especifique la opción NORECOVERY cuando haya registros de transacciones que haya que restaurar; en caso contrario, especifique la opción RECOVERY.

En el ejemplo siguiente se restaura una copia de seguridad diferencial sin recuperar la base de datos. El archivo **NwindBacDiff** contiene una copia de seguridad diferencial.

Mediante la opción NORECOVERY, SQL Server permite restaurar registros de transacciones antes de llevar la base de datos **Northwind** a un estado coherente.

USE master

RESTORE DATABASE Northwind

FROM NwindBacDiff

WITH NORECOVERY

### **Restauración de una copia de seguridad del registro de transacciones**

Al restaurar una copia de seguridad del registro de transacciones, SQL Server restaura los cambios de la base de datos que se han grabado en el registro de transacciones.

Normalmente, los registros de transacciones se restauran para aplicar los cambios efectuados en la base de datos desde la última copia de seguridad completa o diferencial. Y, lo que es más importante, se pueden restaurar registros de transacciones para recuperar una base de datos hasta un momento específico en el tiempo.

### **Consideraciones para restaurar registros de transacciones**

Aunque la restauración de una copia de seguridad diferencial puede acelerar el proceso de restauración, es posible que para asegurar la coherencia de los datos tenga que restaurar copias de seguridad adicionales del registro de transacciones creadas después de la copia de seguridad diferencial.

Antes de restaurar registros de transacciones, debe restaurar la copia de seguridad completa de la base de datos. Cuando tenga varios registros de transacciones que deba aplicar, especifique la opción NORECOVERY para todos ellos, excepto en el último. SQL Server retrasará el proceso de recuperación hasta que se restaure el último registro de transacciones

### **Sintaxis**

```
RESTORE LOG {nombreBaseDatos | @varNombreBaseDatos}
```

```
[FROM <dispositivoCopiaSeguridad> [, ...n]]
```

[WITH

[[{NORECOVERY | RECOVERY | STANDBY =

*nombreArchivoParaDeshacer*}}

[[,] STOPAT = {*fechaHora* | @*varFechaHora*}]

[[,] STOPBEFOREMARK = *nombreMarca* [AFTER *fechaHora*]

[[,] STOPATMARK = *nombreMarca* [AFTER *fechaHora*]

Si está planeando una operación con un riesgo elevado, puede ser aconsejable agregar una marca de registro de forma que pueda restaurar la base de datos hasta ese punto anterior a que comenzara la operación. Para restaurar hasta una marca de registro especificada:

\_ Use la cláusula **WITH STOPATMARK='nombreMarca'** para aplicar todas las transacciones hasta la marca e incluir la que la contiene.

\_ Use la cláusula **WITH STOPBEFOREMARK='nombreMarca'** para aplicar todas las transacciones hasta la marca y excluir la que la contiene.

En este ejemplo se supone que existe una base de datos completa en un archivo de copia de seguridad y que hay dos copias de seguridad del registro de transacciones en otro archivo de copia de seguridad. Se realizan tres operaciones de restauración distintas para asegurar la coherencia de la base de datos.

1. El primer paso restaura una copia de seguridad completa sin recuperar la base de datos.

```
USE master
```

```
RESTORE DATABASE Northwind
```

```
FROM NwindBac
```

```
WITH NORECOVERY
```

2. En el segundo paso se restaura el primer registro de transacciones sin recuperar la base de datos. Se muestra el progreso de la operación de restauración.

```
USE master
```

RESTORE LOG Northwind

FROM NwindBacLog

WITH FILE = 1,

STATS,

NORECOVERY

3. En el tercer paso se restaura el segundo registro de transacciones, se aplican las transacciones confirmadas y se deshacen las que no se han confirmado.

La opción RECOVERY devuelve la base de datos **Northwind** a un estado coherente.

USE master

RESTORE LOG Northwind

FROM NwindBacLog

WITH FILE = 2,

RECOVERY

## Ejemplos de BACKUP y RESTORE

--Pasos preliminares, para el mejor desarrollo de la práctica

--01 Eliminar la base de datos CopiaNorthwind y Northwind .

--02 Crear la base de datos Northwind.

--03 Crear la carpeta en C:\BACKUP.

--04 Eliminar la base de datos pubs

--05 Eliminar la base de datos CopiaPubs

--06 Crear la base de datos pubs.

--06 Verificar que no existan dispositivos de copia de seguridad.

--07 Crear la carpeta C:\DATA.



--01 Verificar el modo de recuperación de la base de datos Northwind

--02 Modificar el modo de recuperación de Northwind, a Completa

USE master

GO

ALTER DATABASE Northwind

SET RECOVERY FULL

WITH NO\_WAIT

GO

--03 Crear la base de datos CopiaNorthwind

--03.1 Crear el dispositivo de copia de seguridad NWC1.

