



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®

Instituto Tecnológico de Cd. Cuauhtémoc

**INFORME DEL 1er. PERIODO SABÁTICO:
02/09/2024 al 01/03/2025**

**B.1.2 Elaboración de apuntes de asignatura:
Administración de Base de Datos / SCB-1001 del plan de
estudio Sistemas Computacionales 2010**

PRESENTADO POR: M.A. ENRIQUE GARCÍA GRAJEDA

Cd. Cuauhtémoc Chihuahua, Marzo de 2025



INTRODUCCIÓN

En todos los aspectos de la vida cotidiana, las personas buscamos herramientas que nos ayuden a facilitar nuestras actividades y a hacerlas más llevaderas. Si nos enfocamos en el ámbito organizacional, ya sea que se trate de empresas con fines de lucro o instituciones sin ánimo de lucro, es lógico que tengan que seguir un proceso administrativo que abarque la planeación, la organización, la dirección y el control. En cada una de estas etapas, utilizar herramientas adecuadas es clave para simplificar las tareas y lograr mejores resultados.

Una necesidad que suele ser común en cualquier organización es contar con información confiable y accesible, pues con base en ella se pueden tomar decisiones acertadas que ayuden a alcanzar los objetivos. Al final del día, quienes facilitan este acceso a la información son los conocidos Sistemas de Información, que se convierten en aliados esenciales para el éxito y el buen funcionamiento de cualquier organización.

Las bases de datos son como el corazón de la gestión de información en cualquier empresa o institución. Imagínate que tienes que manejar muchísimos datos todos los días; una base de datos te ayuda a tener todo en orden, como si tuvieras un archivero digital donde cada cosa tiene su lugar y es muy fácil encontrar lo que buscas cuando lo necesitas.

Además, no solo se trata de tener la información guardada, sino también de poder acceder a ella de manera rápida y segura. Gracias a las bases de datos, es posible implementar sistemas que cuiden la confidencialidad de los datos y aseguren que solo las personas autorizadas puedan ver o modificar la información. Así, puedes estar tranquilo de que tus datos están protegidos y en buenas manos.

Otro punto importante es la consistencia. Las bases de datos ayudan a que toda la información esté alineada y actualizada, sin errores ni duplicados, aunque varias personas estén trabajando al mismo tiempo. Esto se logra con reglas y validaciones que mantienen el orden y la calidad de los datos.

Contar con una base de datos bien hecha también te facilita la vida a la hora de tomar decisiones. Puedes generar reportes, hacer análisis y revisar estadísticas de manera sencilla, porque tienes acceso a datos precisos y actualizados. Esto te ayuda a tomar decisiones más acertadas y a planear mejor el futuro de tu organización.

Por si fuera poco, las bases de datos tienen la ventaja de que pueden crecer junto contigo. Si tu negocio se expande, la base de datos puede adaptarse y manejar más información sin que tengas que cambiar todo el sistema. Es flexible y escalable, lo que te permite estar preparado para cualquier cambio.

Esta asignatura aporta al perfil del Ingeniero en Sistemas Computacionales las siguientes habilidades:

- Coordinar y participar en equipos multidisciplinarios para la aplicación de soluciones innovadoras en diferentes contextos.
- Diseñar, implementar y administrar bases de datos optimizando los recursos disponibles, conforme a las normas vigentes de manejo y seguridad de la información.
- Desarrollar y administrar software para apoyar la productividad y competitividad de las organizaciones cumpliendo con estándares de calidad.

Actualmente la información es un activo importante en las organizaciones, ya que si posee las características necesarias apoya a las personas en la eficiente toma de decisiones, por lo que se requiere que inviertan en infraestructura que les permita gestionar y resguardar dicho activo, pero también necesitan de profesionistas especializados que sean capaces de administrarlo.

Esta asignatura provee al estudiante con la capacidad para integrar eficientemente la infraestructura existente en una organización y Sistemas Gestores de Base de Datos con el propósito de apoyar la toma de decisiones.

En esta asignatura se abordan las bases teóricas y prácticas referentes a la administración de bases de datos desde la instalación, configuración, mantenimiento, seguridad de acceso y datos, monitoreo del desempeño, así como, métodos de respaldo y recuperación de un SGBD; aplica conocimientos de otras asignaturas, tales como: Taller de Base de Datos, Fundamentos de Base de Datos, Redes de Computadoras, Fundamentos de Ingeniería del Software y Sistemas Operativos.

ÍNDICE

Tema 1 Perspectiva de la administración de base de datos	9
1.1. Administrador de Base de Datos (DBA)	9
1.2. Análisis de los manejadores de bases de datos	11
1.3. Consideraciones para elegir un SGBD	13
1.4. Nuevas tecnologías y aplicaciones de los sistemas de bases de datos	16
Tema 2 Arquitectura e instalación del SGBD	21
2.1 Estructura de Memoria y Procesos de la Instancia de SQL Server.....	21
2.2 Estructura física de la base de datos	22
2.3 Requerimientos para instalación.....	23
2.4 Instalación del SGBD en modo transaccional	25
2.5 Variables de Ambiente y archivos importantes para instalación.	26
2.6 Procedimiento general de instalación	28
2.8 Comandos generales de alta y baja del SGBD	33
Tema 3: Configuración y Administración del Espacio en Disco.....	38
3.1 Definición de espacio de almacenamiento	38
3.2. Definición y creación del espacio asignado para cada base de datos	39
3.3 Asignación de cuotas de espacio para usuarios	44
3.4 Espacios para objetos de la base de datos.....	46
3.5 Roles.....	46
Tema 4 Operación y Mantenimiento	55
4.1 Archivos log del SGBD.....	55
4.3 Índices, reorganización y reconstrucción	56
Tema 5 Seguridad	62
5.1 Espejeo (mirroring).....	62
5.2 Réplica (replication).	72
5.3 Métodos de respaldo de un SGBD.....	93
5.4 Métodos de recuperación de un SGBD.....	100
5.5 Migración de la Base de Datos en sql server	102
Tema 6 Monitoreo y auditoría.....	107
6.1 Monitoreo de la Base de Datos	107

6.2 Auditoría de la Base de Datos.....	108
Bibliografía	112

ÍNDICE DE FIGURAS

Figura 1 Pantalla de Instalación Principal, donde realizaremos la nueva instalación independiente.....	28
Figura 2 Verificación de Reglas auxiliares de Instalación.....	28
Figura 3 Características a Instalar.....	29
Figura 4 Validación de Reglas de Instalación.....	29
Figura 5 Requisitos de Espacio en Disco Duro	30
Figura 6 Informe de Errores	30
Figura 7 Reglas de Configuración	31
Figura 8 Progreso de Instalación.....	31
Figura 9 Operación Terminada.....	32
Figura 10 Microsoft Sql Server Management Studio	32
Figura 11 Disk Usage.....	41
Figura 12 Reporte	42
Figura 13 Script.....	42
Figura 14 Reporte de uso de disco	43
Figura 15 Propiedades de la base de datos	43
Figura 16 Disk Usage modificado.....	44
Figura 17 Roles	49
Figura 18 Nuevo rol de Base de Datos.....	49
Figura 19 Definición de rol.....	50
Figura 20 tipo de objetos	51
Figura 21 Selección de BD.....	51
Figura 22 Rol de acceso.....	52
Figura 23 Usuarios	63
Figura 24 Instancias	64
Figura 25 Instancias creadas	64
Figura 26 Respaldo	65
Figura 27 Restaurar archivos	65
Figura 28 BD completa.....	66
Figura 29 Espejo de BD	66
Figura 30 Servidor testigo	67
Figura 31 Puerto 5022.....	67
Figura 32 Creación de reflejo	68
Figura 33 Finalización de asistente	69
Figura 34 Configurando extremos	69
Figura 35 Proceso de creación.....	70
Figura 36 High safety	71
Figura 37 Prueba de funcionamiento	71
Figura 38 Cuenta Administrador.....	73
Figura 39 Carpeta compartida.....	73
Figura 40 Conexión ambos motores	74

Figura 41 Configuración de distribución	74
Figura 42 Siguiente	75
Figura 43 Selección de primera opción	75
Figura 44 Inicio automático	76
Figura 45 Selección de carpeta compartida	76
Figura 46 Asignación de nombre.....	77
Figura 47 Publicador	77
Figura 48 Distribución.....	78
Figura 49 Finalización de configuración	78
Figura 50 Nueva publicación	79
Figura 51 Asistente de configuración	79
Figura 52 Selección de BD.....	80
Figura 53 Publicación de mezcla.....	80
Figura 54 Sql 2008	81
Figura 55 Selección de tablas	81
Figura 56 Continuación	82
Figura 57 Agregar filtros	82
Figura 58 Replica	83
Figura 59 Cuenta de windows	83
Figura 60 Maquina principal	84
Figura 61 Continuación del proceso	84
Figura 62 Crear publicación	85
Figura 63 Finalización de publicación	85
Figura 64 Pantalla de confirmación	86
Figura 65 Verificación de configuración ambas maquinas	86
Figura 66 Nueva suscripción	87
Figura 67 Asistente	87
Figura 68 Maquina principal y selección de publicación anterior.....	88
Figura 69 Ejecutar cada agente en su suscriptor	88
Figura 70 Backup anterior	89
Figura 71 Windows distribuidor	89
Figura 72 Ejecutar continuamente.....	90
Figura 73Iniciación de suscripción	90
Figura 74 Tipo cliente.....	91
Figura 75 Crear suscripción	91
Figura 76 Finalización de suscripción	92
Figura 77 Verificación de creación	92
Figura 78 Nuevo trabajo	95
Figura 79 Nombre de trabajo.....	95
Figura 80 Comando de respaldo	96
Figura 81 Nombre paso 1.....	96
Figura 82 Nueva programación	97
Figura 83 Elección de día.....	97

Figura 84 Plan de mantenimiento.....	98
Figura 85 Nombre de plan de mantenimiento	98
Figura 86 Tarea Copia de seguridad.....	99
Figura 87 Doble clic.....	99
Figura 88 Selección de ubicación y BD	100
Figura 89 Ubicación	100

Tema 1 Perspectiva de la administración de base de datos

Comprende las actividades de la administración de bases de datos, identifica las funciones del DBA y las características de los diferentes SGBD con el fin de dimensionar su importancia en las organizaciones.

1.1. Administrador de Base de Datos (DBA)

¿Qué es un Administrador de Base de Datos (DBA)?

Un Administrador de Base de Datos, o DBA, es la persona que se encarga de cuidar y administrar las bases de datos dentro de una empresa. Su trabajo consiste en asegurarse de que toda la información esté siempre segura, disponible y bien organizada, para que quienes la necesiten puedan acceder a ella sin problemas. En pocas palabras, el DBA es como el guardián y el encargado de que todo funcione correctamente en el mundo de los datos, que hoy en día son clave para cualquier organización.

¿Cuáles son las responsabilidades de un DBA?

Las responsabilidades de un DBA son muy variadas y van desde poner en marcha la base de datos hasta estar al pendiente de su buen funcionamiento día a día. Aquí te cuento, de manera sencilla, lo que suele hacer:

- **Diseño y desarrollo:** El DBA planea y arma la estructura de las bases de datos, crea las tablas, índices y todo lo necesario para organizar los datos. Además, se encarga de establecer cómo se relacionan unos datos con otros, como si estuviera armando un rompecabezas que tiene que quedar perfecto.
- **Mantenimiento:** Es como el “mecánico” de las bases de datos. Realiza respaldos periódicos (por si las moscas), restaura información si hay alguna falla, revisa que todo corra rápido y sin trabas, y aplica actualizaciones o parches para que el sistema esté al día.
- **Seguridad:** Protege los datos como si fueran un tesoro. Implementa medidas para evitar que alguien acceda sin permiso, vigila quién entra y sale, y está atento a cualquier amenaza que pueda poner en riesgo la información.
- **Solución de problemas:** Cuando algo no funciona o hay algún error, el DBA es quien investiga, encuentra la causa y soluciona el problema. También se encarga de mejorar el rendimiento si nota que la base de datos anda lenta o tiene “cuellos de botella”.
- **Documentación:** Lleva un registro detallado de cómo está organizada la base de datos, lo que se ha hecho y los cambios realizados, para que todo esté claro y cualquiera pueda entenderlo si necesita trabajar con ella después.

El DBA es el encargado de que la información esté bien organizada, protegida y siempre disponible para quienes la necesitan dentro de la empresa.

¿Qué habilidades necesita un DBA?

Para ser un buen DBA, no basta con saber mucho de tecnología; también hace falta tener ciertas habilidades personales. Mira, te explico cómo se combinan estas dos partes:

En cuanto a lo técnico:

El DBA debe tener “colmillo” en varios lenguajes de bases de datos, sobre todo en SQL, que es como el idioma principal para trabajar con la información. También es útil conocer diferentes sistemas como MySQL, PostgreSQL, Oracle o SQL Server, porque cada empresa usa lo que le conviene. Además, debe entender bien cómo funcionan los sistemas operativos, las redes y el almacenamiento, para que todo esté conectado y funcionando de maravilla. Saber hacer scripts y automatizar tareas le ayuda a ahorrarse tiempo y evitar errores, como quien pone el auto en “piloto automático” para las tareas repetitivas.

En lo personal:

Un DBA necesita tener buen ojo para analizar y resolver problemas, como un detective que busca pistas cuando algo no cuadra. También debe saber comunicarse bien, porque todo el tiempo habla con usuarios y otros equipos para entender lo que necesitan y explicar soluciones. Tener iniciativa y estar enfocado a lograr resultados es clave; se requiere actitud para buscar siempre mejorar las cosas y adelantarse a los problemas antes de que se vuelvan un dolor de cabeza.

¿Por qué es importante la figura del DBA?

Las bases de datos son, literalmente, el motor que hace funcionar a las aplicaciones en las empresas. Aquí es donde el papel del DBA se vuelve fundamental, porque es quien se encarga de que todo marche bien, como el mecánico que cuida el carro para que nunca te deje tirado. Un buen DBA se asegura de varias cosas:

- Que los datos estén seguros: Protege la información confidencial de la empresa, evitando que caiga en manos equivocadas.
- Que los datos siempre estén disponibles: Se las ingenia para que los usuarios puedan consultar lo que necesitan justo cuando lo requieren, sin retrasos ni caídas.
- Que los datos sean correctos: Cuida que la información esté bien, sin errores ni confusiones, para que todo cuadre y se puedan tomar buenas decisiones.
- Que los sistemas trabajen rápido y sin trabas: Está pendiente de que las bases de datos no se “atoren” y todo funcione ágil, como debe ser.

En pocas palabras, el DBA es como el guardián que mantiene el orden y el buen funcionamiento en el mundo de los datos. Gracias a su trabajo, la empresa puede

seguir operando sin problemas y tomar decisiones acertadas basadas en información confiable.

1.2. Análisis de los manejadores de bases de datos

Análisis de los Manejadores de Bases de Datos (SGBD)

Cuando hablamos de sistemas de gestión de bases de datos (SGBD), no se trata solo de tecnología; más bien, es como elegir la herramienta adecuada para que todo funcione bien en el día a día. Hay que pensar en varias cosas antes de decidirse por uno, como qué tan rápido y seguro es, si aguanta el ritmo cuando la empresa crece, y por supuesto, cuánto cuesta mantenerlo.

Tipos de SGBD y sus Características

Los SGBD vienen en diferentes “sabores”, dependiendo de cómo manejan la información. Aquí te explico de manera sencilla los principales tipos:

- **Relacionales:** Imagina que los datos se guardan en tablas, como hojas de cálculo, y se conectan entre sí por medio de claves. Algunos ejemplos son MySQL, PostgreSQL, Oracle Database y SQL Server. Lo bueno de estos sistemas es que son fáciles de usar, siguen reglas conocidas (SQL) y llevan años funcionando, así que son confiables. ¿La parte no tan padre? A veces se quedan cortos cuando tienes que almacenar datos muy diferentes o poco estructurados.
- **NoSQL:** Estos son como los todoterreno, pensados para manejar montañas de datos que no siempre tienen forma de tabla. Ejemplos famosos son MongoDB, Cassandra y Redis. Lo mejor es que son súper flexibles y pueden crecer rápido; además, son rapidísimos para leer y escribir datos. Pero ojo, aún están madurando y no siempre sirven para operaciones muy complejas.
- **Híbridos:** Aquí tienes lo mejor de ambos mundos: mezclan lo tradicional con lo nuevo. Algunos, como VoltDB y Amazon Aurora, permiten usar tablas y también datos más libres. Son ideales si buscas flexibilidad, escalabilidad y además necesitas que tus transacciones sean seguras y rápidas.

Elegir un SGBD es como buscar el compañero perfecto para tus datos: uno que se adapte a tus necesidades, que te dé confianza y que no te complique la vida, para que puedas concentrarte en lo importante.

Factores a Considerar al Elegir un SGBD

Modelo de datos: ¿Qué tipo de información vas a guardar? Pregúntate si tus datos necesitan estar súper ordenados, como una lista de invitados, o si pueden ser más libres, como comentarios en redes sociales.

Rendimiento: ¿Qué tan rápido necesitas que se hagan las búsquedas y los registros? Si tu negocio depende de que todo corra en friega, busca un SGBD que responda sin tardanzas.

Escalabilidad: Piensa si tu sistema debe crecer fácil, ya sea agregando más computadoras (escala horizontal) o mejorando las que ya tienes (escala vertical).

Seguridad: ¿Tus datos son delicados o confidenciales? Si es así, elige un sistema que te ayude a protegerlos bien y que no se ande con rodeos en temas de seguridad.

Costo: ¿Cuánto estás dispuesto a invertir? Considera no solo el precio de la licencia, sino también lo que vas a gastar en computadoras y en el mantenimiento.

Facilidad de uso: ¿Qué tan sencillo te resulta aprender y manejar el sistema? Si no quieres complicarte la vida, busca uno que tenga buena fama por ser fácil de usar.

Soporte: ¿Qué tan fácil es encontrar ayuda si algo sale mal? Fíjate si hay soporte técnico disponible, ya sea en español o por medio de comunidades en línea.

Características adicionales: ¿Necesitas que el sistema pueda hacer copias de respaldo, funcionar en varios lugares al mismo tiempo, o conectarse con otras aplicaciones? Checa bien estas funciones para no quedarte corto más adelante.

Comparativa de SGBD Populares

Característica	MySQL	PostgreSQL	Oracle Database	SQL Server	MongoDB
Modelo de datos	Relacional	Relacional	Relacional	Relacional	Documental
Licencia	Comercial	Open source	Comercial	Comercial	Open source
Rendimiento	Alto	Alto	Muy alto	Muy alto	Muy alto para lectura
Escalabilidad	Buena	Muy buena	Excelente	Excelente	Excelente
Seguridad	Buena	Muy buena	Excelente	Excelente	Buena
Facilidad de uso	Buena	Buena	Moderada	Moderada	Buena

Tendencias Actuales

Hoy en día, migrar bases de datos a la nube se ha vuelto algo muy común, ya que servicios como AWS, Azure y GCP ofrecen muchas ventajas en flexibilidad y escalabilidad. Además, existen opciones de bases de datos sin servidor, como Amazon DynamoDB o Firebase Realtime Database, que permiten olvidarse de la gestión de servidores y enfocarse en el desarrollo.

Por otro lado, GraphQL se ha popularizado como un lenguaje de consulta para APIs, ya que ayuda a que los clientes obtengan exactamente la información que necesitan, sin pedir de más ni de menos. En aplicaciones donde la velocidad es clave, como chats o plataformas de streaming, las bases de datos en tiempo real (por ejemplo, Redis o Apache Kafka) son ideales porque garantizan baja latencia y alta disponibilidad.

Al final, elegir el sistema de gestión de bases de datos adecuado depende mucho de lo que busca tu proyecto. No existe una solución única que sea la mejor para todos; lo importante es analizar bien las necesidades de la aplicación y tomar la decisión que mejor se adapte a ellas.

1.3. Consideraciones para elegir un SGBD

Consideraciones Clave al Elegir un Sistema de Gestión de Bases de Datos (SGBD)

La elección del SGBD adecuado es una decisión crucial para cualquier proyecto de desarrollo de software, ya que impacta directamente en el rendimiento, la escalabilidad y la seguridad de la aplicación. A continuación, se presentan algunas de las consideraciones más importantes a tener en cuenta:

1. Tipo de Datos y Modelo de Datos:

Antes de decidirte por un sistema de gestión de bases de datos, es fundamental que te preguntes cómo son los datos con los que vas a trabajar. ¿Tus datos están perfectamente ordenados en tablas, o más bien tienen una estructura flexible o incluso caótica? También considera si entre tus datos hay relaciones complicadas, como cuando varias tablas se conectan entre sí y dependen unas de otras. Por último, piensa en el tamaño de la base de datos: ¿es algo pequeño que apenas va a crecer, o esperas que con el tiempo maneje una cantidad enorme de información? Todas estas preguntas te ayudarán a elegir la herramienta más adecuada para tu proyecto.

2. Requisitos de Rendimiento:

Velocidad de consulta: ¿Qué tan rápido necesitas que se obtenga la información de tu base de datos? Por ejemplo, si tu aplicación maneja muchos usuarios al mismo tiempo, como una tienda en línea o una red social, es fundamental que las consultas sean casi instantáneas para no desesperar a los clientes.

Transacciones: ¿Tu proyecto requiere operaciones complejas donde no se puede perder ni un dato, como en bancos o sistemas de reservas? En estos casos, es clave que el sistema ofrezca transacciones seguras y cumpla con los principios ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad) para evitar errores y mantener todo bajo control.

Concurrencia: ¿Cuántas personas o sistemas van a estar accediendo a la base de datos al mismo tiempo? Si esperas cientos o miles de conexiones simultáneas, necesitas un SGBD que no se caiga ni se vuelva lento, garantizando que todos puedan trabajar sin interrupciones.

3. Escalabilidad:

Crecimiento de datos: Piensa si tu base de datos va a crecer mucho con el tiempo. ¿Será necesario que el sistema se adapte fácilmente cuando tengas más información o usuarios? También considera si necesitas que el crecimiento sea "hacia arriba" (mejorando el mismo servidor) o "hacia los lados" (agregando más servidores cuando lo requieras).

Escalabilidad geográfica: Pregúntate si tu aplicación va a funcionar en distintos lugares del país o incluso del mundo. Si es así, tal vez requieras que la base de datos se pueda replicar en varias ubicaciones geográficas, para que los usuarios siempre tengan acceso rápido y confiable sin importar desde dónde se conecten.

4. Seguridad:

Confidencialidad: ¿Qué tan importante es mantener protegidos los datos sensibles? Piensa si tu información debe estar resguardada para que solo las personas autorizadas puedan verla y no caiga en manos equivocadas.

Integridad: ¿Cómo te aseguras de que los datos no se modifiquen por accidente o de forma maliciosa? Es fundamental que la información se mantenga completa y sin alteraciones no deseadas, para que siempre puedas confiar en lo que tienes almacenado.

Disponibilidad: ¿Tu proyecto necesita que la base de datos esté siempre accesible, incluso si hay fallas o algún desastre? Considera si requieres mecanismos que garanticen que los datos estén disponibles en todo momento y puedas recuperarlos rápidamente en caso de algún problema.

5. Facilidad de Uso:

Lenguaje de consulta: ¿Tú o tu equipo tienen experiencia usando SQL o el lenguaje específico del sistema de base de datos? Esto puede ahorrarte muchos dolores de cabeza, porque si ya saben manejarlo, la curva de aprendizaje será mucho más suave.

Herramientas de administración: ¿La base de datos trae programas o interfaces amigables para gestionarla? Es mucho más sencillo trabajar con un sistema que tenga herramientas visuales, menús claros y opciones fáciles de usar, en vez de tener que hacer todo desde la línea de comandos.

Documentación: ¿Hay manuales, tutoriales o guías disponibles que realmente se entiendan? Una buena documentación te puede sacar de apuros cuando surgen

dudas y ayuda a resolver problemas rápido, sin tener que estar buscando en foros o inventando soluciones.

6. Costo:

Licencias: Antes de elegir tu base de datos, fíjate bien en cuánto cuesta la licencia. Hay opciones gratuitas, pero otras pueden pedirte una buena lana por usuario o por servidor, así que revisa bien para que no te lleves sorpresas.

Hardware: Pregúntate si vas a necesitar equipos especiales o servidores potentes para que tu base de datos funcione bien. A veces, con una compu sencilla basta, pero en otros casos podrías necesitar invertir en infraestructura más robusta.

Mantenimiento: No olvides considerar cuánto te va a costar mantener y darle soporte a tu base de datos. Esto incluye actualizaciones, respaldos y resolver cualquier bronca que salga en el camino, ya sea con ayuda de expertos o de tu propio equipo.

7. Características Adicionales:

Clustering: Pregúntate si tu proyecto va a necesitar un entorno donde varias computadoras trabajen juntas como un equipo para manejar la base de datos. Esto es útil si buscas repartir la carga de trabajo o asegurarte de que todo siga funcionando, aunque alguna máquina falle.

Replicación: Considera si quieres que los datos se copien automáticamente en varios servidores. Así, si uno deja de funcionar, tus datos siguen disponibles en otro y evitas pérdidas o caídas del sistema.

Geolocalización: ¿Tu aplicación necesita trabajar con mapas, ubicaciones o datos geoespaciales? Si es así, revisa si la base de datos ofrece funciones especiales para manejar este tipo de información, como búsquedas por cercanía o almacenamiento de coordenadas.

Integración con otras herramientas: Piensa si tienes que conectar la base de datos con otros programas o sistemas (por ejemplo, hojas de cálculo, sistemas de análisis o aplicaciones externas). Es importante que el sistema permita integrarse fácilmente para que todo fluya sin complicaciones.

8. Soporte técnico:

Disponibilidad: Antes de decidirte, revisa bien qué tan fácil es conseguir ayuda cuando la necesites. ¿El soporte técnico está disponible las 24 horas o solo en ciertos horarios? Saber esto puede ahorrarte muchos dolores de cabeza si tienes un problema urgente.

Comunidad: También checa si hay una comunidad activa de usuarios. Una comunidad movida puede ser un gran apoyo, porque seguro encuentras foros, grupos o tutoriales donde resolver dudas o compartir experiencias.

Ejemplos de SGBD y sus Casos de Uso:

- MySQL, PostgreSQL: Ideales para aplicaciones web de tamaño medio a grande, aplicaciones empresariales y desarrollo de software libre.
- Oracle Database, SQL Server: Adecuados para grandes empresas que requieren un alto nivel de rendimiento, seguridad y escalabilidad.
- MongoDB, Cassandra: Excelentes para aplicaciones que manejan grandes volúmenes de datos no estructurados o semiestructurados, como análisis de datos y aplicaciones de IoT.
- Redis: Ideal para almacenamiento en caché, sesiones, y aplicaciones que requieren un acceso muy rápido a los datos.

1.4. Nuevas tecnologías y aplicaciones de los sistemas de bases de datos

Las bases de datos han cambiado muchísimo en los últimos años, todo por la enorme cantidad de información que generamos y porque necesitamos herramientas cada vez más especiales para manejarla. Hoy en día existen muchas tecnologías y usos nuevos para los sistemas de bases de datos, y aquí vamos a platicar sobre las más importantes y cómo impactan nuestro día a día.

Nuevas Tecnologías

- Bases de Datos NoSQL:
 - Diseñadas para manejar grandes volúmenes de datos no estructurados o semiestructurados.
 - Ejemplos: MongoDB, Cassandra, Redis.
 - Aplicaciones: Análisis de big data, aplicaciones en tiempo real, almacenamiento de documentos.
- Bases de Datos en la Nube:
 - Ofrecen escalabilidad, alta disponibilidad y facilidad de gestión.
 - Ejemplos: Amazon RDS, Google Cloud SQL, Azure SQL Database.
 - Aplicaciones: Desarrollo de aplicaciones web y móviles, análisis de datos en la nube.
- Gráficas de Conocimiento:
 - Representan información como un grafo, permitiendo modelar relaciones complejas entre entidades.
 - Ejemplos: Neo4j, Amazon Neptune.
 - Aplicaciones: Recomendación de productos, detección de fraudes, análisis de redes sociales.
- Bases de Datos en Memoria:
 - Almacenan datos en la memoria principal para un acceso extremadamente rápido.
 - Ejemplos: Redis, Memcached.

- Aplicaciones: Cachés de datos, sesiones de usuario, sistemas de mensajería en tiempo real.

Nuevas Aplicaciones

- Inteligencia Artificial y Machine Learning:
 - Las bases de datos son fundamentales para entrenar modelos de machine learning y almacenar los resultados.
 - Aplicaciones: Recomendación de productos, análisis de sentimientos, visión por computadora.
- Internet de las Cosas (IoT):
 - Los datos generados por dispositivos IoT se almacenan y analizan en bases de datos.
 - Aplicaciones: Monitoreo de sensores, análisis de patrones de uso, mantenimiento predictivo.
- Blockchain:
 - Las bases de datos distribuidas y seguras son esenciales para las aplicaciones blockchain.
 - Aplicaciones: Criptomonedas, gestión de la cadena de suministro, registros médicos.
- Realidad Virtual y Aumentada:
 - Las bases de datos almacenan los modelos 3D y los datos necesarios para crear experiencias inmersivas.
- Análisis de Big Data:
 - Las bases de datos de gran escala permiten procesar y analizar grandes volúmenes de datos para obtener insights valiosos.

Tendencias Futuras

Las bases de datos autónomas son sistemas inteligentes que se encargan de configurarse, adaptarse y corregir problemas por sí mismas, lo que facilita mucho el trabajo de quienes las gestionan. Por otro lado, las bases de datos multimodelo tienen la gran ventaja de poder almacenar y manejar varios tipos de información en un solo lugar, haciendo más sencillo el acceso y la organización de los datos.

En cuanto a las bases de datos quantum, estas aprovechan la potencia de la computación cuántica para resolver desafíos que antes eran demasiado complejos para las tecnologías tradicionales. Además, la integración de las bases de datos con otras innovaciones, como el edge computing y el 5G, permite procesar y transmitir datos de manera más rápida y eficiente, impulsando el desarrollo de nuevas aplicaciones y servicios.

Desafíos y Consideraciones

Seguridad: Es fundamental proteger toda la información sensible que guardamos en las bases de datos, para evitar que personas no autorizadas accedan a datos importantes como contraseñas, números de tarjeta o historiales médicos.

Privacidad: Además de cuidar los datos, es necesario asegurarse de que se cumplan todas las leyes y regulaciones sobre privacidad, como el GDPR y el CCPA, para que la información personal de los usuarios esté siempre protegida y se maneje de manera responsable.

Escalabilidad: Hoy en día, la cantidad de datos crece a un ritmo acelerado, por lo que las bases de datos deben ser capaces de adaptarse y responder a ese crecimiento sin perder rendimiento ni afectar la experiencia de quienes las usan.

Gestión: La administración de bases de datos complejas puede ser todo un reto, por eso es importante contar con herramientas y procesos que hagan más fácil y eficiente el trabajo diario de quienes se encargan de su gestión.

Las nuevas tecnologías y aplicaciones están impulsando la innovación en diversos sectores, desde la salud hasta las finanzas.

CUESTIONARIO TEMA 1

1. ¿Cuál de los siguientes NO es una responsabilidad típica de un Administrador de Base de Datos (DBA)?

- a) Diseñar la estructura de las bases de datos y crear tablas e índices.
- b) Desarrollar el software de aplicaciones externas para usuarios finales.
- c) Realizar respaldos de periódicos y aplicar actualizaciones para mantener el sistema al día.

2. ¿Cuál es una habilidad técnica fundamental que debe poseer un DBA?

- a) Conocimiento avanzado en lenguajes de programación orientados a objetos como Java o C++.
- b) Dominio del lenguaje SQL y comprensión de sistemas operativos y redes.
- c) Experiencia en diseño gráfico para mejorar la interfaz de usuario.

3. ¿Por qué es crucial la figura del DBA en una empresa?

- a) Porque se encarga exclusivamente de la programación de aplicaciones móviles.
- b) Porque garantiza la seguridad, disponibilidad, integridad y rendimiento óptimo de los datos.
- c) Porque supervisar el marketing digital y las ventas en línea.

4. ¿Cuál es la principal diferencia entre un SGBD relacional y uno NoSQL?

- a) Los SGBD relacionales almacenan datos en tablas con relaciones definidas, mientras que los NoSQL manejan datos no estructurados o semiestructurados.

- b) Los SGBD NoSQL solo se usan para bases de datos pequeñas, mientras que los relacionales son para grandes volúmenes.
- c) Los SGBD relacionales no permiten consultas complejas, a diferencia de los NoSQL.

5. ¿Qué factor NO se considera al elegir un SGBD?

- a) Facilidad de uso y disponibilidad de soporte técnico.
- b) Popularidad del sistema en redes sociales.
- c) Escalabilidad y costo total del sistema.

6. ¿Cuál de las siguientes bases de datos se considera híbrida, combinando características relacionales y no relacionales?

- a) MongoDB.
- b) Auroras amazónicas.
- c) Redis.

7. ¿Qué característica es esencial para un SGBD que maneja transacciones complejas en un banco?

- a) Capacidad de replicación automática sin garantía de integridad.
- b) Cumplimiento de los principios ACID para garantizar transacciones seguras.
- c) Gestión de bases de datos exclusivamente en la nube.

8. ¿Cuál es una clave de ventaja que ofrecen las bases de datos en memoria como Redis?

- a) Altísima velocidad de acceso a datos para aplicaciones en tiempo real.
- b) Escalabilidad horizontal ilimitada sin necesidad de hardware adicional.
- c) Almacenamiento permanente de grandes volúmenes de datos históricos.

9. Entre las nuevas tecnologías de bases de datos, ¿qué función tienen las bases de datos gráficos de conocimiento?

- a) Modelar relaciones complejas entre entidades usando gráficos.
- b) Almacenar únicamente datos no estructurados sin relaciones definidas.
- c) Optimizar la gestión de bases de datos en memoria.

10. ¿Qué desafío es fundamental para la gestión y uso de bases de datos modernas, considerando la legislación actual?

- a) La creación de interfaces gráficas para usuarios finales.
- b) Cumplir regulaciones de privacidad como GDPR y CCPA para proteger datos personales.

- c) Incrementar la velocidad de consultas sin importar la seguridad de los datos.

Respuestas correctas:

1. b) Desarrollar el software de aplicaciones externas para usuarios finales.b
2. b) Dominio del lenguaje SQL y comprensión de sistemas operativos y redes.
3. b) Porque garantiza la seguridad, disponibilidad, integridad y rendimiento óptimo de los datos.
4. a) Los SGBD relacionales almacenan datos en tablas con relaciones definidas, mientras que los NoSQL manejan datos no estructurados o semiestructurados.
5. b) Popularidad del sistema en redes sociales.
6. b) Auroras amazónicas.
7. b) Cumplimiento de los principios ACID para garantizar transacciones seguras.
8. a) Altísima velocidad de acceso a datos para aplicaciones en tiempo real.
9. a) Modelar relaciones complejas entre entidades usando gráficos.
- 10.b) Cumplir regulaciones de privacidad como GDPR y CCPA para proteger datos personales.

Tema 2 Arquitectura e instalación del SGBD

Comprende los componentes de la Arquitectura del Manejador de Base de Datos con el fin de identificar las funciones de cada uno.

Instala y configura un SGBD cumpliendo con los requisitos recomendados para su funcionamiento.

2.1 Estructura de Memoria y Procesos de la Instancia de SQL Server

La instancia de SQL Server, que representa una ejecución específica del motor de base de datos, se compone de una compleja estructura de memoria y procesos que trabajan en conjunto para gestionar eficientemente los datos y las consultas.

Área de Datos Global (SGA)

El SGA es una región de memoria compartida que contiene datos e información de control esenciales para la instancia de SQL Server. Esta área se divide en varias subestructuras:

- **Buffer cache:** Almacena las páginas de datos más recientemente utilizadas, reduciendo el número de lecturas físicas al disco.
- **Shared pool:** Contiene objetos compartidos como instrucciones SQL compiladas, planes de ejecución y metadatos de la base de datos.
- **Large page allocations:** Se utiliza para almacenar grandes objetos como índices y tablas de datos.
- **Row cache:** Almacena filas individuales de datos que se acceden con frecuencia.
- **GAIA:** Proporciona servicios de gestión de memoria dinámica.
- **Otras áreas:** Lock manager, procesos de fondo, etc.

Procesos de SQL Server

Los procesos de SQL Server se encargan de ejecutar las tareas de la instancia, como gestionar las conexiones de los usuarios, procesar consultas, escribir datos en el disco, etc. Algunos de los procesos más importantes son:

- **SQL Server (sqlservr.exe):** El proceso principal que controla la instancia.
- **SQL Server Agent:** Ejecuta tareas programadas, como copias de seguridad y trabajos de mantenimiento.
- **Distributed Transaction Coordinator:** Coordina transacciones distribuidas.
- **SQL Server Browser:** Permite a los clientes localizar instancias de SQL Server en la red.
- **SQL Server Integration Services:** Se utiliza para la extracción, transformación y carga de datos (ETL).

Interacción entre Memoria y Procesos

La interacción entre la memoria y los procesos de SQL Server es fundamental para el rendimiento de la base de datos. Los procesos utilizan la memoria del SGA para

almacenar los datos y los resultados intermedios de las consultas. Cuando un proceso necesita acceder a datos que no están en la memoria, se realiza una lectura física al disco.

Optimización de la Memoria:

Para optimizar el rendimiento de SQL Server, es importante ajustar la configuración de la memoria. Algunos factores a considerar incluyen:

- **Tamaño del SGA:** Un SGA demasiado pequeño puede provocar un exceso de lectura/escritura en el disco, mientras que un SGA demasiado grande puede desperdiciar memoria.
- **Distribución de la memoria:** La distribución de la memoria entre las diferentes subestructuras del SGA debe ajustarse según las necesidades de la carga de trabajo.
- **Configuración de los índices:** Los índices adecuados pueden reducir significativamente el número de lecturas físicas al disco.
- **Optimización de las consultas:** Las consultas mal escritas pueden provocar un uso ineficiente de los recursos de la base de datos.

2.2 Estructura física de la base de datos

La estructura física de una base de datos en SQL Server se organiza en una jerarquía de archivos, grupos de archivos y unidades de almacenamiento más pequeñas, como las páginas y las extensiones. Esta organización permite a SQL Server gestionar los datos de forma eficiente.

1. Archivos

Una base de datos de SQL Server se compone de al menos dos archivos principales:

- **Archivos de datos (.mdf, .ndf):** Contienen todos los datos de la base de datos, incluyendo tablas, índices, vistas, etc.
 - **Archivo de datos principal (.mdf):** El punto de partida de la base de datos. Cada base de datos tiene solo un archivo de este tipo.
 - **Archivos de datos secundarios (.ndf):** Archivos adicionales que se usan para distribuir los datos entre múltiples discos. Es útil para mejorar el rendimiento y la gestión de bases de datos muy grandes.
- **Archivo de registro de transacciones (.ldf):** Almacena el **registro de transacciones**, que es una bitácora de todas las transacciones y modificaciones de datos. Es crucial para la recuperación de la base de datos,

ya que permite deshacer o rehacer operaciones en caso de fallo, garantizando la consistencia de los datos.

2. Grupos de Archivos (Filegroups)

Los grupos de archivos son una capa lógica que agrupa uno o más archivos de datos. La finalidad de los grupos de archivos es simplificar la administración y optimizar el rendimiento.