USE master

GO

EXECUTE sp\_addumpdevice

    @devtype='DISK'

    ,@logicalname='NWC1'

    ,@physicalname='C:\BACKUP\NWC1.bak'

GO

--03.2 Crear la copia de seguridad completa de Northwind

BACKUP DATABASE Northwind TO NWC1

WITH

NOINIT

,STATS = 10

GO

--03.3 Ahora, crear la base de datos CopiaNorthwind

--03.4 Antes se debe determinar el número y nombre de los archivos de la copia de seguridad.

```
RESTORE FILELISTONLY FROM NWC1
```

```
GO
```

```
RESTORE DATABASE CopiaNorthwind FROM NWC1
```

```
WITH
```

```
REPLACE
```

```
,RECOVERY
```

```
,MOVE 'Northwind' TO 'C:\BACKUP\copiaNorthwind.mdf'
```

```
,MOVE 'Northwind_log' TO 'C:\BACKUP\copiaNorthwind.ldf'
```

```
GO
```

--04 Crear el dispositivo de copia de seguridad NWC2.

```
USE master
```

```
GO
```

```
EXECUTE sp_addumpdevice
```

```
    @devtype='DISK'
```

```
    ,@logicalname='NWC2'
```

```
    ,@physicalname='C:\BACKUP\NWC2.bak'
```

```
GO
```

--05 Activar la base de datos de prueba CopiaNorthwind.

```
USE CopiaNorthwind
```

```
GO
```

--06 Agregar un registro a la tabla Products de CopiaNorthwind

```
INSERT INTO Products(ProductName,SupplierID,CategoryID,  
  
QuantityPerUnit,UnitPrice,UnitsInStock,UnitsOnOrder,  
  
ReorderLevel,Discontinued)
```

```
VALUES ('Nuevo producto1',1,1,'Cajas 12 ',4.00,200,0,20,0)
```

--07 Consultar la tabla Products.

```
SELECT * FROM Products
```

--08 Crear la primera copia completa de CopiaNorthwind

```
BACKUP DATABASE CopiaNorthwind TO NWC2
```

```
WITH
```

```
NOINIT
```

```
,NAME='Primera copia completa de CopiaNorthwind'
```

```
,STATS = 10
```

```
GO
```

--08 Simular cambios accidentales en la base de datos. Actualizar todos los precios a 0.0

```
UPDATE Products
```

```
    SET UnitPrice = 0.0
```

```
SELECT * FROM Products
```

--09 Restaurar la base de datos CopiaNorthwind, desde la última copia completa de base de datos.

```
RESTORE DATABASE CopiaNorthwind FROM NWC2
```

```
WITH
```

```
FILE =1
```

```
,REPLACE
```

```
,STATS = 10
```

# Capítulo 06

## Automatización de tareas

Microsoft SQL Server le permite automatizar las tareas administrativas. Para automatizar la administración, se definen las tareas administrativas previsibles y, después, se especifican las condiciones en las que se produce cada tarea. El uso de la administración automatizada para controlar las tareas y eventos habituales le permite disponer de tiempo para realizar otras funciones administrativas.

### El agente SQL Server

El Agente SQL Server es un servicio de Microsoft Windows que ejecuta tareas administrativas programadas, denominadas trabajos. El Agente SQL Server utiliza SQL Server para almacenar información de los trabajos. Los trabajos contienen uno o más pasos. Cada paso contiene su propia tarea; por ejemplo, realizar una copia de seguridad de una base de datos. El Agente SQL Server puede ejecutar un trabajo según una programación, como respuesta a un evento específico o a petición. Por ejemplo, si desea realizar una copia de seguridad de todos los servidores de la organización todos los días entre semana después del horario de trabajo, puede automatizar esta tarea. Programe la copia de seguridad para que se ejecute después de las 22:00 h de lunes a viernes; si la copia de seguridad encuentra un problema, el Agente SQL Server puede registrar el evento y notificárselo.

### Definir operadores

El servicio Agente SQL Server admite la notificación de administradores a través de operadores. Los operadores son alias para personas o grupos que pueden recibir una notificación electrónica cuando los trabajos finalizan o se activa una alerta. Los atributos principales de un operador son:

- Nombre del operador

- Información de contacto

Se recomienda definir operadores antes de definir alertas.

Cada operador debe tener asignado un nombre. Los nombres de los operadores deben ser únicos en la instancia de SQL Server y no pueden tener más de **128** caracteres.

#### Requisitos para notificar a un operador

Debe configurar al menos uno de los elementos siguientes para poder notificar a un operador:

- Para enviar un mensaje de correo electrónico mediante la funcionalidad Correo electrónico de base de datos, debe tener acceso a un servidor de correo electrónico que admita SMTP.
- Para enviar un mensaje de correo electrónico mediante la funcionalidad SQL Mail (MAPI extendido), debe tener acceso a un servidor Microsoft Exchange e instalar el cliente Microsoft Outlook y Microsoft Exchange en el equipo en el que se ejecuta SQL Server.
- Para notificar mediante un localizador, debe disponer de hardware o software de otros fabricantes para enviar mensajes de localizador a correo electrónico.
- Para utilizar **net send**, el operador debe haber iniciado sesión en el equipo especificado y el equipo especificado debe permitir la recepción de mensajes desde Windows Messenger.

#### Designar un operador para notificaciones de error

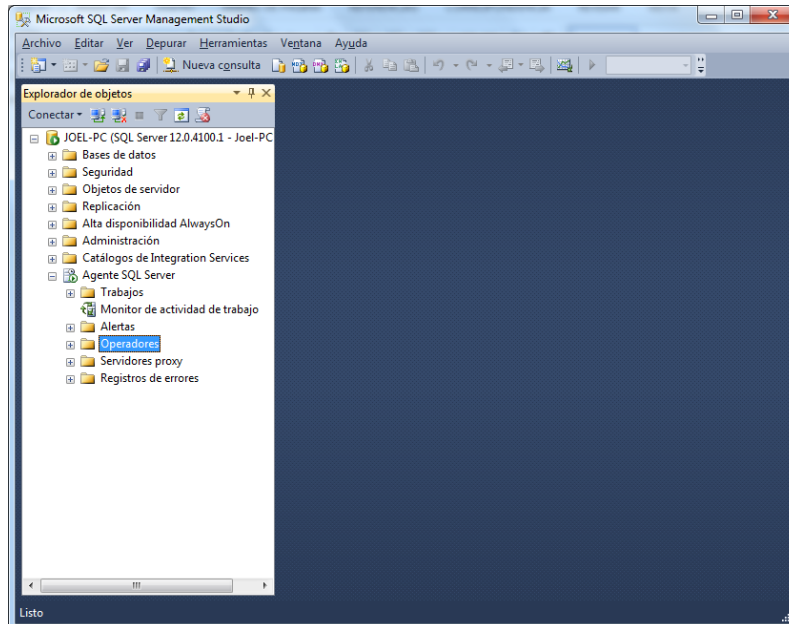
El operador para notificaciones de error recibe la notificación de una alerta después de que no se haya recibido respuesta a ninguna de las notificaciones enviadas mediante localizador a los operadores designados. Se notifica al operador para notificaciones de error cuando:

- No se pueden enviar mensajes al localizador de ninguno de los operadores responsables de la alerta.

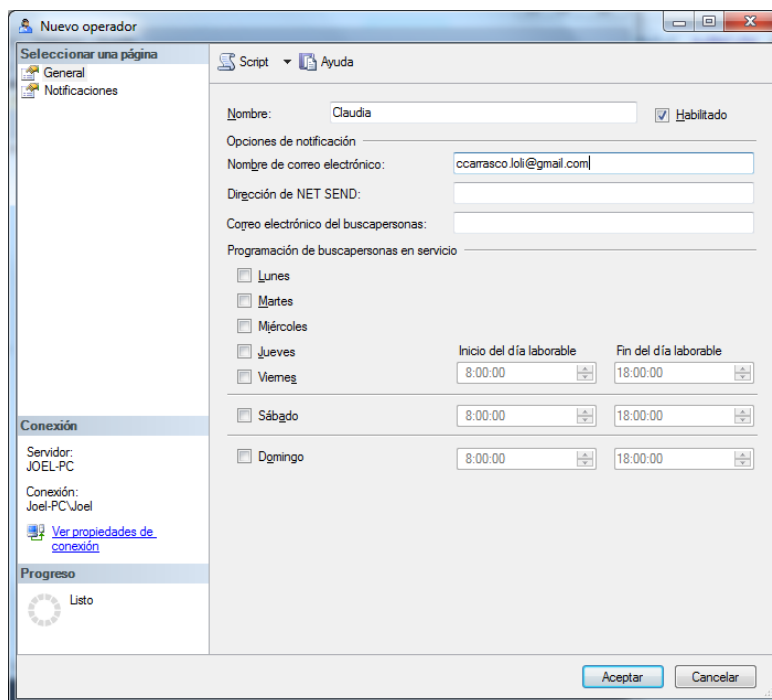
El operador para notificaciones de error es una característica de seguridad.

Crear un operador

En el explorador de objetos seleccionar el Agente SQL Server, Operadores.

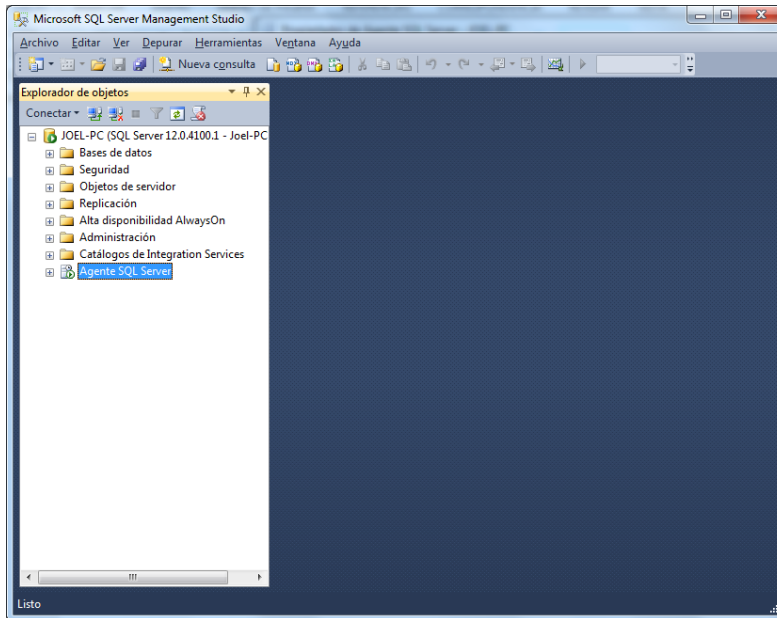


En Operadores clic con derecha Nuevo operador. Ingresar el nombre del operador y la dirección de correo electrónico.