- **Grupo de archivos principal (Primary):** Contiene el archivo de datos principal (.mdf) y, opcionalmente, cualquier archivo secundario no asignado a otro grupo. Es el grupo por defecto de la base de datos.
- **Grupos de archivos definidos por el usuario:** Puedes crear estos grupos para organizar objetos de la base de datos (tablas e índices) y distribuirlos en diferentes discos. Por ejemplo, puedes tener un grupo de archivos para datos de uso frecuente en un disco de alta velocidad y otro para datos históricos en un disco más lento, mejorando así el rendimiento de las consultas.

Los archivos de registro de transacciones (.ldf) **no pueden** ser parte de un grupo de archivos.

3. Unidades de Almacenamiento

Dentro de los archivos, los datos se organizan en unidades de almacenamiento más pequeñas:

- **Páginas:** Es la unidad fundamental de almacenamiento y E/S (Entrada/Salida) en SQL Server. Cada página tiene un tamaño fijo de **8 KB**. SQL Server lee y escribe datos en el disco en unidades de página.
- **Extensiones:** Son la unidad básica para la asignación de espacio. Una extensión consiste en ocho páginas contiguas físicamente, lo que equivale a **64 KB**. Agrupar las páginas en extensiones ayuda a SQL Server a gestionar el espacio de forma más eficiente.

2.3 Requerimientos para instalación.

Para la instalación de SQL Server, basados en la versión más reciente (SQL Server 2022). Es importante tener en cuenta que estos son los requisitos mínimos; para un rendimiento óptimo en entornos de producción, se recomiendan especificaciones más altas.

Requisitos de Hardware

- **Procesador:** Un procesador **x64** con una velocidad de 1.4 GHz como mínimo. Se recomienda 2.0 GHz o superior para un mejor rendimiento.

- **Memoria RAM:**
 - **Mínima:** 2 GB de RAM.
 - **Recomendada:** 4 GB o más. Tene en cuenta que para una carga de trabajo pesada, la memoria es uno de los factores más importantes para el rendimiento. Las versiones Express tienen un límite de 1 GB.
- **Espacio en Disco Duro:**
 - **Mínimo:** 6 GB de espacio disponible.
 - **Recomendado:** Se debe tener en cuenta el espacio para los archivos de la base de datos, los registros de transacciones y los backups, que pueden crecer significativamente.
- **Pantalla:** Se requiere una resolución de 1024 x 768 píxeles.

Requisitos de Software

- **Sistema Operativo:** SQL Server 2022 solo es compatible con sistemas operativos de **64 bits**. Las versiones soportadas incluyen:
 - Windows 10 (versión 1607 o superior).
 - Windows 11.
 - Windows Server 2016 o versiones posteriores (2019, 2022).
- **.NET Framework:** Se requiere la versión **4.7.2** o superior. El instalador de SQL Server generalmente detecta y solicita la instalación de este componente si no está presente.
- **Red:** El sistema operativo debe tener un software de red integrado. Los protocolos de red compatibles son Memoria compartida, Canalizaciones con nombre y TCP/IP.
- **Privilegios de Administrador:** El usuario que realiza la instalación debe tener **privilegios de administrador** en el sistema operativo.

Consideraciones Adicionales

- **Ediciones de SQL Server:** Las distintas ediciones de SQL Server (Express, Standard, Enterprise) tienen diferentes limitaciones y requisitos. Por ejemplo, la versión gratuita **SQL Server Express** tiene limitaciones en el tamaño máximo de la base de datos (10 GB) y en el uso de CPU y RAM.
- **Servicios y Componentes:** Los requisitos de espacio en disco pueden variar dependiendo de los componentes que decidas instalar, como Analysis Services, Reporting Services, Integration Services, etc.

- **Virtualización y Nube:** SQL Server también se puede instalar en entornos virtualizados o en la nube a través de servicios como **Azure SQL Database**, donde Microsoft se encarga de la gestión del hardware y la infraestructura.

2.4 Instalación del SGBD en modo transaccional

La Replicación Transaccional, que es una forma de replicar datos y es crucial en entornos de alta disponibilidad, la instalación y configuración es un proceso específico.

Instalación de un SGBD (SQL Server)

La instalación base de SQL Server se realiza de manera estándar. Solo se requiere seguir el asistente de instalación.

1. **Ejecutar el instalador:** Inicia el archivo setup.exe de la distribución de SQL Server.
2. **Seleccionar "Nueva instalación independiente de SQL Server".**
3. **Seleccionar características:** En la página de selección de características, a marcar "Database Engine Services". Para la replicación transaccional, también marcar "SQL Server Replication".
4. **Configuración de la instancia:** Definir un nombre para la instancia de SQL Server.
5. **Configuración del servicio:** Configurar las cuentas de servicio para SQL Server y SQL Server Agent, asegurar de que tengan permisos adecuados.
6. **Autenticación:** Elegir el modo de autenticación (modo mixto es común para desarrollo y producción).
7. **Finalizar la instalación.**

Configuración de la Replicación Transaccional

Una vez que el SGBD está instalado, el modo transaccional no se "activa" automáticamente; se configura a través de la **replicación transaccional**. Este proceso se lleva a cabo dentro de SQL Server Management Studio (SSMS) y tiene tres componentes principales:

1. **Configurar el Distribuidor:** Es la primera etapa y el corazón de la replicación.
 - En SSMS, hacer clic derecho en el nodo "Replicación" y seleccionar **"Configurar Distribución"**.

- El asistente guiará para designar un servidor como distribuidor, crear una base de datos de distribución y establecer la carpeta de instantáneas donde se almacenarán los archivos de la réplica inicial.
2. **Crear una Publicación:** Define qué datos se van a replicar.
- En el servidor "publicador", hacer clic derecho en "**Publicaciones locales**" y seleccionar "**Nueva publicación**".
 - El asistente permitirá elegir la base de datos y las tablas (llamadas **artículos**) que se desea replicar.
 - Seleccionar "Replicación transaccional" como el tipo de publicación.
 - Configurar las propiedades del "agente de instantáneas" y la seguridad.
3. **Crear una Suscripción:** Conectar el publicador y el suscriptor.
- En la publicación recién creada, hacer clic derecho y seleccionar "**Nuevas suscripciones**".
 - El asistente guiará para elegir el servidor "suscriptor" y la base de datos donde se recibirán los datos replicados.
 - Definir el "agente de distribución" y la programación de la sincronización.

2.5 Variables de Ambiente y archivos importantes para instalación.

Las variables de entorno son un conjunto de valores dinámicos que afectan el comportamiento de los procesos en un sistema operativo. Para la instalación de SQL Server, no se requieren variables de entorno específicas, ya que el instalador se encarga de configurar todo automáticamente. Sin embargo, algunas variables son relevantes para el funcionamiento y la seguridad del sistema en general:

- **TEMP y TMP:** Estas variables definen las ubicaciones para archivos temporales. Asegúrate de que apunten a un disco con suficiente espacio libre para los archivos de instalación.
- **PATH:** Esta variable contiene una lista de directorios donde el sistema operativo busca archivos ejecutables. Después de la instalación de SQL Server, es común que la ruta a las herramientas de línea de comandos, como sqlcmd.exe o las utilidades de copia masiva (bcp.exe), se agreguen a la variable PATH para poder usarlas desde cualquier ubicación en la línea de comandos.

Archivos y Directorios Importantes

Durante la instalación de SQL Server, se crean varios archivos y directorios clave. Conocer su ubicación es vital para la administración, la resolución de problemas y las copias de seguridad.

1. Archivos del Sistema de la Base de Datos:

- **Archivos de datos (.mdf, .ndf):** Contienen los datos de la base de datos, las tablas y los índices. Por defecto, se encuentran en C:\ProgramFiles\MicrosoftSQLServer\MSSQLXX.NombreDeInstancia\MSSQL\DATA. Es una práctica recomendada mover estos archivos a una unidad de disco separada (que no sea la unidad C) para mejorar el rendimiento y la capacidad de recuperación.
- **Archivos de registro (.ldf):** Almacenan el registro de transacciones de la base de datos. Se ubican en el mismo directorio que los archivos de datos.

2. Archivos de Configuración:

- **errorlog:** Es el registro principal de SQL Server. Contiene información sobre el inicio y el apagado del servidor, mensajes de error y eventos críticos. Es un archivo de texto simple y se encuentra en el subdirectorio Log del directorio de datos.
- **Archivos de registro de instalación:** El instalador de SQL Server crea registros detallados de todo el proceso en la carpeta %programfiles%\Microsoft SQL Server\160\Setup Bootstrap\Log. Estos archivos son cruciales para diagnosticar problemas de instalación.

3. Ejecutables y Binarios:

- El motor de la base de datos (sqlservr.exe), junto con otros ejecutables de los servicios, se encuentra en el directorio Binn de la instancia. Ejemplo: C:\ProgramFiles\MicrosoftSQLServer\MSSQLXX.NombreDeInstancia\MSSQL\Binn.

4. Archivos de Base de Datos del Sistema:

- master.mdf y mastlog.ldf: Almacenan toda la información de configuración del sistema de la base de datos y sus metadatos.
- msdb.mdf y msdblog.ldf: Usados por el Agente SQL Server para tareas de programación, replicación y administración de copias de seguridad.

- model.mdf y modellog.ldf: Sirven como plantilla para todas las nuevas bases de datos creadas.
- tempdb.mdf y templog.ldf: Almacenan objetos temporales y resultados intermedios. Se recrean cada vez que se inicia SQL Server y no deben ser modificados.

2.6 Procedimiento general de instalación

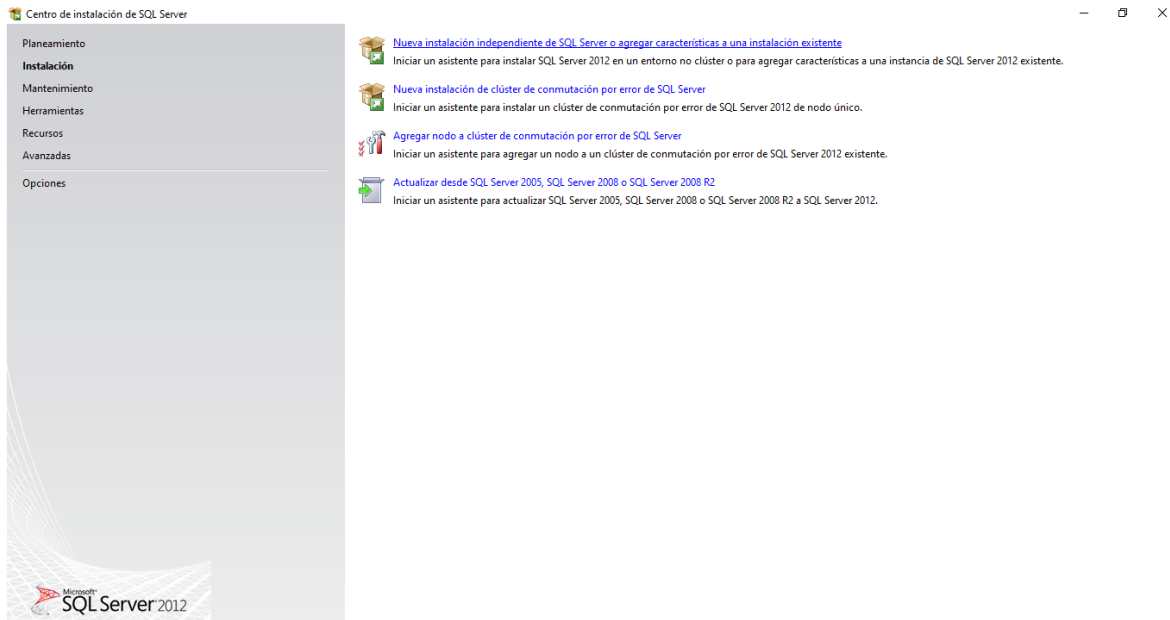


Figura 1 Pantalla de Instalación Principal, donde realizaremos la nueva instalación independiente.

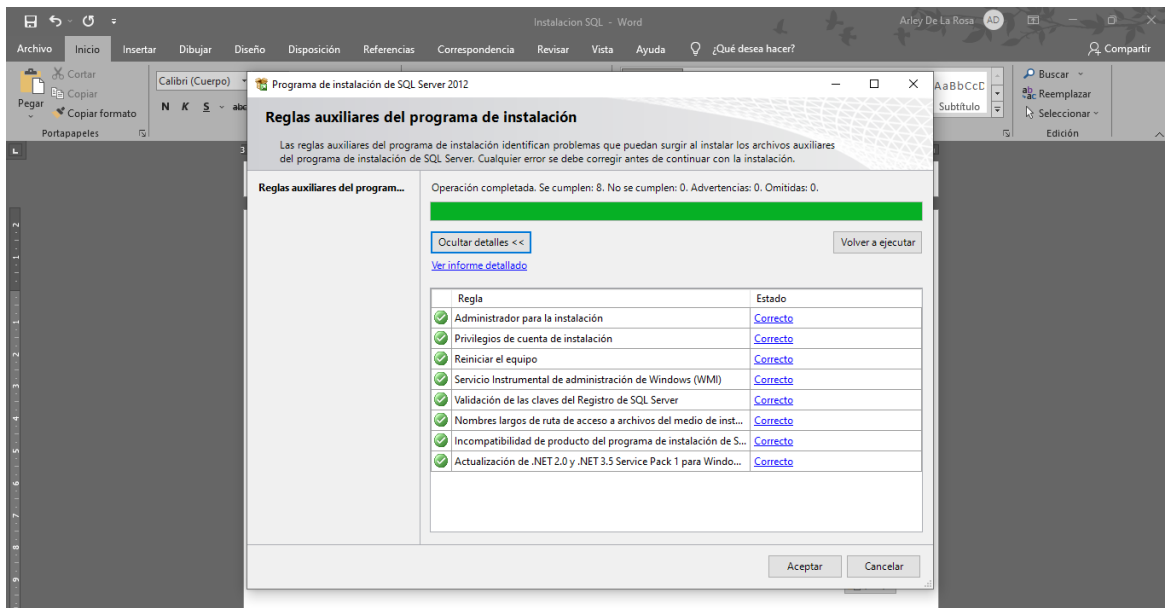


Figura 2 Verificación de Reglas auxiliares de Instalación.

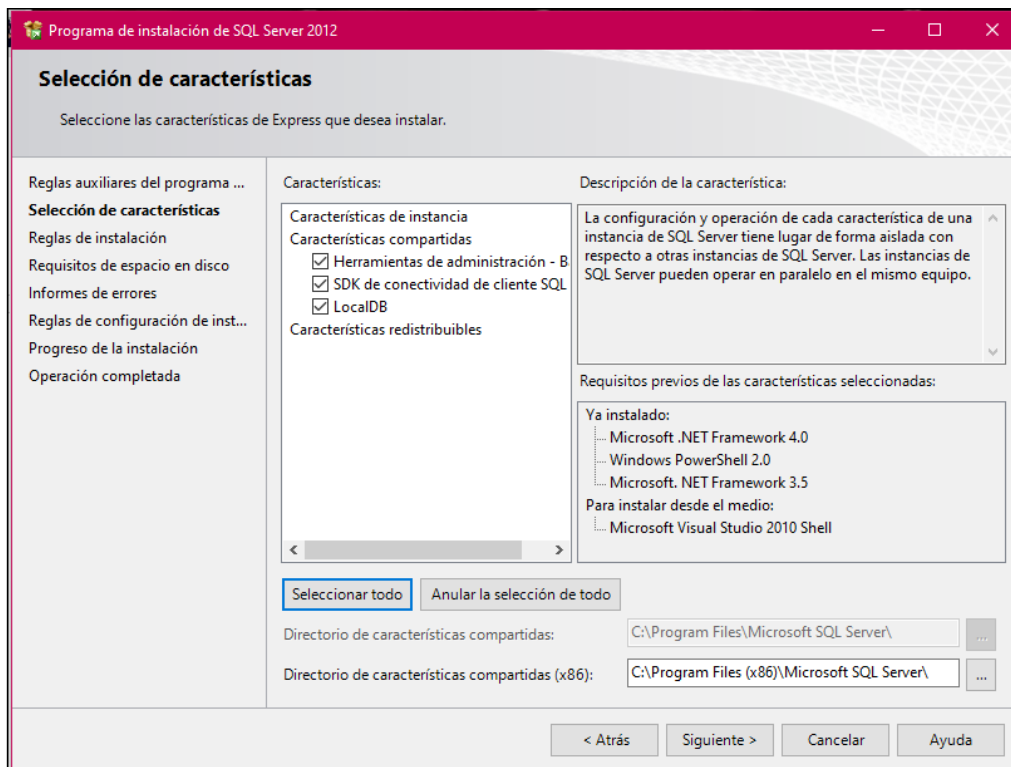


Figura 3 Características a Instalar

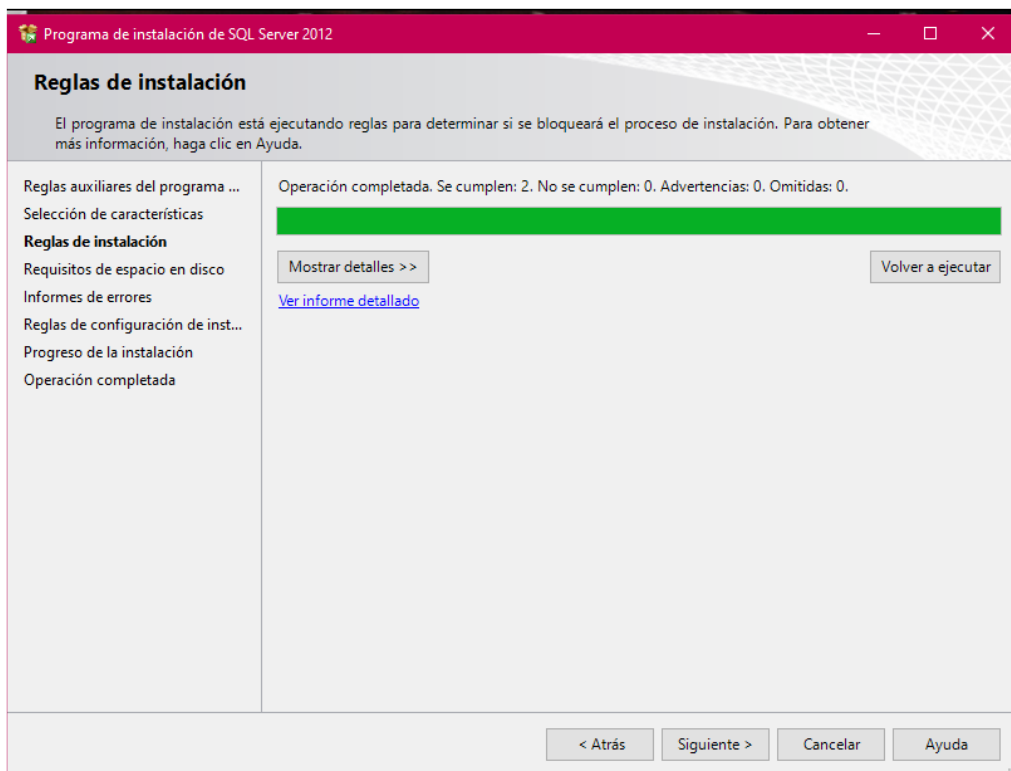


Figura 4 Validación de Reglas de Instalación

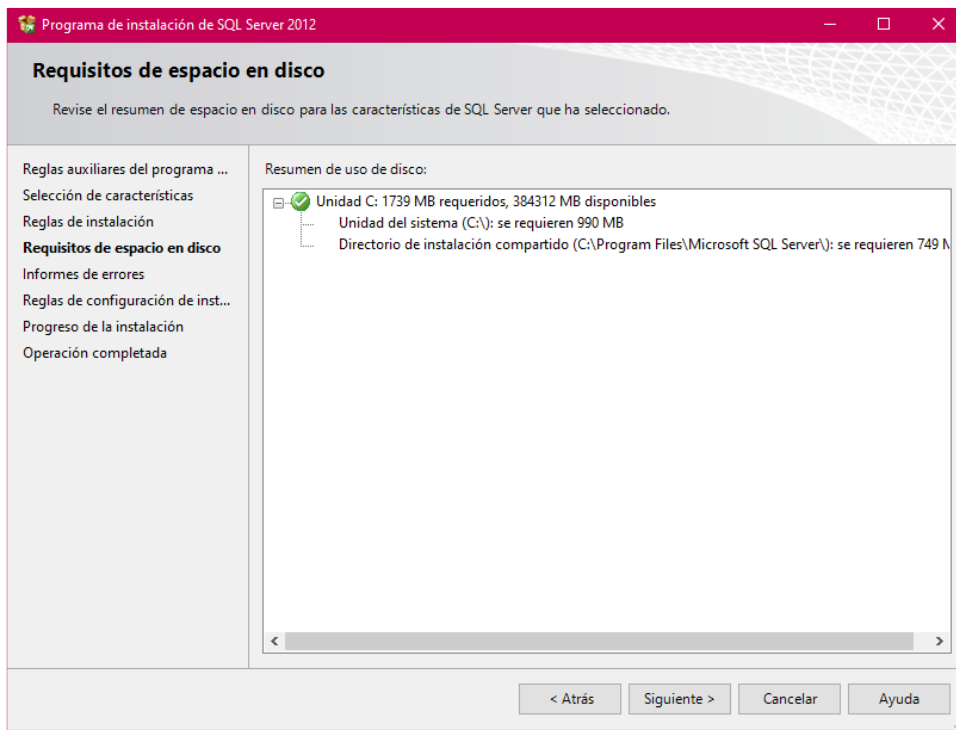


Figura 5 Requisitos de Espacio en Disco Duro

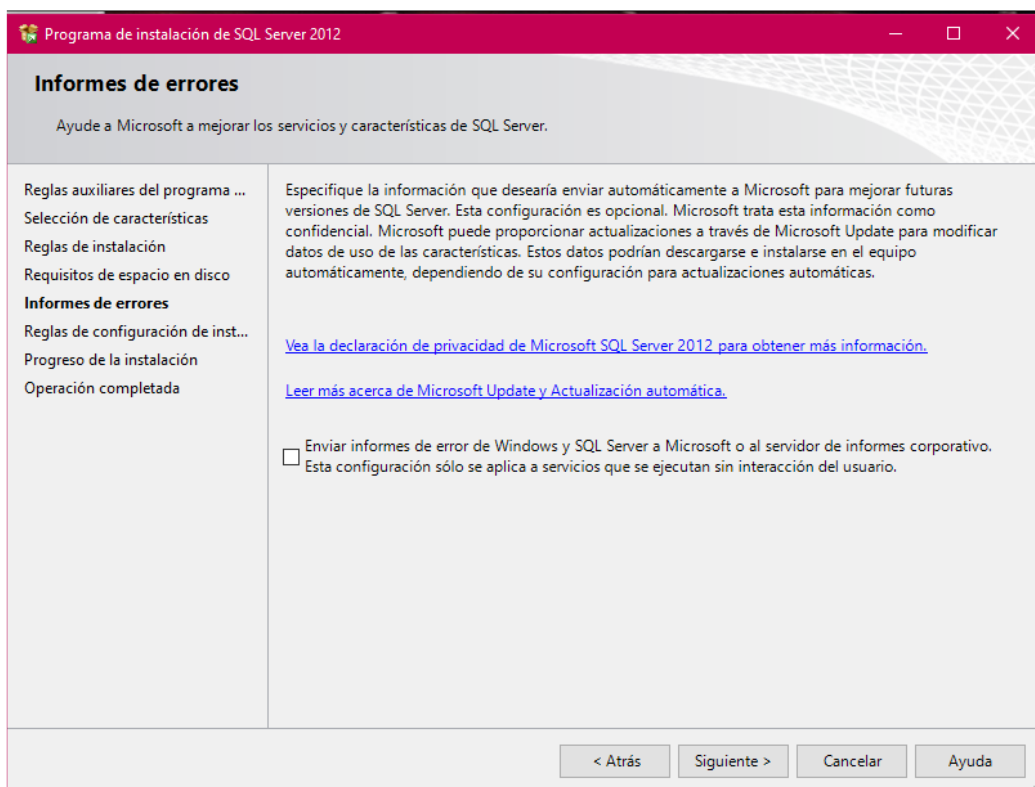


Figura 6 Informe de Errores

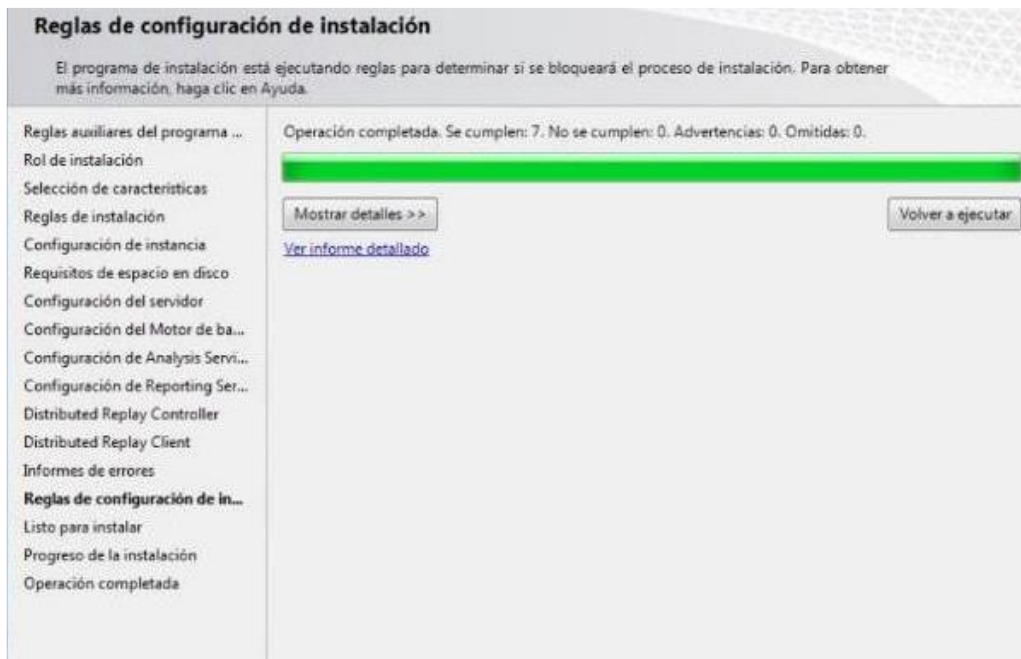


Figura 7 Reglas de Configuración

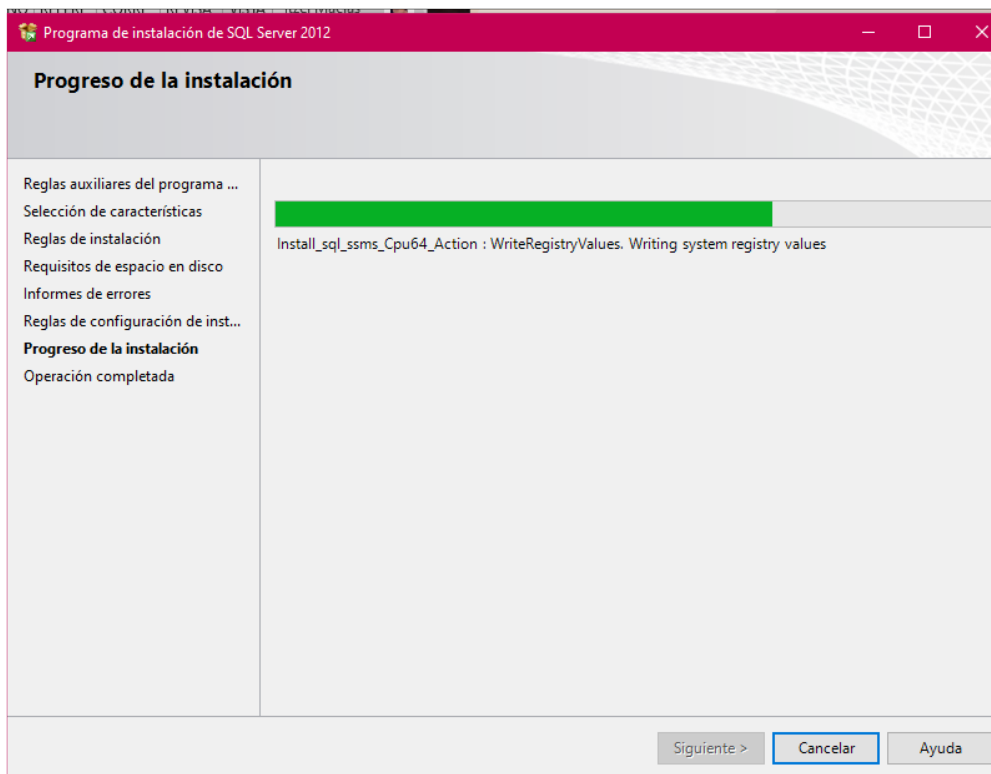


Figura 8 Progreso de Instalación

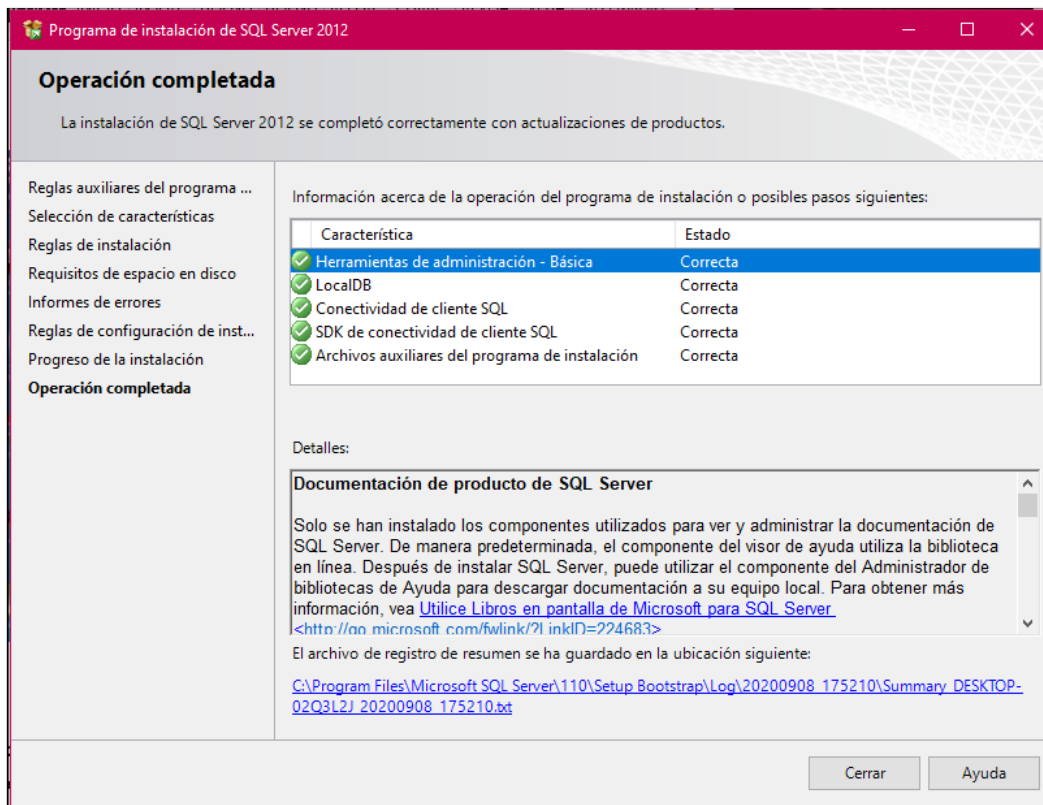


Figura 9 Operación Terminada

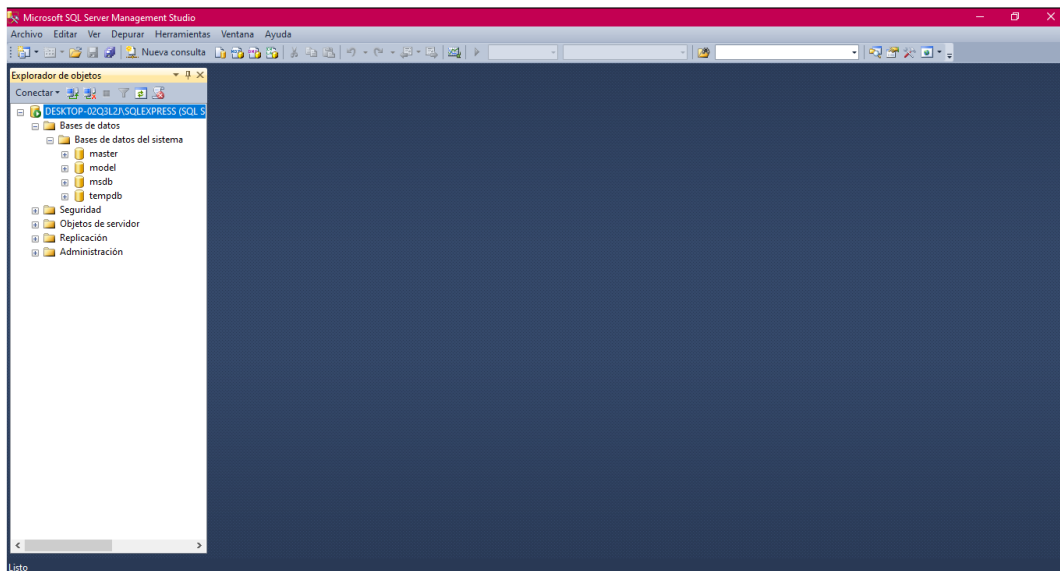


Figura 10 Microsoft Sql Server Management Studio

2.8 Comandos generales de alta y baja del SGBD

Comandos de Alta (Creación)

Los comandos de alta se utilizan para crear nuevas bases de datos, tablas, usuarios y otros objetos dentro de SQL Server.

Crear una Base de Datos:

SQL

```
CREATE DATABASE NombreDeLaBaseDeDatos;
```

Puedes especificar la ubicación de los archivos, el tamaño y otros parámetros. Por ejemplo:

SQL

```
CREATE DATABASE NombreDeLaBaseDeDatos ON
```

```
( NAME = NombreDeLaBaseDeDatos_dat,
```

```
FILENAME          =          'C:\Program          Files\Microsoft          SQL  
Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\NombreDeLaBaseDeDatos.mdf  
,
```

```
    SIZE = 10,
```

```
    MAXSIZE = 50,
```

```
    FILEGROWTH = 5 )
```

```
LOG ON
```

```
( NAME = NombreDeLaBaseDeDatos_log,
```

```
FILENAME='C:\ProgramFiles\MicrosoftSQL  
Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\NombreDeLaBaseDeDatos_log  
.ldf',
```

```
    SIZE = 5,
```

```
    MAXSIZE = 25,
```

```
    FILEGROWTH = 5 );
```

Crear una Tabla:

SQL

```
CREATE TABLE NombreDeLaTabla (
```

```
    Columna1 TipoDeDato,
```

```
    Columna2 TipoDeDato,
```

...

);

Ejemplo:

SQL

```
CREATE TABLE Empleados (  
    EmpleadoID INT PRIMARY KEY,  
    Nombre VARCHAR(50),  
    ApellidoPat VARCHAR(50),  
    ApellidoMat VARCHAR(50),  
    FechaContratacion DATE  
);
```

Crear un Login (para iniciar sesión en el servidor):

SQL

```
CREATE LOGIN NombreDeLogin WITH PASSWORD = 'ElPassword';
```

Crear un Usuario (para acceder a una base de datos específica):

SQL

```
CREATE USER NombreDeUsuario FOR LOGIN NombreDeLogin;
```

Comandos de Baja (Eliminación)

Los comandos de baja se utilizan para eliminar bases de datos, tablas, logins y otros objetos.

Eliminar una Base de Datos:

SQL

```
DROP DATABASE NombreDeLaBaseDeDatos;
```

Este comando elimina permanentemente la base de datos y todos sus archivos físicos.

Eliminar una Tabla:

SQL

```
DROP TABLE NombreDeLaTabla;
```

Este comando elimina la tabla y todos los datos que contiene.

Eliminar un Login:

SQL

DROP LOGIN NombreDeLogin;

Esto elimina la capacidad de un usuario para iniciar sesión en el servidor.

Eliminar un Usuario:

SQL

DROP USER NombreDeUsuario;

Esto elimina el acceso de un usuario a una base de datos específica.

Comandos de Modificación (No solicitados pero relevantes)

Modificar una Tabla:

SQL

ALTER TABLE NombreDeLaTabla

ADD Columna3 TipoDeDato;

Modificar una Base de Datos:

SQL

ALTER DATABASE NombreDeLaBaseDeDatos

SET OFFLINE;

Esto pone la base de datos fuera de línea.

CUESTIONARIO TEMA 2

1. ¿Cuál es la función principal del Buffer Cache en el Área de Datos Global (SGA) de SQL Server?

- a) Almacenar instrucciones SQL compiladas y planos de ejecución.
- b) Almacenar las páginas de datos más recientemente utilizadas para reducir lecturas físicas al disco.
- b) Gestionar la memoria dinámica del sistema operativo.

2. **¿Cuál de los siguientes procesos de SQL Server se encarga de ejecutar tareas programadas como copias de seguridad y mantenimiento?**
- a) Agente SQL Server
 - b) Coordinador de transacciones distribuidas
 - c) Explorador SQL Server
3. **¿Qué tipo de archivo en SQL Server contiene el registro de transacciones para recuperación y consistencia de datos?**
- a) Archivo de datos principal (.mdf)
 - b) Archivo de registro de transacciones (.ldf)
 - c) Archivo de datos secundario (.ndf)
4. **¿Cuál es la unidad fundamental de almacenamiento y Entrada/Salida (E/S) en SQL Server?**
- a) Extensión de 64 KB
 - b) Página de 8 KB
 - c) Bloque de 4 KB
5. **Para instalar SQL Server 2022, ¿cuál es el requisito mínimo recomendado de memoria RAM para un entorno de producción?**
- a) 1 GB
 - b) 2 GB
 - c) 4 GB
6. **¿Qué sistema operativo NO es compatible con SQL Server 2022 según el texto?**
- a) Windows 11
 - b) Windows Server 2012
 - c) Windows 10 versión 1607 o superior
7. **En la instalación de SQL Server para replicación transaccional, ¿cuál es el primer paso para configurar la replicación?**
- a) Crear una publicación
 - b) Configurar el distribuidor
 - c) Crear una suscripción
8. **¿Cuál de las siguientes variables de entorno es importante para definir la ubicación de archivos temporales durante la instalación?**
- a) RUTA
 - b) TEMP
 - c) RUTA SQL

9. ¿Dónde se recomienda ubicar los archivos de datos (mdf, .ndf) para mejorar el rendimiento y la recuperación?

- a) En la unidad C:\ junto con el sistema operativo
- b) En una unidad de disco separada diferente a la unidad C:\
- c) En el mismo directorio que los archivos de registro de transacciones

10. ¿Cuál es el comando SQL correcto para eliminar una tabla llamada "Empleados"?

- a) BAJA DE BASE DE DATOS Empleados;
- b) BORRAR TABLA Empleados;
- c) DROP TABLE Empleados;

Respuestas correctas:

- 1. b) Almacenar las páginas de datos más recientemente utilizadas para reducir lecturas físicas al disco.
- 2. a) Agente SQL Server
- 3. b) Archivo de registro de transacciones (.ldf)
- 4. b) Página de 8 KB
- 5. c) 4 GB
- 6. b) Windows Server 2012
- 7. b) Configurar el distribuidor
- 8. b) TEMP
- 9. b) En una unidad de disco separada diferente a la unidad C:\
- 10. c) TABLA DE Empleados DROP;

Tema 3: Configuración y Administración del Espacio en Disco

Configura y administra el espacio en disco y memoria del servidor para que el funcionamiento del SGBD sea congruente con la infraestructura existente.