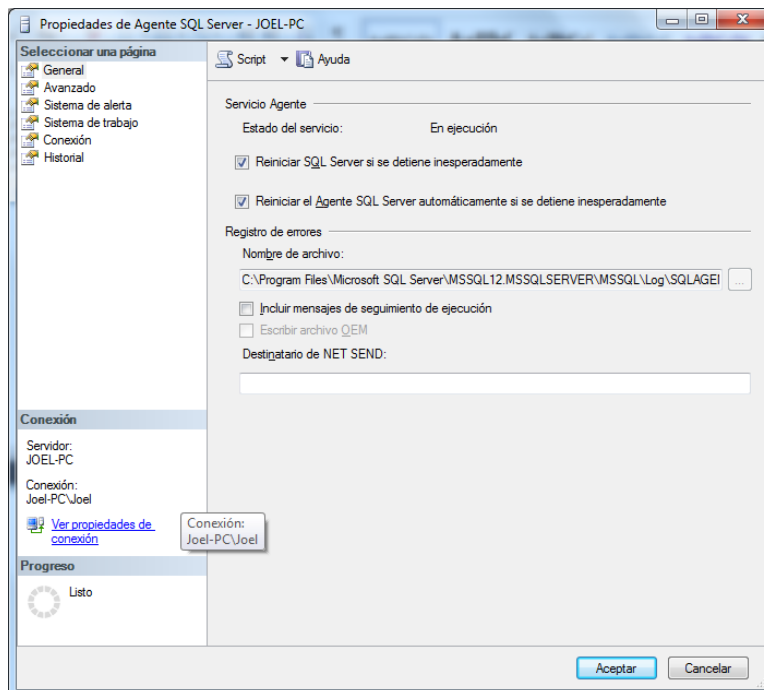


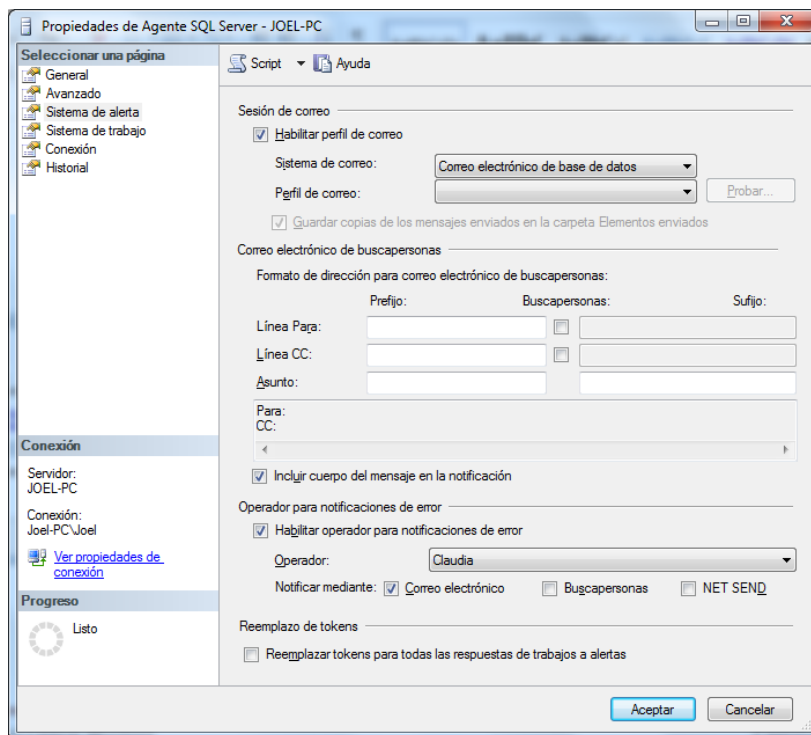
## Configurar el agente SQL Server

Para configurar el Agente SQL Server, en el explorador de objetos seleccionar Agente SQL Server, clic con derecha y del menú contextual seleccionar Propiedades



En la ventana Propiedades del Agente SQL Server, seleccionar la pagina Sistema de alerta





## Implementar trabajos

Puede utilizar los trabajos del Agente SQL Server para automatizar tareas administrativas rutinarias y ejecutarlas periódicamente, lo que permite que la administración sea más eficaz.

Un trabajo es una serie específica de operaciones que el Agente SQL Server realiza secuencialmente. Un trabajo puede realizar una amplia variedad de actividades, incluidas secuencias de comandos Transact-SQL, aplicaciones de línea de comandos, secuencias de comandos de Microsoft ActiveX, paquetes de Integration Services, comandos y consultas de Analysis Services o tareas de réplica. Los trabajos pueden ejecutar tareas repetitivas o que se pueden programar y pueden notificar automáticamente a los usuarios el estado del trabajo mediante alertas, simplificando mucho la administración de SQL Server.

Puede ejecutar los trabajos manualmente o configurarlos para que se ejecuten de acuerdo con una programación o en respuesta a alertas.



## **Crear pasos de trabajo**

Los pasos de trabajo son acciones que el trabajo realiza en una base de datos o en un servidor. Cada trabajo debe estar formado por un paso, como mínimo. Los pasos de trabajo pueden ser:

- Programas ejecutables y comandos del sistema operativo.
- Instrucciones Transact-SQL, incluidos los procedimientos almacenados y los procedimientos almacenados extendidos.
- Secuencias de comandos Microsoft ActiveX.
- Tareas de réplica.
- Tareas de Analysis Services.
- Paquetes de Integration Services.