3.1 Definición de espacio de almacenamiento

- **Páginas y extensiones:**

En SQL Server, la página es como el ladrillo básico donde se guarda la información. Cada vez que el sistema necesita leer o escribir datos, lo hace trabajando con estas páginas, que siempre tienen el mismo tamaño: 8KB.

Por otro lado, la extensión es como un bloque más grande que se compone de ocho páginas, lo que suma en total 64KB. SQL Server utiliza estas extensiones para asignar espacio a las tablas e índices, facilitando así el manejo y organización de la información dentro de la base de datos.

- **Archivos de base de datos:**

- **Archivos de datos (.mdf, .ndf):**

- .mdf (Master Data File): Archivo de datos principal que contiene la información de inicio de la base de datos y los punteros a los demás archivos.
 - .ndf (Secondary Data File): Archivos de datos adicionales que permiten distribuir la base de datos en múltiples ubicaciones físicas (discos), lo que puede mejorar el rendimiento al permitir operaciones de I/O concurrentes.

- **Archivos de registro de transacciones (.ldf):**

- Contienen el registro de todas las transacciones realizadas en la base de datos.
 - Es fundamental para la integridad de los datos, la recuperación ante fallos y las operaciones de respaldo.
 - Las escrituras en el registro son secuenciales, lo que lo hace sensible a la latencia del disco.

- **Grupos de archivos (Filegroups):**

Un grupo de archivos es como una especie de "carpeta lógica" dentro de la base de datos que te ayuda a organizar uno o más archivos de datos. Gracias a esto, puedes administrar el espacio de almacenamiento de manera más

sencilla y detallada. Por ejemplo, si tienes tablas o índices que se consultan mucho (los famosos datos calientes), puedes colocarlos en un grupo de archivos específico que esté en un disco rápido. En cambio, si tienes información que casi no se usa (datos fríos), puedes guardarla en otro grupo de archivos situado en un disco más lento o económico. Así aprovechas mejor los recursos y el rendimiento de tus discos.

3.2. Definición y creación del espacio asignado para cada base de datos

- Cálculo del tamaño inicial de las tablas y los índices.
- Estimación del crecimiento futuro de los datos y las transacciones.
- **Diseño de la arquitectura de almacenamiento:**
 - Separación física de los archivos de datos, registro de transacciones y TempDB. Esta es una de las mejores prácticas más importantes.
 - **Datos:** Unidad de alta velocidad (SSD, NVMe) con alta capacidad de I/O aleatorio.
 - **Registro de transacciones:** Unidad de alta velocidad (SSD, NVMe) con baja latencia y alta capacidad de escritura secuencial.
 - **TempDB:** Unidad de la mayor velocidad posible, idealmente una SSD local o incluso RAMDisk para cargas de trabajo extremas.
 - Considerar el uso de RAID (Redundant Array of Independent Disks) para mejorar la redundancia y el rendimiento (ej. RAID 10 para balancear ambas).

Crecimiento automático (Autogrowth):

- **Recomendaciones:**

En vez de dejar que los archivos de la base de datos crezcan de manera aleatoria, es mejor definir un aumento fijo en megabytes o gigabytes. Así evitas sorpresas y sabes exactamente cuánto espacio se va a usar cada vez que crezca. Además, procura que este incremento sea lo suficientemente grande para que no se esté ejecutando el crecimiento muy seguido, ya que eso puede fragmentar el archivo y afectar el rendimiento.

Otra recomendación útil: ponle un límite máximo al tamaño de los archivos. Esto te ayuda a evitar que una base de datos mal configurada termine llenando todo el disco duro y cause problemas en el servidor.

- **Riesgos:**

Si la configuración de crecimiento automático (autogrowth) no está bien hecha, los archivos de la base de datos pueden fragmentarse y esto puede afectar el desempeño. Además, si el registro de transacciones crece sin control, se pueden generar muchos archivos de registro virtuales (VLF) pequeños, lo que también puede hacer que la base de datos funcione más lenta.

- **Inicialización instantánea de archivos (Instant File Initialization - IFI):**

Descripción: Es un permiso de seguridad que le permite a SQL Server crear o aumentar archivos de datos casi al instante, sin tener que pasar por el proceso lento de llenar el espacio con ceros. Esto hace que todo sea mucho más rápido.

Importancia: Gracias a este permiso, las operaciones como el crecimiento automático de la base de datos, restauraciones y creaciones de bases de datos se hacen en mucho menos tiempo. Es decir, el servidor puede trabajar más ágilmente y no te hace esperar tanto.

Configuración: Para activar este beneficio, el usuario que ejecuta el servicio de SQL Server necesita contar con el permiso de “Realizar tareas de mantenimiento de volumen” en Windows. Es un ajuste sencillo que puede marcar la diferencia en el rendimiento.

- **Reducción de archivos (Shrinking):**

- **Advertencia:** Es una de las operaciones más malinterpretadas y peligrosas para el rendimiento.
- **Problemas:** El shrinking reorganiza las páginas de datos y puede causar una **fragmentación severa de los índices y los datos**, lo que degrada el rendimiento de las consultas.
- **Mejor práctica:** Evitar el shrinking siempre que sea posible. Si es absolutamente necesario, se debe programar la reconstrucción de los índices inmediatamente después de la operación.

¡Aguas con la reducción de archivos! Muchos piensan que es una solución fácil, pero si no la usas con precaución, puede meterte en serios problemas de rendimiento en tu base de datos. ¿Por qué? Porque al hacer el shrinking, el sistema se pone a mover y reacomodar las páginas de datos, y esto puede dejar los índices y los datos todos fragmentados. Como resultado, tus consultas se vuelven más lentas y el servidor batalla para responder rápido.

La recomendación: no uses el shrinking a menos que de verdad no tengas otra opción. Y si tienes que hacerlo sí o sí, es súper importante que reconstruyas los índices en cuanto termines, así evitas que el desempeño se vea afectado.

- **Administración del registro de transacciones:**

- El archivo de registro crece continuamente. Para liberar espacio, se debe realizar una copia de seguridad del registro de transacciones.
- El modelo de recuperación de la base de datos determina cómo se gestiona el registro de transacciones (Simple vs. Full).

Ejemplo :

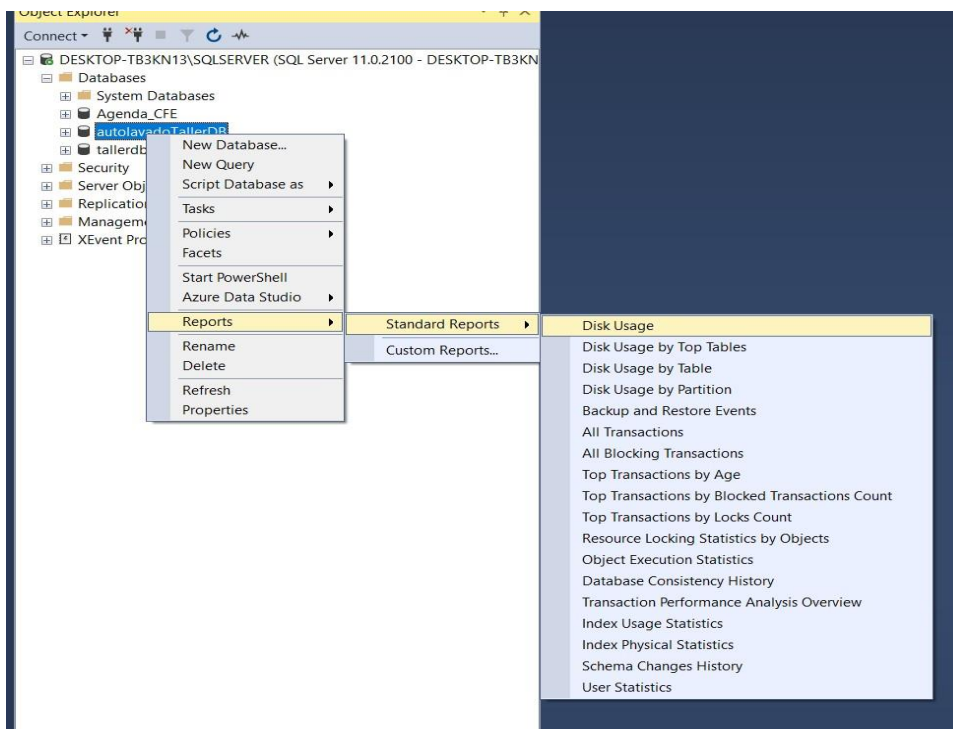


Figura 11 Disk Usage

El espacio en memoria de la base de datos podemos consultarla haciendo un reporte en la base de datos de tipo 'Uso de disco' o 'Disk Usage'.

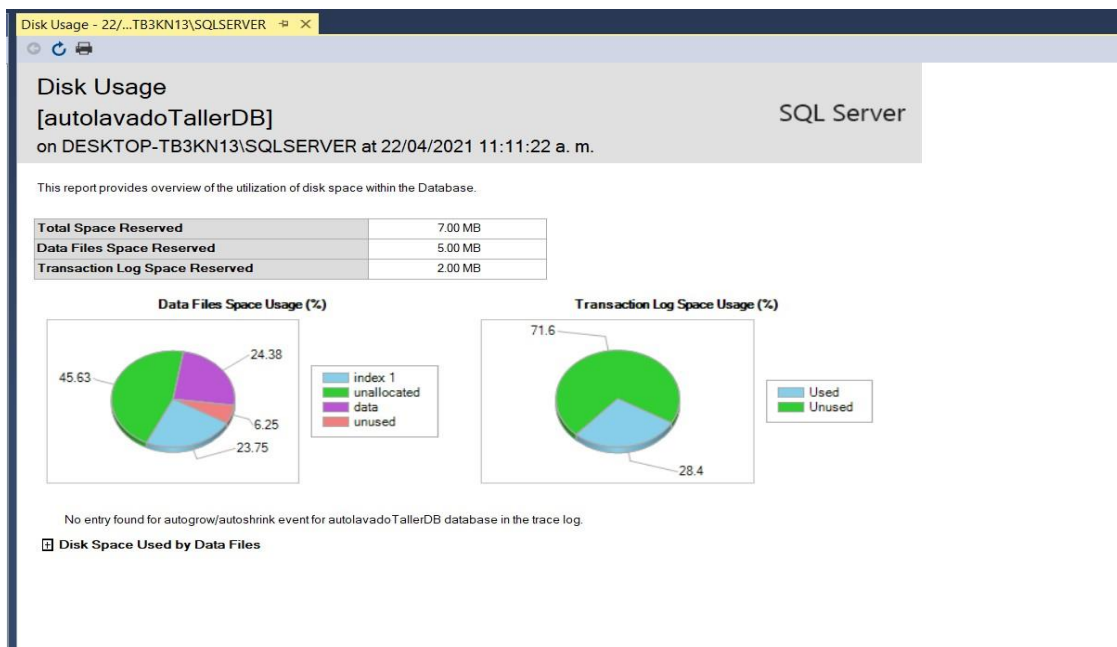


Figura 12 Reporte

El reporte nos muestra con gráficas y tablas lo referente al uso de disco que la base de datos tiene en la máquina.

```
SQLQuery1.sql - D...B3KN13\Oscar (53))* X Disk Usage - 22/...TB3KN13\SQLSERVER
USE master;
GO
ALTER DATABASE autolavadoTallerDB
MODIFY FILE
(NAME = autolavadoTallerDB,
SIZE = 20MB);
GO
```

Figura 13 Script

Para modificar el tamaño de disco que utiliza la base de datos puede ser con este script SQL utilizando el master para modificar directamente este atributo.

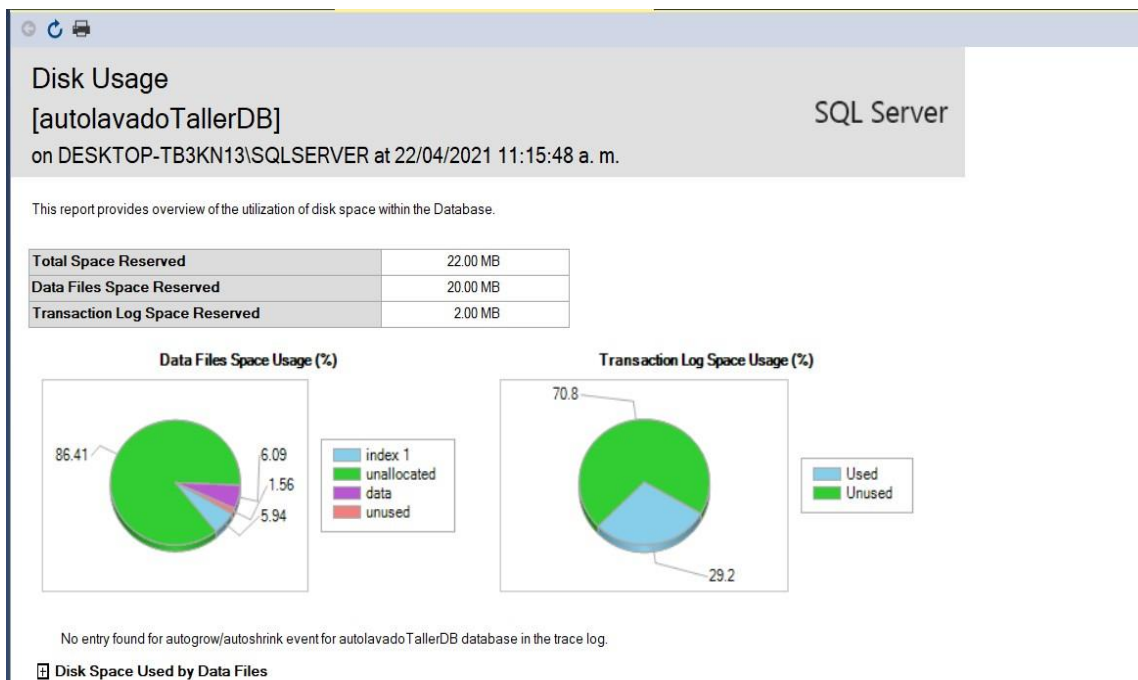


Figura 14 Reporte de uso de disco

Con F5 refrescamos el reporte de uso de disco y podemos ver que el uso reservado de la base de datos se ha modificado.

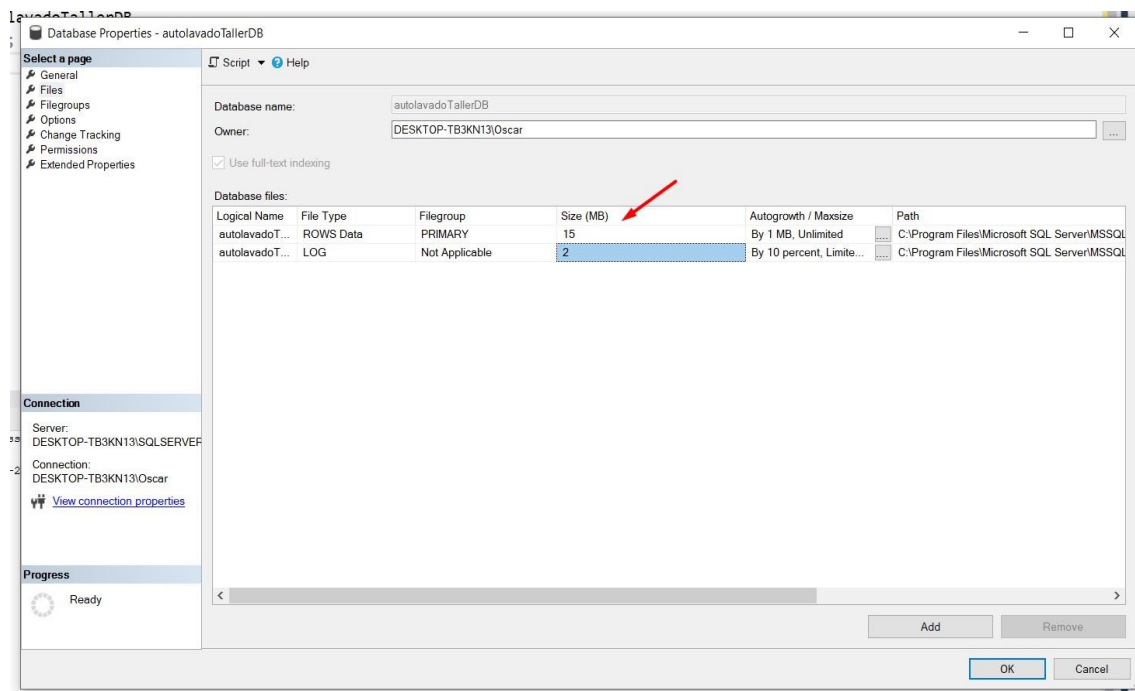


Figura 15 Propiedades de la base de datos

También otra forma de cambiar el tamaño de la base de datos es directamente en las propiedades de la base de datos, en el apartado de archivos, cambiar el tamaño del espacio reservado.

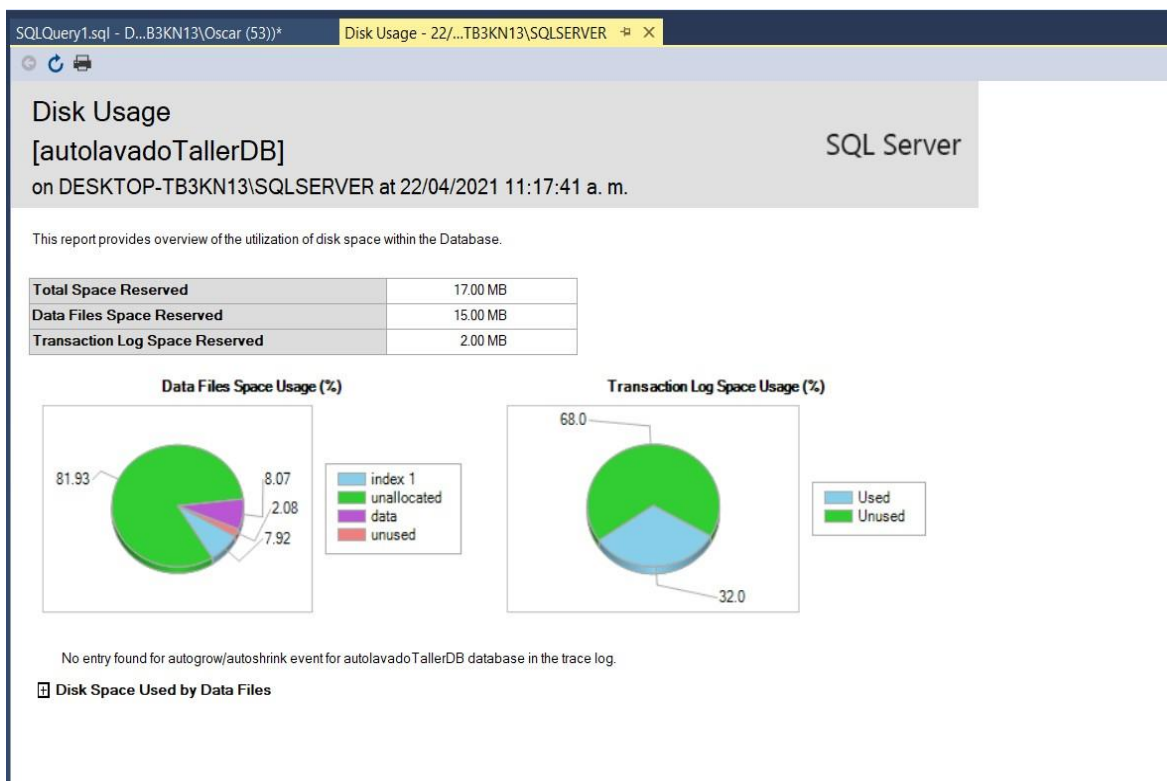


Figura 16 Disk Usage modificado

Volvemos a comprobar que el tamaño se ha modificado.