Puesto que los pasos de trabajo se ejecutan en el contexto de un usuario específico de Microsoft Windows, dicho usuario debe disponer de los permisos y la configuración necesarios para que se ejecute el paso de trabajo. Por ejemplo, si crea un trabajo que requiere una letra de unidad o una ruta de acceso UNC (Convención de nomenclatura universal), los pasos de trabajo se pueden ejecutar con la cuenta de usuario de Microsoft Windows durante la comprobación de las tareas. Sin embargo, el usuario de Windows para el paso de trabajo debe tener también los permisos y configuraciones de letra de unidad necesarios, o acceso a la unidad requerida. De lo contrario, se producirá un error en el paso de trabajo. Para evitar este problema, asegúrese de que el proxy para cada paso de trabajo dispone de los permisos necesarios para la tarea que realiza dicho paso.

## **Crear programaciones**

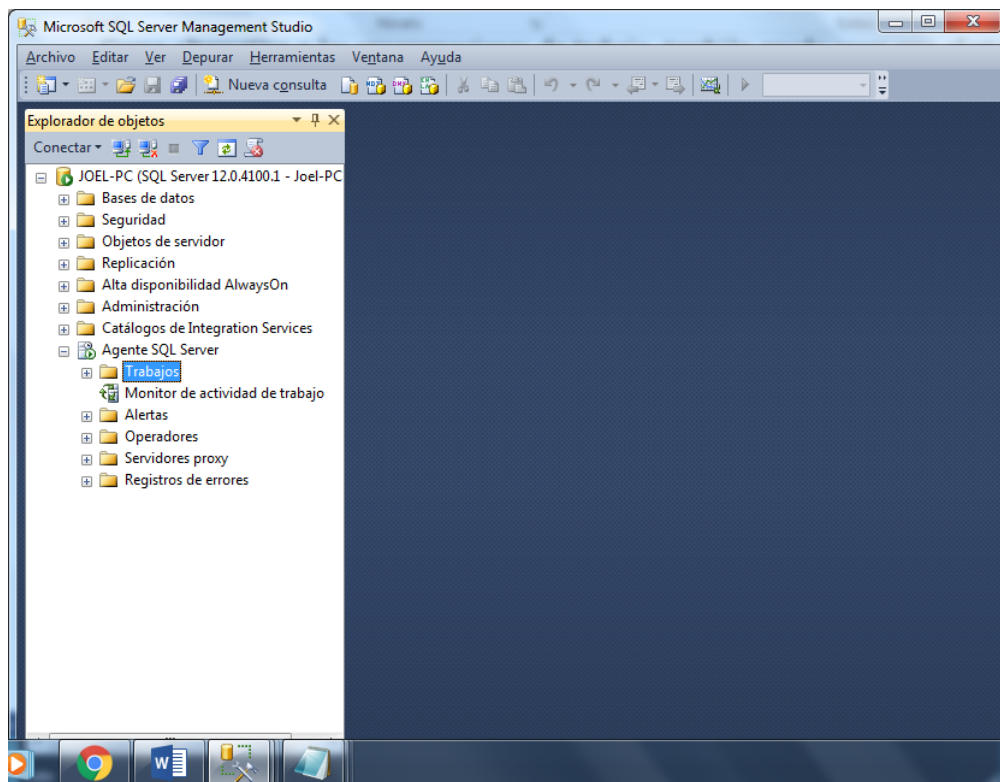
La programación de trabajos administrativos consiste en definir las condiciones que provocan que se empiecen a ejecutar. Se puede programar cualquier tipo de trabajo. Varios trabajos pueden utilizar la misma programación. Los usuarios pueden adjuntar y separar programaciones de los trabajos.

Por ejemplo, puede programar un trabajo para que se ejecute en los casos siguientes:

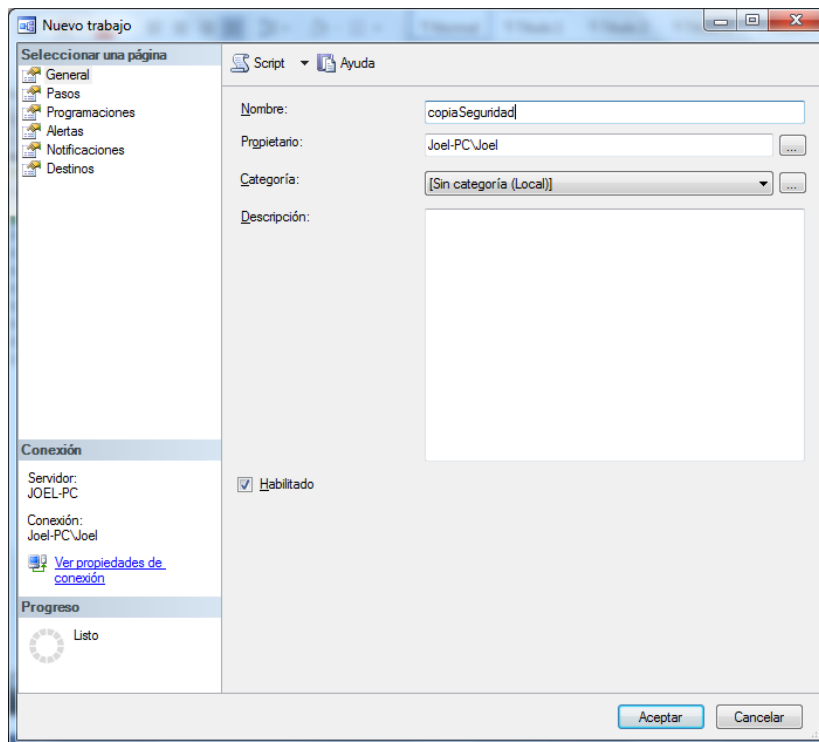
- Cuando se inicia el Agente SQL Server.
- Cuando el uso de la CPU del equipo se encuentre en un nivel que se haya definido como inactivo.
- Una vez, a una hora y una fecha específicas.
- Periódicamente.