3.3 Asignación de cuotas de espacio para usuarios

La asignación de cuotas de espacio por usuario no es una funcionalidad integrada de forma nativa en SQL Server 2012 ni en versiones más recientes. A diferencia de otros sistemas operativos o gestores de bases de datos, SQL Server no tiene un comando o una característica directa para limitar el espacio de disco que puede consumir un usuario o un login específico.

Sin embargo, hay estrategias y métodos alternativos para gestionar el consumo de espacio a nivel de usuario, que implican una combinación de monitoreo y programación.

Métodos para gestionar el espacio de usuarios

1. Restricciones a nivel de base de datos o esquema:

- Puedes limitar el crecimiento de una base de datos estableciendo un tamaño máximo (MAXSIZE). Aunque esto no es por usuario, puede evitar que una sola base de datos consuma todo el espacio del disco.
- A través de esquemas, se puede segregar los objetos de diferentes usuarios. Esto no limita el espacio, pero sí organiza los datos para un mejor monitoreo.

2. Monitoreo del espacio ocupado:

- Puedes monitorear el espacio que consumen los usuarios o los objetos que les pertenecen. Esto se hace consultando las vistas del sistema.
- Se puede usar la vista `sys.dm_db_partition_stats` para ver el número de filas y el espacio ocupado por cada tabla. Al combinar esta información con los permisos de los usuarios, puedes identificar quién está consumiendo más espacio.

3. Implementación de políticas y programación:

- Se pueden crear triggers (disparadores) para que se activen cuando se insertan datos en una tabla. El trigger puede verificar el tamaño de la tabla o el espacio disponible en el disco y, si se excede un límite, denegar la inserción.
- Se pueden usar trabajos del Agente SQL Server para ejecutar scripts que monitoreen periódicamente el uso del espacio de disco. Estos scripts pueden enviar notificaciones o incluso desactivar temporalmente el acceso de un usuario si ha superado un umbral de uso.

En SQL Server, la seguridad funciona principalmente a través de permisos, como GRANT, DENY y REVOKE, que se aplican sobre objetos y el acceso a la base de datos. Sin embargo, no existe una forma directa de limitar el espacio en disco que puede usar cada usuario. Esto se debe a que SQL Server está pensado para administrar bases de datos completas de manera eficiente, dejando en manos del administrador la responsabilidad de vigilar el espacio disponible y el crecimiento de los archivos, en vez de controlar detalladamente cuánto espacio usa cada usuario por separado.

3.4 Espacios para objetos de la base de datos

Para asignar una tabla, un índice o cualquier otro objeto de la base de datos a un grupo de archivos específico, se utiliza la cláusula **ON** en los comandos **CREATE TABLE** o **CREATE INDEX**. Por ejemplo al crear una tabla se especifica :

```
CREATE TABLE Ventas (
```

```
    VentaID INT PRIMARY KEY,
```

```
    FechaVenta DATE,
```

```
    Total DECIMAL(10, 2)
```

```
ON Ventas_2024_FG; -- Asigna la tabla al grupo de archivos 'Ventas_2024_FG'
```

De esta manera, los datos de la tabla Ventas se almacenarán físicamente en los archivos de datos asociados a ese grupo de archivos. Esto permite una granularidad precisa en la gestión del almacenamiento.

3.5 Roles

SQL Server tiene una forma de agrupar permisos y asignarlos a usuarios o logins, simplificando la administración de la seguridad. En lugar de otorgar permisos a cada usuario individualmente, se le otorga acceso a un rol y este hereda los permisos del rol.

Tipos de Roles

Existen dos tipos principales de roles:

1. Roles Fijos del Servidor

Son roles predefinidos por SQL Server que tienen permisos de nivel de servidor. No se pueden modificar ni eliminar. Se usan para otorgar permisos a nivel de la instancia de SQL Server.

Nombre del Rol	Descripción y Uso
----------------	-------------------

sysadmin	El rol de más alto nivel. Un miembro de este rol tiene acceso completo a todas las bases de datos y configuraciones del servidor. Es el equivalente a ser el "administrador del sistema" del SQL Server.
-----------------	---

serveradmin	Puede realizar tareas de configuración a nivel de servidor, como apagarlo o reiniciar. Es un rol de alto nivel, pero con menos privilegios que sysadmin.
--------------------	--

Nombre del Rol Descripción y Uso

setupadmin	Puede agregar y eliminar servidores vinculados (linked servers) y realizar otras tareas de configuración de alto nivel.
securityadmin	Puede gestionar logins y sus propiedades, crear, eliminar y cambiar contraseñas.
processadmin	Puede terminar procesos que se ejecutan en el servidor, como consultas que se han quedado atascadas.
dbcreator	Puede crear, modificar y eliminar bases de datos.
diskadmin	Puede gestionar los archivos de la base de datos, como los archivos de datos y de registro.
bulkadmin	Permite a los usuarios ejecutar la instrucción BULK INSERT para cargar datos de archivos en tablas de la base de datos.

2. Roles Fijos de Base de Datos

Estos roles tienen permisos predefinidos a nivel de una base de datos específica.

Nombre del Rol	Descripción y Uso
db_owner	El rol más poderoso a nivel de base de datos. Un miembro tiene control total sobre todos los objetos y permisos de la base de datos, incluyendo la capacidad de eliminarla.
db_securityadmin	Puede gestionar permisos, roles y miembros de roles dentro de la base de datos.
db_accessadmin	Puede agregar o eliminar logins de Windows y usuarios de SQL Server dentro de la base de datos.
db_datareader	Puede leer todos los datos de todas las tablas y vistas en la base de datos. Es ideal para usuarios que solo necesitan ver la información.
db_datawriter	Puede agregar, eliminar y modificar datos en todas las tablas de la base de datos.
db_ddladmin	Puede ejecutar comandos DDL (Data Definition Language) para crear, modificar o eliminar tablas, vistas e índices.

Para asignar un rol a un usuario, se utiliza la siguiente sintaxis:

En SQL

-- Agregar un login (Usuario) al rol fijo de servidor 'sysadmin'

```
ALTER SERVER ROLE sysadmin ADD MEMBER [NombreDeLogin];
```

-- Agregar un usuario al rol fijo de base de datos 'db_datareader'

```
USE NombreDeLaBaseDeDatos;
```

```
ALTER ROLE db_datareader ADD MEMBER [NombreDeUsuario];
```

El uso de roles simplifica enormemente la gestión de la seguridad, ya que permite que los administradores asignen permisos de forma centralizada y granular, sin tener que gestionar cada permiso para cada usuario de forma individual.

Ejemplo

La creación de roles es una actividad muy relevante al momento de administrar base de datos, es recomendable que el usuario sa o super administrador no sea utilizado para manejar la base de datos de manera constante, el sa tendría el rol de administrar los demás roles que se creen en la base de datos.

En SQL Server la administración de roles de una base de datos se encuentra en la misma en una carpeta llamada seguridad.

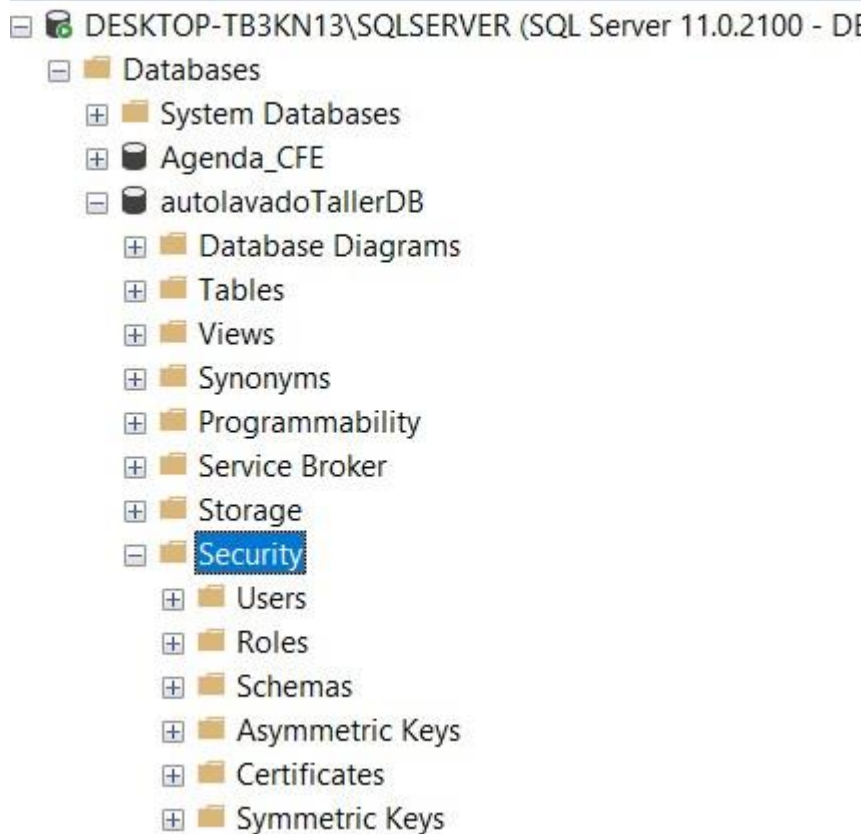


Figura 17 Roles

Para crear un nuevo rol, con click derecho en Roles, nuevo, nuevo rol de base de datos.

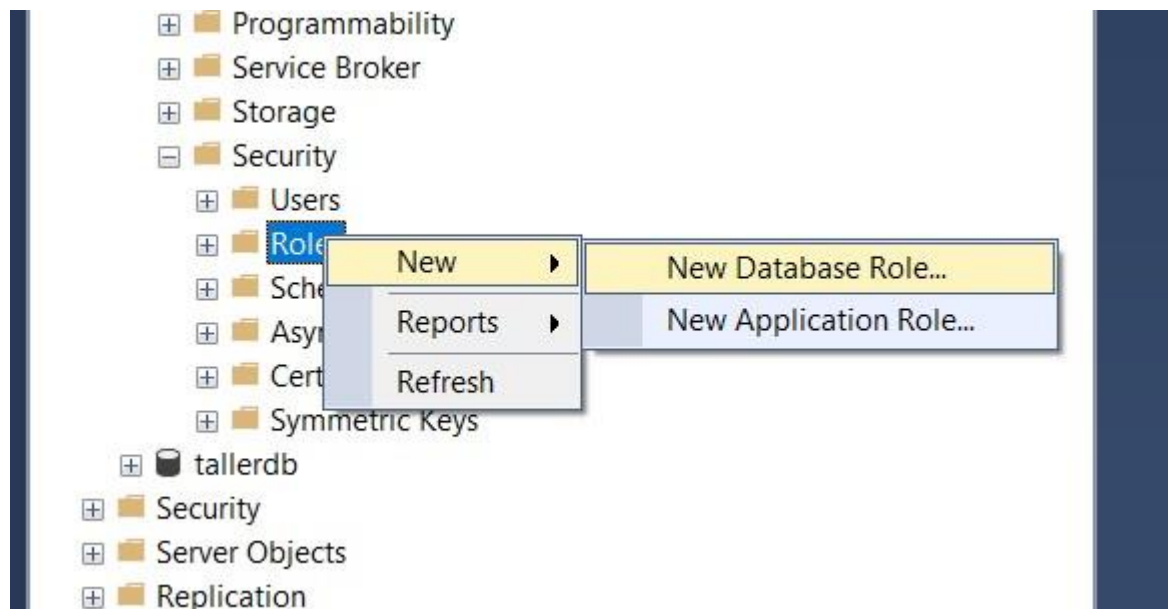


Figura 18 Nuevo rol de Base de Datos

El rol se llamará Servicio Técnico, en el solo tendrá acceso a ciertas tablas para leerlas y en la tabla reportes podrá insertar nuevos datos.

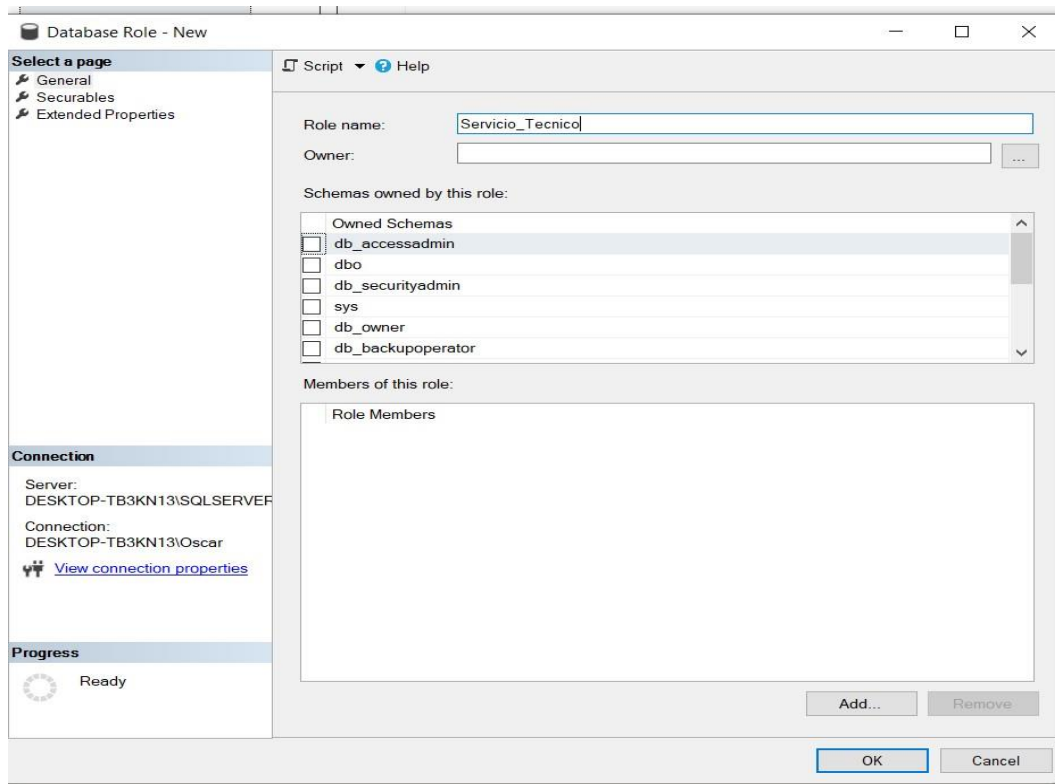


Figura 19 Definición de rol

En la sección Securables es donde definiremos a qué secciones de la base de datos este rol tendrá acceso.

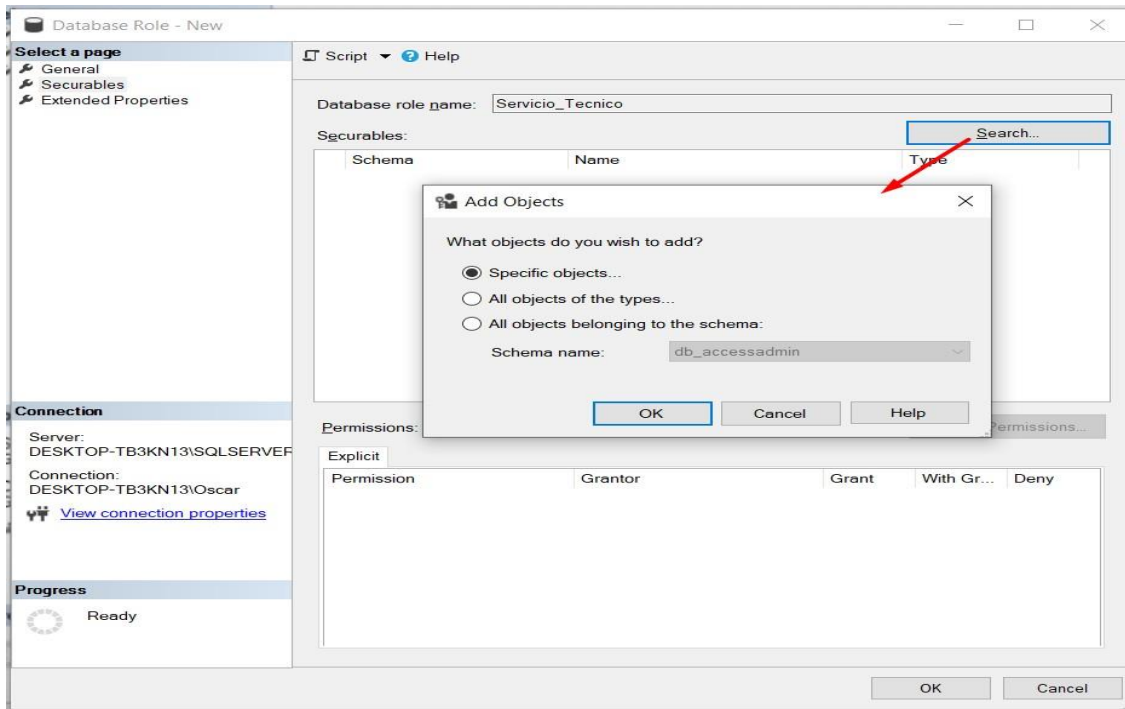


Figura 20 tipo de objetos

Buscamos el tipo de objetos, en este caso serán las Tablas, solamente. En esta ventana encontramos muchísimos objetos más que nos ayudarán a tener bastantes roles en el futuro.

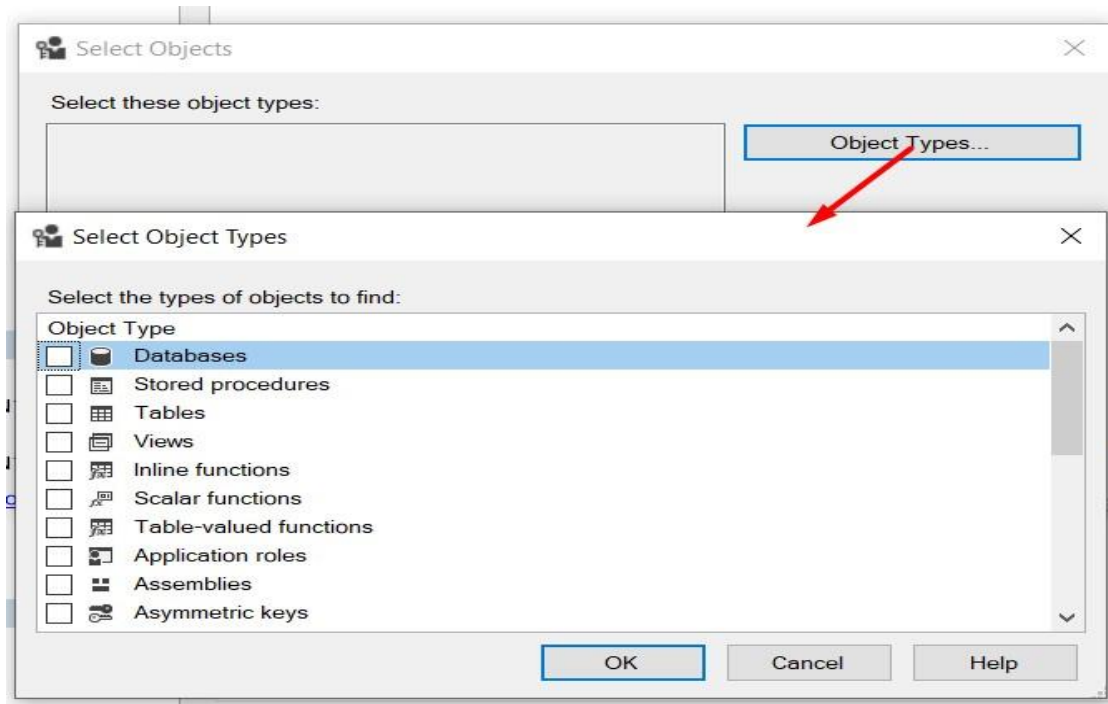


Figura 21 Selección de BD

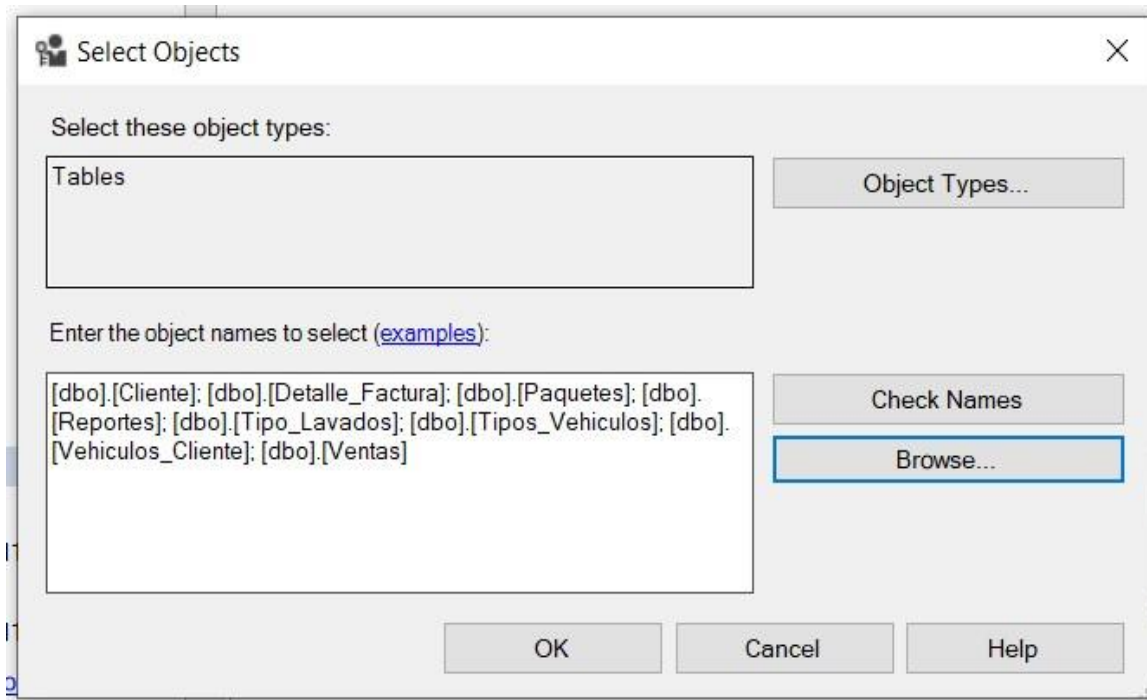


Figura 22 Rol de acceso

El objeto que queremos para este rol es el de tablas, y en la segunda caja agregaremos las tablas de la base de datos que queremos que este rol tenga acceso.

CUESTIONARIO TEMA 3

1. **¿Cuál es el tamaño fijo de una página en SQL Server?**
 - a) 4 KB
 - b) 8 KB
 - c) 16 KB
2. **¿Qué es una extensión en SQL Server?**
 - a) Un archivo de datos adicional
 - b) La unidad de asignación de espacio que consta de ocho páginas
 - c) Un permiso de seguridad para la creación de archivos
3. **¿Cuál es la función principal del archivo ldf en una base de datos SQL Server?**
 - a) Contener los datos principales de la base de datos

- b) Almacenar el registro de todas las transacciones realizadas
 - c) Guardar los objetos temporales de la base de datos
4. **¿Qué permite hacer un grupo de archivos (filegroup) en SQL Server?**
- a) Agrupar los archivos de registro para mejorar la seguridad
 - b) Administrar el almacenamiento agrupando uno o más archivos de datos
 - c) Limitar el tamaño máximo de los archivos de la base de datos
5. **¿Cuál es una buena práctica recomendada para el crecimiento automático de archivos en SQL Server?**
- a) Configurar incrementos porcentuales para un crecimiento flexible
 - b) Configurar un incremento fijo en MB o GB para evitar la fragmentación
 - c) No establecer tamaño máximo para permitir crecimiento ilimitado
6. **¿Qué requisito debe cumplir el usuario del servicio de SQL Server para que funcione la Inicialización Instantánea de Archivos (Instant File Initialization)?**
- a) Tener permisos de administrador local
 - b) Tener el permiso “Realizar tareas de mantenimiento de volumen” en Windows
 - c) Estar asignado al rol sysadmin en SQL Server
7. **¿Cuál es uno de los principales riesgos al realizar la operación de encogimiento en archivos de base de datos?**
- a) Incremento del tamaño del archivo de registro
 - b) Fragmentación severa de índices y datos
 - c) Pérdida de datos irreversibles
8. **¿Cómo se puede liberar espacio en el archivo de registro de transacciones?**
- a) Eliminando el archivo .ldf manualmente
 - b) Realizando una copia de seguridad del registro de transacciones
 - c) Ejecutando una reducción del archivo de datos
9. **¿SQL Server permite asignar cuotas de espacio directamente a los usuarios o inicios de sesión?**
- a) Sí, mediante comandos nativos específicos
 - b) No, pero se pueden usar métodos alternativos de monitoreo y control
 - c) Sí, pero sólo en versiones posteriores a SQL Server 2012

10. **¿Cuál es la forma correcta de asignar una tabla a un grupo de archivos específicos?**

- a) Usando la cláusula ON en el comando CREATE TABLE
- b) Mediante la configuración del archivo .mdf
- c) A través de la creación de un rol fijo de base de datos

Respuestas Correctas:

- 1: b) 8 KB
- 2: b) La unidad de asignación de espacio que consta de ocho páginas
- 3: b) Almacenar el registro de todas las transacciones realizadas
- 4: b) Administrar el almacenamiento agrupando uno o más archivos de datos
- 5: b) Configurar un incremento fijo en MB o GB para evitar la fragmentación
- 6: b) Tener el permiso “Realizar tareas de mantenimiento de volumen” en Windows
- 7: b) Fragmentación severa de índices y datos
- 8: b) Realizando una copia de seguridad del registro de transacciones
- 9: b) No, pero se pueden usar métodos alternativos de monitoreo y control
- 10: a) Usando la cláusula ON en el comando CREATE TABLE

Tema 4 Operación y Mantenimiento

Identifica en los archivos log el funcionamiento de SGBD para prevenir cualquier problema del mismo.

Manipula los modos de operación del SGBD con el propósito de optimizar el funcionamiento de la base de datos.

Organiza los índices y planea su reorganización periódica para el acceso eficiente de datos.

4.1 Archivos log del SGBD

Los archivos de bitácora, o *log files*, son registros detallados que el Sistema de Gestión de Bases de Datos (SGBD) mantiene sobre todas las operaciones y transacciones que ocurren en el sistema. Piensa en ellos como la "caja negra" de la base de datos.

- **Funciones principales:**
 - **Recuperación:** Son esenciales para la recuperación de la base de datos en caso de fallos (p. ej., un fallo del servidor). Permiten deshacer transacciones incompletas (*rollback*) o rehacer transacciones completadas (*commit*) para restaurar la base de datos a un estado consistente.
 - **Auditoría y control:** Proporcionan un historial de quién hizo qué y cuándo. Esto es vital para la seguridad y la auditoría.
 - **Monitoreo del rendimiento:** Ayudan a identificar problemas de rendimiento o cuellos de botella en el sistema.
- **Modos de operación:** Los SGBD tienen diferentes modos de operación que controlan cómo se accede y se modifica la base de datos. Esto puede incluir modos en línea (disponibles para los usuarios) y modos fuera de línea (utilizados para el mantenimiento).
- **Concepto de transacción:** Una transacción es una secuencia de operaciones de base de datos que se trata como una única unidad lógica de trabajo. Una transacción debe cumplir con las propiedades **ACID**:
 - **Atomicidad (Atomicity):** Una transacción se completa en su totalidad o no se completa en absoluto.
 - **Consistencia (Consistency):** Una transacción debe llevar la base de datos de un estado válido a otro.

- **Aislamiento (Isolation):** Las transacciones concurrentes no deben interferir entre sí.
- **Durabilidad (Durability):** Una vez que una transacción ha sido confirmada (*committed*), sus cambios son permanentes.
- **Comandos básicos:** Se estudian los comandos esenciales para el manejo de transacciones, como COMMIT (para guardar los cambios de forma permanente) y ROLLBACK (para deshacer los cambios).

Respaldo y Recuperación de una Base de Datos

- **Importancia:** La pérdida de datos es uno de los mayores riesgos para cualquier organización. Un plan de respaldo y recuperación sólido es fundamental para garantizar la continuidad del negocio.
- **Planeación del respaldo:** Esto implica definir una estrategia que responda a preguntas como: ¿Qué datos respaldar? ¿Con qué frecuencia? ¿Dónde almacenar los respaldos? ¿Cómo probar los respaldos?
- **Métodos de respaldo:**
 - **Respaldo completo (*full backup*):** Se copia toda la base de datos.
 - **Respaldo diferencial (*differential backup*):** Se copian solo los datos que han cambiado desde el último respaldo completo.
 - **Respaldo incremental (*incremental backup*):** Se copian solo los datos que han cambiado desde el último respaldo de cualquier tipo.
- **Métodos de recuperación:** Se revisan los procedimientos para restaurar la base de datos a partir de los respaldos, utilizando los archivos de bitácora para aplicar las transacciones necesarias y llevar la base de datos al estado deseado antes del fallo.

4.3 Índices, reorganización y reconstrucción

- **Optimización de índices:** Los índices son estructuras que mejoran la velocidad de las consultas. El mantenimiento incluye:
 - **Desfragmentación de índices:** Reorganizar los índices para mejorar la eficiencia del almacenamiento.
 - **Reconstrucción de índices:** Volver a crear los índices para que estén optimizados.

- Un índice es una estructura de disco asociada a una tabla o vista que acelera la recuperación de filas. Almacena una copia de una o varias columnas de una tabla en una estructura de datos organizada, como un árbol B-Tree, que permite a SQL Server encontrar datos de manera mucho más rápida que si tuviera que recorrer toda la tabla fila por fila.

Existen dos tipos principales de índices:

- **Índices Clúster (Clustered Index):** Son la forma más rápida de acceder a los datos. Organizan físicamente el almacenamiento de las filas de la tabla en el orden de las claves del índice. Cada tabla solo puede tener un índice clúster. Si una tabla tiene una clave primaria, SQL Server la crea como un índice clúster por defecto.
- **Índices No Clúster (Non-Clustered Index):** No ordenan físicamente las filas de la tabla. En su lugar, crean una estructura de datos separada que contiene los valores de las columnas del índice y un puntero a la ubicación física de la fila en la tabla. Una tabla puede tener muchos índices no clúster.

Reorganización (Reorganize)

La reorganización de índices es como darle mantenimiento a una carretera: acomoda y ordena las hojas del índice para quitar la fragmentación, pero sin hacer cambios drásticos. Esta técnica es ideal cuando la fragmentación es baja, es decir, menor al 30%.

¿Cómo lo hace? Reubica las páginas de datos dentro del mismo archivo, eliminando los huecos y mejorando el orden, todo sin ocupar espacio extra y sin interrumpir a los usuarios. Es decir, el índice sigue funcionando mientras se reorganiza, así que nadie se queda sin acceso a la información.

¿Cuándo conviene usarla? Cuando quieres realizar una mejora rápida y sencilla, sin bloquear el acceso a la base de datos. Es perfecto como mantenimiento preventivo para mantener todo en orden y funcionando de manera eficiente.

Reconstrucción (Rebuild)

La reconstrucción de índices es como hacerle una renovación profunda a una carretera muy dañada: básicamente, borra el índice actual y lo crea desde cero. Esta tarea es mucho más pesada que solo darle mantenimiento y se recomienda cuando la fragmentación es alta, es decir, cuando supera el 30%.

¿Cómo funciona? Imagina que construyes un índice completamente nuevo y, una vez listo, sustituyes el viejo por el nuevo. Así, desaparecen los problemas de fragmentación, se actualizan las estadísticas y el espacio en disco se aprovecha de mejor manera.

¿Cuándo conviene aplicarla? Cuando la fragmentación ya es considerable y el mantenimiento ligero ya no es suficiente. Toma en cuenta que esta operación puede dejar la tabla fuera de servicio por un rato (offline), aunque si tienes la versión Enterprise de SQL Server, puedes hacer la reconstrucción sin dejar a los usuarios sin acceso.

Característica	Reorganización (ALTER INDEX ... REORGANIZE)	Reconstrucción (ALTER INDEX ... REBUILD)
Nivel de fragmentación	Baja (< 30%)	Alta (> 30%)
Disponibilidad	Siempre en línea.	Puede ser en línea (Enterprise Edition) o sin conexión.
Uso de recursos	Bajo (optimiza en su lugar).	Alto (requiere espacio temporal).
Actualiza estadísticas	No por defecto.	Sí.
Objetivo	Eliminar fragmentación lógica.	Eliminar fragmentación lógica y física.

Actualizar la base de datos es fundamental porque permite que el sistema seleccione la mejor forma de realizar consultas, lo que se traduce en procesos más rápidos y eficientes.

En cuanto a la limpieza de datos, el administrador de la base de datos (DBA) a veces tiene que eliminar o archivar información que ya no se utiliza, especialmente cuando se trata de datos antiguos. Esto ayuda a que la base de datos no crezca demasiado y funcione mejor.

El mantenimiento general abarca actividades como instalar parches y actualizar el software del sistema de gestión de bases de datos (SGBD), lo cual es esencial para mantenerlo seguro y al día.

CUESTIONARIO TEMA 4

1. **¿Cuál es la función principal de los archivos log en un SGBD?**
 - a) Almacenar copias de seguridad completas de la base de datos.
 - b) Registrar todas las operaciones y transacciones para recuperación, auditoría y monitoreo.
 - c) Mejorar la velocidad de las consultas mediante índices.
2. **¿Qué propiedad ACID asegura que una transacción se completa en su totalidad o no se completa?**
 - a) Durabilidad
 - b) Atomicidad
 - c) Consistencia
3. **¿Cuál comando se utiliza para guardar de forma permanente los cambios realizados por una transacción?**
 - a) ROLLBACK
 - b) COMMIT
 - c) SAVEPOINT
4. **¿Cuál método de respaldo copia solamente los datos que han cambiado desde el último respaldo completo?**
 - a) Respaldo incremental
 - b) Respaldo diferencial
 - c) Respaldo completo
5. **¿Qué indica un índice cluster en una tabla?**
 - a) No organiza básicamente las filas, solo crea punteros a las filas.
 - b) Organiza básicamente las filas de la tabla en el orden de las claves del índice.
 - c) Solo se utiliza para vistas, no para tablas.
6. **¿Cuál de las siguientes afirmaciones sobre la reorganización de índices es correcta?**
 - a) Se utiliza para fragmentación superior al 30% y bloquea el acceso a la tabla.
 - b) Reorganiza las hojas del índice y es una operación en línea que no requiere espacio temporal adicional.
 - c) Elimina y vuelve a crear el índice desde cero.

7. ¿En qué situación se recomienda realizar una reconstrucción de índices?

- a) Cuando la fragmentación es menor al 10%.
- b) Cuando la fragmentación es alta, superior al 30%.
- c) Cuando se necesita una operación rápida sin bloqueo.

8. ¿Cuál es la diferencia principal entre un índice clúster y un índice no clúster?

- a) El índice clúster es una estructura separada de la tabla; el no cluster organiza básicamente la tabla.
- b) El índice clúster organiza básicamente las filas; el índice no clúster crea una estructura separada con punteros.
- c) Ambos índices organizan esencialmente la tabla, pero el no clúster es más rápido.

9. ¿Cuál es el nivel de fragmentación recomendado para usar la reorganización de índices?

- a) Mayor al 50%
- b) Menor al 30%
- c) Entre 40% y 60%

10. ¿Qué aspecto es fundamental en la planeación de un plan de respaldo para una base de datos?

- a) Decidir qué datos respaldar, frecuencia, almacenamiento y cómo probar los respaldos.
- b) Solo respaldar datos críticos una vez al año.
- c) Utilizar exclusivamente respaldos completos para asegurar la recuperación total.

Respuestas correctas:

- 1. b) Registrar todas las operaciones y transacciones para recuperación, auditoría y monitoreo.
- 2. b) Atomicidad
- 3. b) COMPROMETERSE
- 4. b) Respaldo diferencial

5. b) Organiza básicamente las filas de la tabla en el orden de las claves del índice.
6. b) Reorganiza las hojas del índice y es una operación en línea que no requiere espacio temporal adicional.
7. b) Cuando la fragmentación es alta, superior al 30%.
8. b) El índice clúster organiza básicamente las filas; el índice no clúster crea una estructura separada con punteros.
9. b) Menor al 30%
10. a) Decidir qué datos respaldar, frecuencia, almacenamiento y cómo probar los respaldos.

Tema 5 Seguridad

Implementa los mecanismos técnicos de seguridad para salvaguardar la información en la organización.

5.1 Espejeo (mirroring).

El **espejeo** o **mirroring** es una solución de alta disponibilidad para SQL Server que permite mantener una copia exacta y en tiempo real de una base de datos en un servidor de respaldo (servidor espejo). Esto garantiza la continuidad del servicio en caso de fallo del servidor principal. Es una tecnología que fue popular en SQL Server 2005 hasta la versión 2012, pero ha sido **sustituida** por los Grupos de Disponibilidad de AlwaysOn a partir de SQL Server 2012 Enterprise.

Componentes principales

Un entorno de espejeo básico consta de tres servidores:

1. **Servidor Principal (Principal Server):** Contiene la copia activa y de lectura/escritura de la base de datos. Todas las transacciones de la aplicación se realizan aquí.
2. **Servidor Espejo (Mirror Server):** Contiene la copia pasiva de la base de datos. Solo recibe los registros de transacciones del servidor principal y los aplica para mantener la base de datos sincronizada. No se puede acceder a la base de datos espejo para consultas, a menos que esté en modo de sólo lectura.
3. **Testigo (Witness):** Este es un servidor opcional pero recomendado. Su función es monitorizar la conexión entre el servidor principal y el espejo. En caso de que el servidor principal falle, el testigo actúa como un árbitro y permite que el servidor espejo asuma automáticamente el rol de principal (failover automático).

Modos de operación

El espejeo puede funcionar en dos modos principales, que determinan el nivel de seguridad y rendimiento.

1. **Modo de alta seguridad (High-Safety Mode):**
 - **Sincronización:** Las transacciones se confirman en el servidor principal **solo después de que se han confirmado en el servidor espejo**. Esto garantiza que no haya pérdida de datos.
 - **Rendimiento:** Hay una ligera latencia en las transacciones, ya que cada una debe esperar la confirmación del servidor espejo.

- **Failover:** Con un testigo, el failover (la transferencia de roles de servidor) es **automático** en caso de fallo del servidor principal.

2. Modo de alto rendimiento (High-Performance Mode):

- **Sincronización:** Las transacciones se confirman en el servidor principal **sin esperar la confirmación del servidor espejo**. El registro de transacciones se envía de forma asíncrona.
- **Rendimiento:** Este modo tiene un mejor rendimiento ya que no hay latencia de confirmación.
- **Failover:** El failover **no es automático**. Si el servidor principal falla, puede haber una pequeña pérdida de datos no confirmados en el servidor espejo.

¿Cómo se configura?

El espejeo se configura mediante el uso de los **puntos de conexión (endpoints)** en SQL Server. Se debe configurar cada servidor para que se comuniquen con los otros a través de un endpoint de espejeo.

Desventajas

Solamente se puede espejear una base de datos por cada instancia de SQL Server. Es importante mencionar que esta opción no brinda redundancia a nivel del servidor completo, sino únicamente para la base de datos seleccionada. Además, esta funcionalidad ha sido sustituida por los Grupos de Disponibilidad AlwaysOn, los cuales representan una alternativa mucho más robusta y completa para asegurar alta disponibilidad y recuperación ante desastres.