Como alternativa a las programaciones de trabajo, también puede crear una alerta que responda a un evento ejecutando un trabajo.

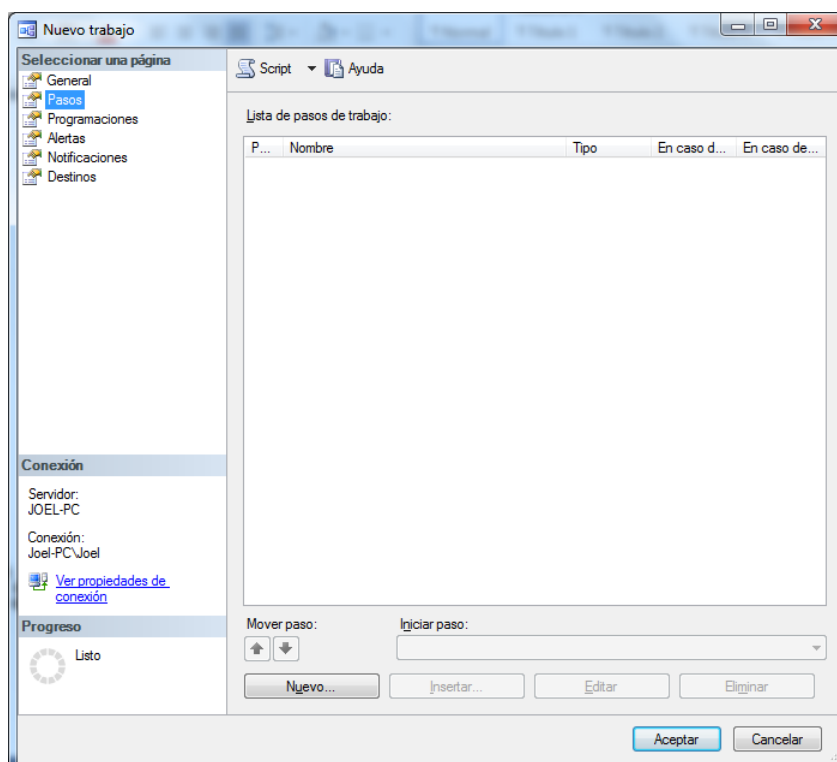
### Ejemplo de creación de Trabajos



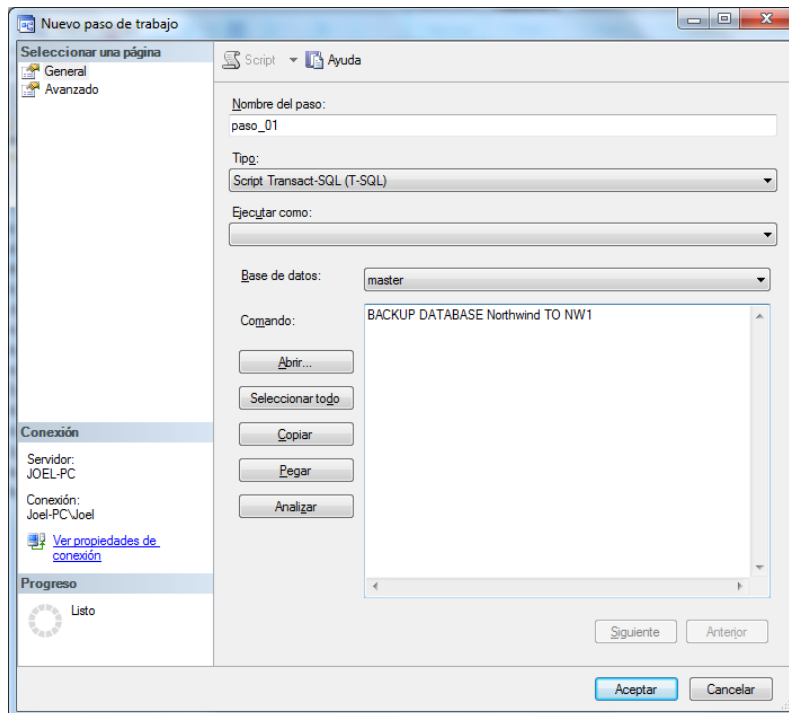
En la ventana Nuevo trabajo, ingresar el Nombre del trabajo.



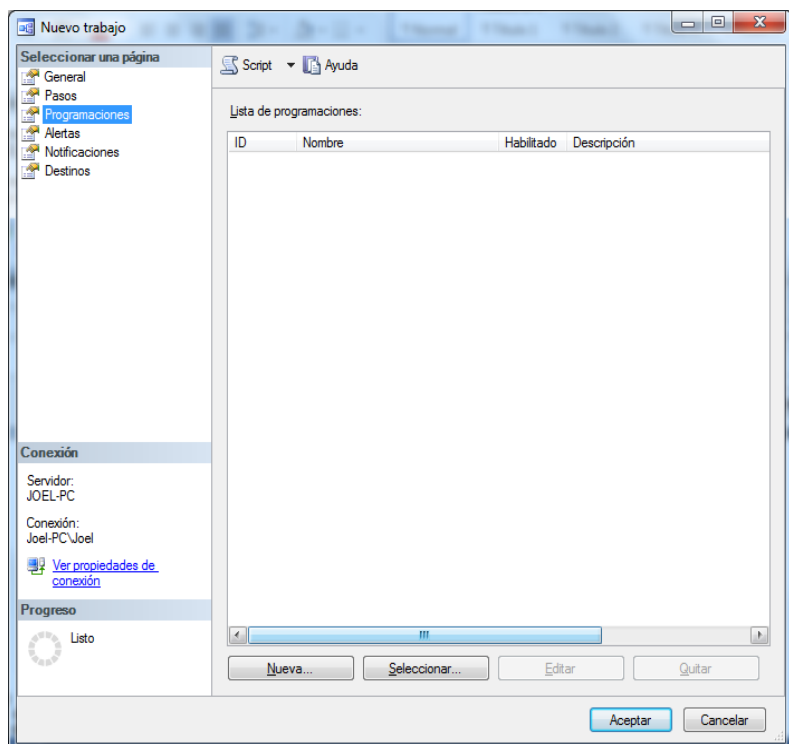
En la ventana Nuevo trabajo, seleccionar la pagina Pasos, Nuevo pasos



En la ventana Nuevo pasos , Ingresar el nombre del paso y el comando al ejecutar.



Seleccionar la pagina Programaciones, dar un clic en Nueva.



En la ficha Nueva Programación, ingresar el nombre de la programación e indicar las opciones de la programación.

The screenshot shows the 'Nueva programación de trabajo' (New Work Scheduling) dialog box. It contains the following fields and options:

- Nombre:** A text box containing 'prog\_01'.
- Trabajos en programación:** A button.
- Tipo de programación:** A dropdown menu set to 'Periódica'.
- Habilitado:** A checked checkbox.
- Única repetición:** A section with 'Fecha:' (07/12/2015) and 'Hora:' (9:42:42).
- Frecuencia:** A section with 'Sucede:' (Semanal), 'Se repite cada:' (1), and 'semanas, el'. Below this are checkboxes for days of the week: Lunes, Miércoles, Viernes, Sábado, Martes, Jueves, and Domingo (checked).
- Frecuencia diaria:** A section with radio buttons for 'Sucede una vez a la(s):' (0:00:00) and 'Sucede cada:' (1 horas). It also has 'A partir de:' (0:00:00) and 'Finaliza:' (23:59:59) time pickers.
- Duración:** A section with 'Fecha de inicio:' (07/12/2015) and radio buttons for 'Fecha de finalización:' (07/12/2015) and 'Sin fecha de finalización:' (selected).
- Resumen:** A section with a 'Descripción:' text box containing 'Sucede el domingo de cada semana a las 0:00:00. Se utilizará la programación que empieza el'.
- Buttons:** 'Aceptar', 'Cancelar', and 'Ayuda' at the bottom.

En la pagina Notificaciones realizar los siguientes cambios.

The screenshot shows the 'Nuevo trabajo' (New Work) dialog box, specifically the 'Notificaciones' (Notifications) tab. It features a left sidebar with a tree view containing 'General', 'Pasos', 'Programaciones', 'Alertas', 'Notificaciones' (selected), and 'Destinos'. The main area is titled 'Acciones que se van a realizar después de finalizar el trabajo:' and contains several options:

- Enviar correo electrónico:** Checked checkbox, with a dropdown set to 'Claudia' and a dropdown set to 'Si el trabajo no tiene éxito'.
- Enviar a buscapersonas:** Unchecked checkbox, with a dropdown set to 'Si el trabajo no tiene éxito'.
- NET SEND:** Unchecked checkbox, with a dropdown set to 'Si el trabajo no tiene éxito'.
- Escribir en el registro de eventos de aplicación de Windows:** Unchecked checkbox, with a dropdown set to 'Si el trabajo no tiene éxito'.
- Eliminar el trabajo automáticamente:** Unchecked checkbox, with a dropdown set to 'Si el trabajo tiene éxito'.

At the bottom, there are 'Aceptar' and 'Cancelar' buttons. The left sidebar also includes sections for 'Conexión' (Servidor: JOEL-PC, Conexión: Joel-PC\Joel, Ver propiedades de conexión) and 'Progreso' (Listo).

# Capítulo 07

## Integration Services

Microsoft Integration Services es una plataforma para la creación de soluciones empresariales de transformaciones de datos e integración de datos. Integration Services sirve para resolver complejos problemas empresariales mediante la copia o descarga de archivos, el envío de mensajes de correo electrónico como respuesta a eventos, la actualización de almacenamientos de datos, la limpieza y minería de datos, y la administración de objetos y datos de SQL Server. Los paquetes pueden funcionar por separado o conjuntamente con otros paquetes para hacer frente a las complejas necesidades de la empresa. Integration Services puede extraer y transformar datos de muchos orígenes distintos, como archivos de datos XML, archivos planos y orígenes de datos relacionales, y, posteriormente, cargarlos en uno o varios destinos.

Integration Services contiene un variado conjunto de tareas y transformaciones integradas, herramientas para la creación de paquetes y el servicio Integration Services para ejecutar y administrar los paquetes. Las herramientas gráficas de Integration Services se pueden usar para crear soluciones sin escribir una sola línea de código. También se puede programar el amplio modelo de objetos de Integration Services para crear paquetes mediante programación y codificar tareas personalizadas y otros objetos de paquete

### Fundamentos de los paquetes de SSIS

Un paquete es el objeto que implementa la funcionalidad de Integration Services para extraer, transformar y cargar datos. Un paquete se crea utilizando el Diseñador SSIS en BI Development Studio. También puede crear un paquete ejecutando el Asistente para importación y exportación de SQL Server o el Asistente para proyectos de conexiones de Integration Services. Para obtener más información, vea [Diseñar paquetes en Business Intelligence Development Studio](#) y [Crear un paquete ejecutando un asistente](#).

Un paquete básico incluye los elementos siguientes:

#### Elementos de flujo de control

Estos elementos necesarios realizan varias funciones, proporcionan estructura y controlan el orden en el que se ejecutan los elementos. Los principales elementos de flujo de control son las tareas, los contenedores y restricciones de precedencia. Es necesario que haya por lo menos un elemento de flujo de control en un paquete.

Para obtener más información, vea [Elementos de flujo de control](#).

#### Elementos de flujo de datos

Estos elementos opcionales extraen, modifican y cargan datos en los orígenes de datos. Los principales elementos de un flujo de datos son orígenes, transformaciones y destinos. No es necesario que haya en este caso elementos de flujo de datos en el paquete.

### **Tarea Flujo de datos**

La tarea Flujo de datos encapsula el motor de flujo de datos que mueve datos entre orígenes y destinos, y permite al usuario transformar, limpiar y modificar datos a medida que se mueven. Agregar una tarea Flujo de datos a un flujo de control de paquetes permite que el paquete extraiga, transforme y cargue datos.

Un flujo de datos se compone de por lo menos un componente de flujo de datos, pero normalmente es un conjunto de componentes de flujo de datos conectados: orígenes que extraen datos, transformaciones que modifican, enrutan o resumen datos, y destinos que cargan datos. Los componentes están conectados en el flujo de datos mediante rutas. Cada ruta especifica los dos componentes que son el inicio y el final de la ruta. En el tiempo de ejecución, la tarea Flujo de datos genera un plan de ejecución a partir del flujo de datos y el motor de flujo de datos ejecuta el plan. Se puede crear una tarea Flujo de datos que no tenga flujo de datos, pero la tarea solo se ejecuta si incluye por lo menos un flujo de datos.

El siguiente diagrama muestra una tarea Flujo de datos con un flujo de datos.



Una tarea Flujo de datos puede incluir varios flujos de datos. Si una tarea copia varios conjuntos de datos, y si el orden en que los datos se copian no es significativo, puede ser más conveniente incluir varios flujos de datos en la tarea Flujo de datos. Por ejemplo, puede crear cinco flujos de datos, y cada uno de ellos copiaría datos de un archivo plano en una tabla de dimensiones diferente en un esquema de estrella de almacenamiento de datos.

Sin embargo, el motor de flujo de datos determina el orden de ejecución cuando existen varios flujos de datos dentro de una tarea de flujo de datos. Por tanto, cuando el orden es importante, el paquete debe usar varias tareas Flujo de datos, cada una de las cuales contiene un flujo de datos. En ese caso, puede aplicar restricciones de precedencia para controlar el orden de ejecución de las tareas.

El siguiente diagrama muestra una tarea Flujo de datos con varios flujos de datos.



Un paquete puede contener varias tareas Flujo de datos, lo que es frecuente en los paquetes complejos. Por ejemplo, si un paquete requiere que los flujos de datos se



ejecuten en un orden especificado, o que se realicen otras tareas entre los flujos de datos, debe utilizar una tarea Flujo de datos independiente para cada flujo de datos.

La tarea Flujo de datos también administra los flujos de errores. En el tiempo de ejecución, pueden generarse errores de nivel de fila cuando los componentes de flujo de datos convierten datos, realizan una búsqueda o evalúan expresiones. Por ejemplo, una columna de datos con un valor de cadena no puede convertirse en un número entero, o una expresión intenta dividirse por cero. Ambas operaciones generan errores, y las filas que contienen los errores se pueden procesar independientemente mediante un flujo de errores..

Para realizar una inserción masiva de datos de archivos de texto en una base de datos de SQL Server, puede usar la tarea Inserción masiva en lugar de la tarea Flujo de datos y un flujo de datos.