Ejercicio

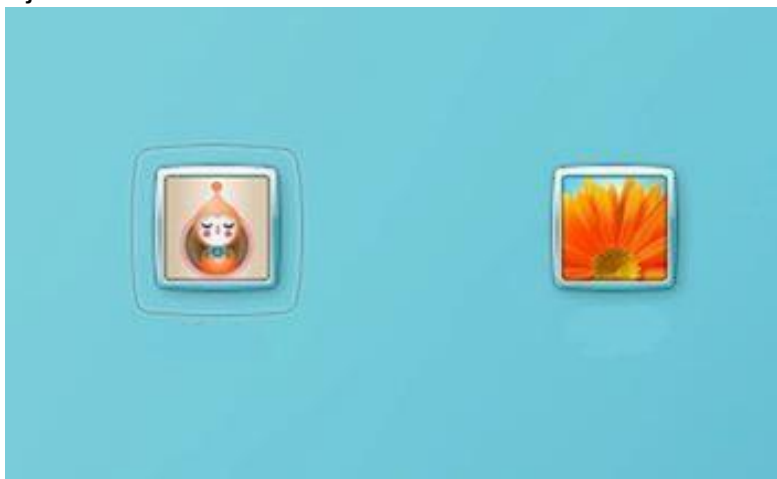


Figura 23 Usuarios

Para que el espejeo sea satisfactorio ambas instancias de sqlserver deben estar iniciadas con el mismo usuario, por lo que se realiza una cuenta que tendrá permisos de administrador para no utilizar la cuenta principal de la máquina.

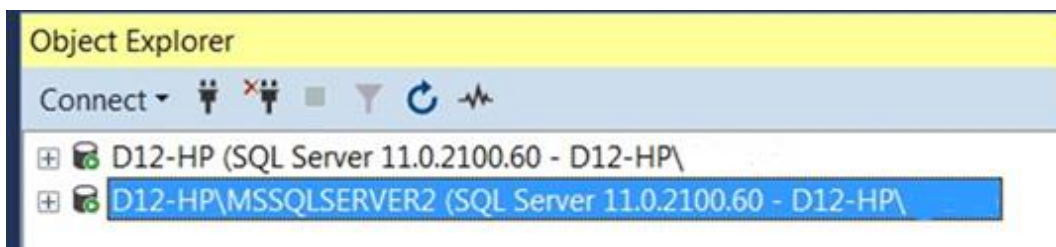


Figura 24 Instancias

Iniciamos ambas instancias en el mismo equipo de cómputo y con el mismo usuario.

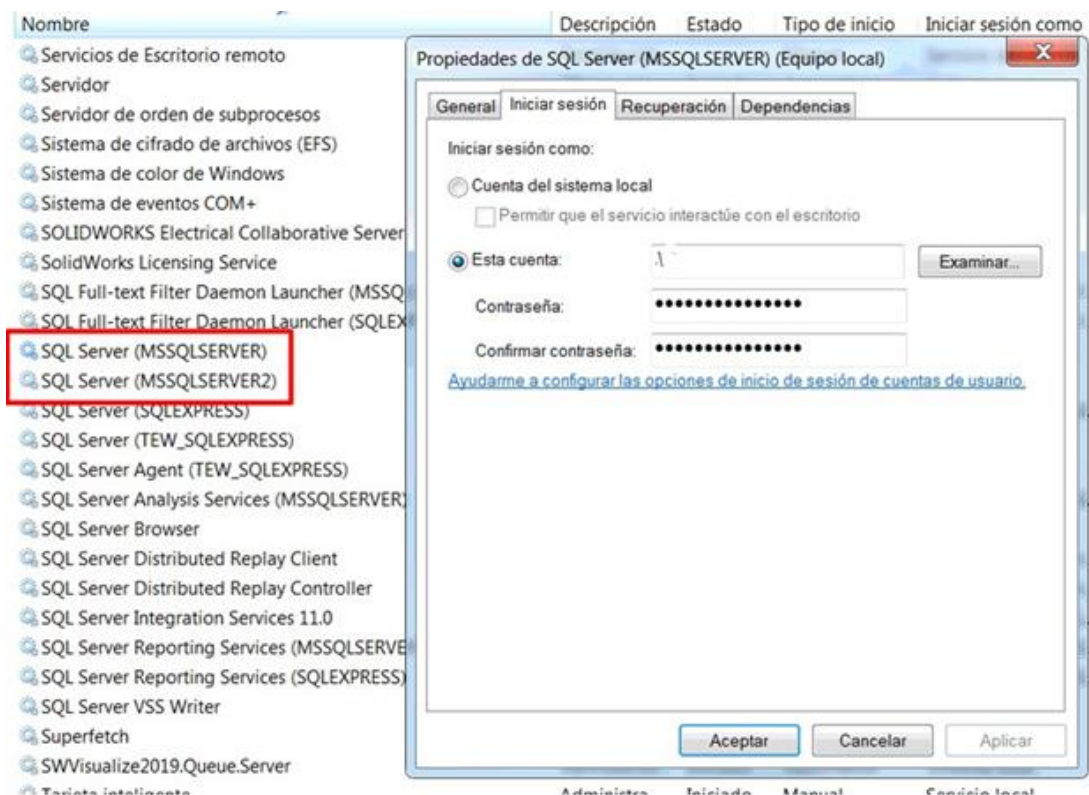


Figura 25 Instancias creadas

En los servicios, hay que cambiar las propiedades de las instancias que usaremos, cambiando la cuenta utilizando la cuenta que se creó anteriormente.

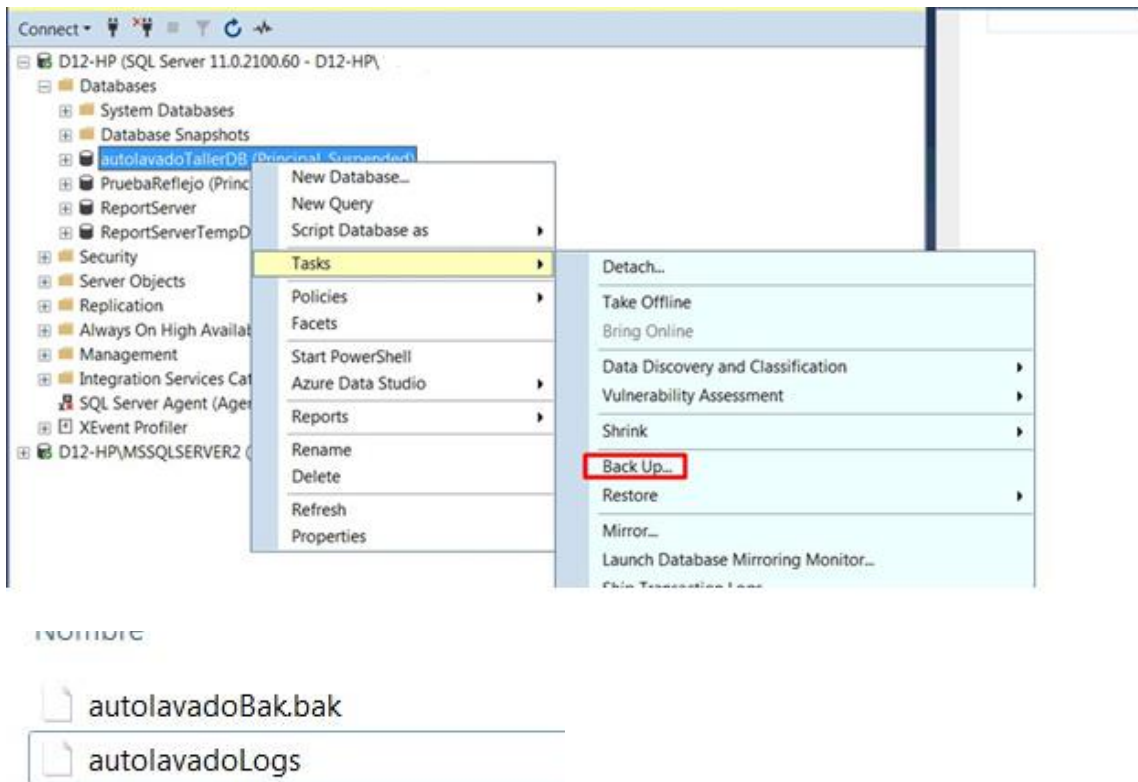


Figura 26 Respaldo

Es importante que se cree un respaldo nuevo de .bak, en el instante para que la base de datos sea idéntica en la máquina espejo.

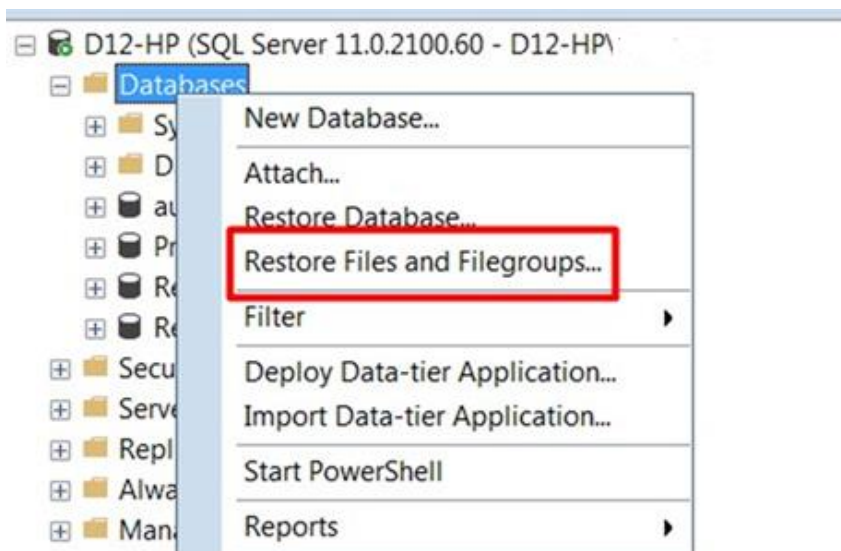


Figura 27 Restaurar archivos

Cuando se tengan los archivos ubicados respaldamos el .bak, y se tendrá que restaurar desde la base de datos:

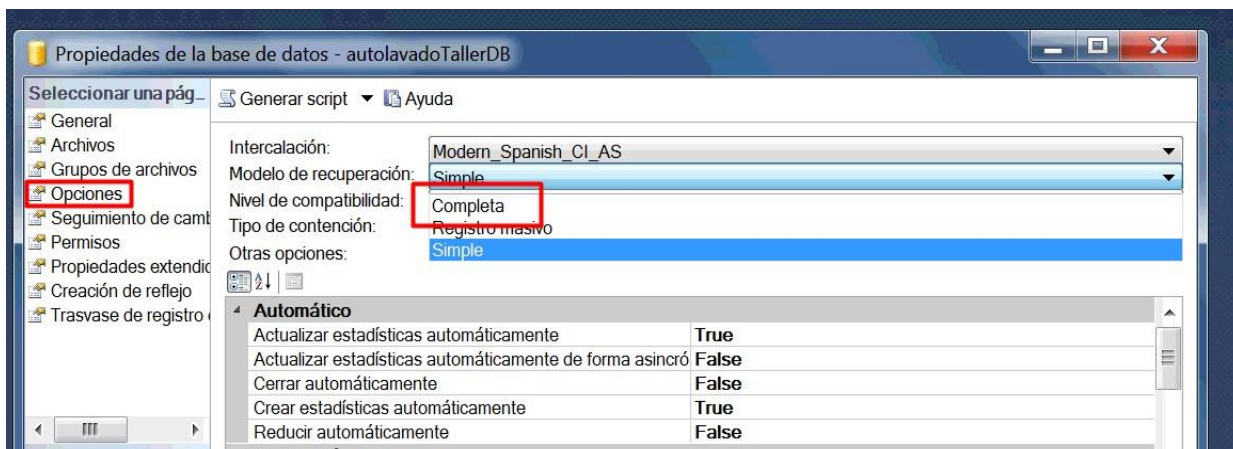


Figura 28 BD completa

Aseguremos que el tipo de recuperación sea completa en la base de datos.

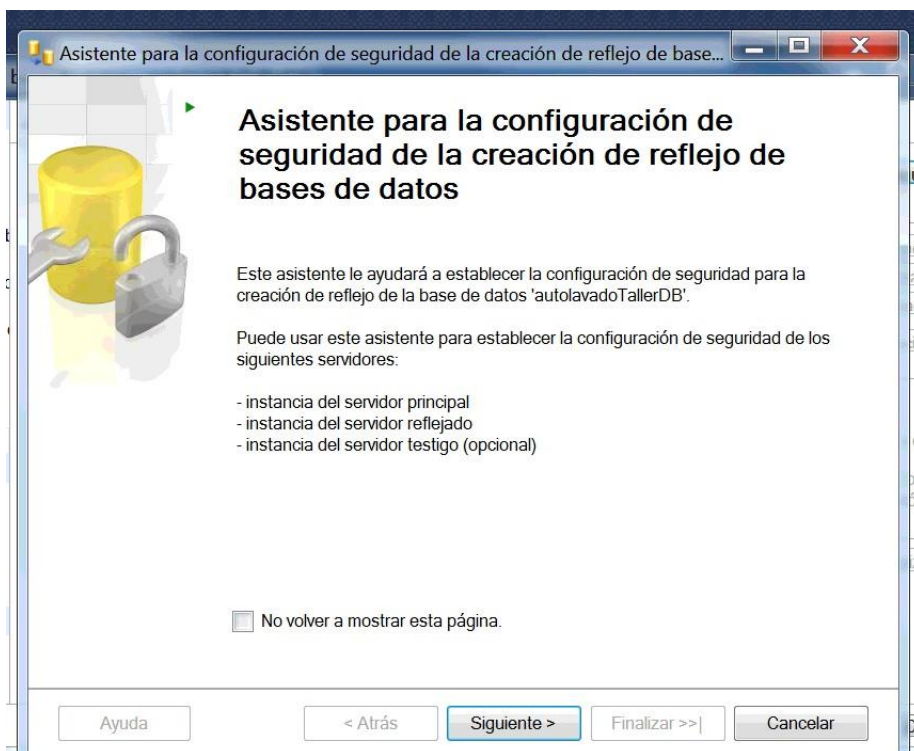


Figura 29 Espejo de BD

Entramos a las propiedades de la base de datos principal, luego en la sección de espejo iniciamos una nueva configuración de seguridad.

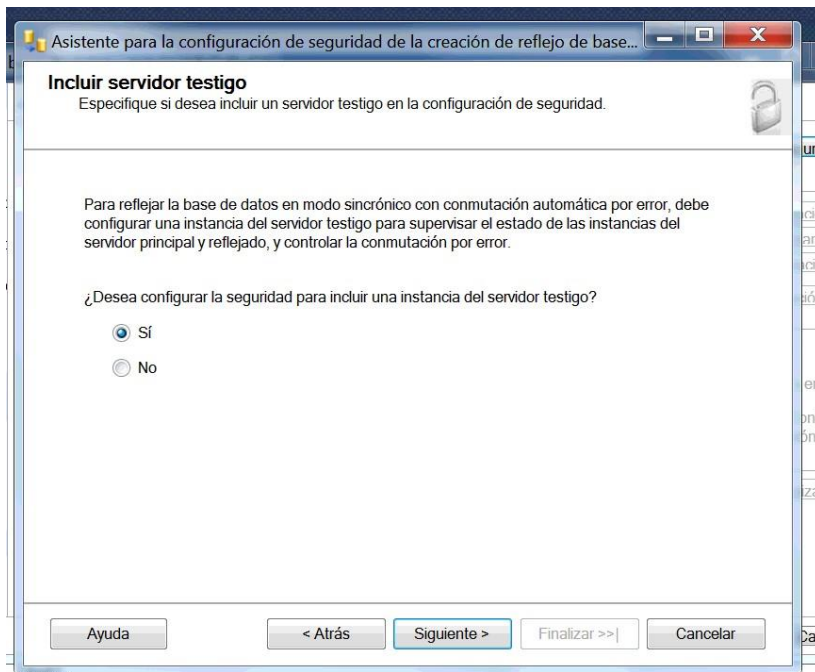


Figura 30 Servidor testigo

Seleccionamos que no queremos un servidor testigo, en este caso.

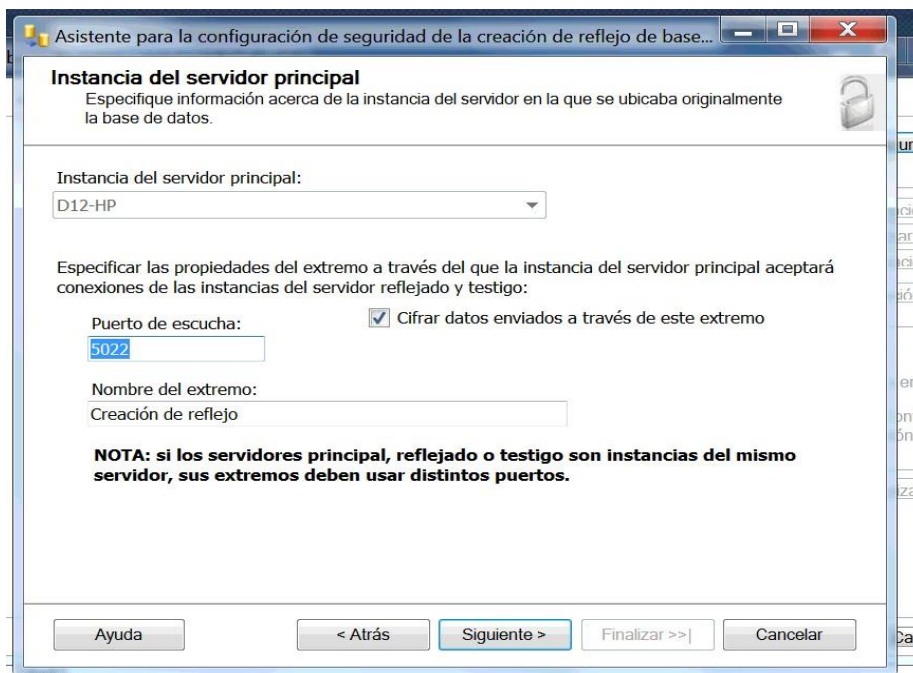


Figura 31 Puerto 5022

Automáticamente detecta que para el servidor principal asigna el puerto 5022. Si ambas instancias están en la misma computadora el puerto será diferente, en caso contrario ambos puertos serán el 5022.

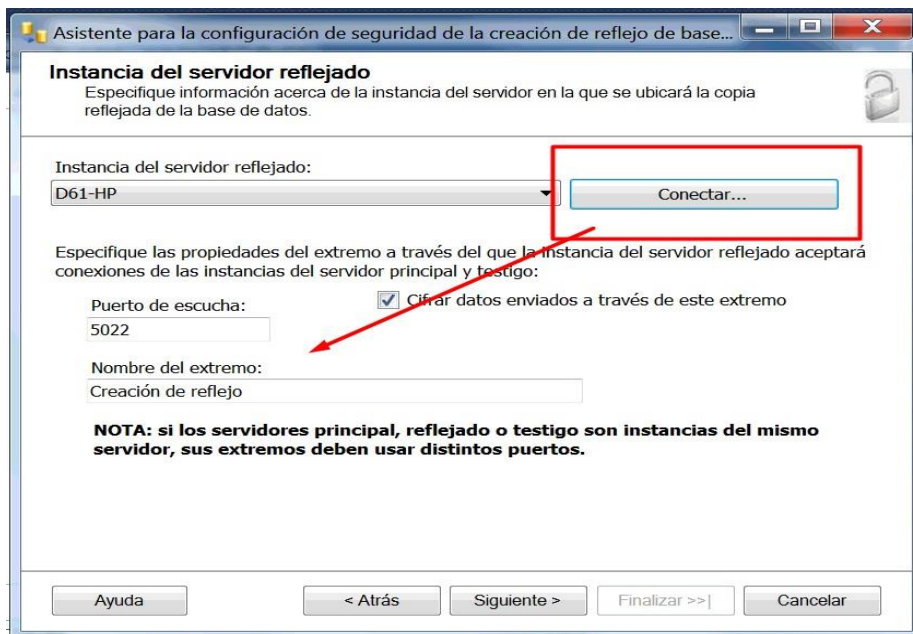


Figura 32 Creación de reflejo

Nos conectamos a la segunda máquina, y nos aseguramos de que esté todo correcto, y finalizamos.

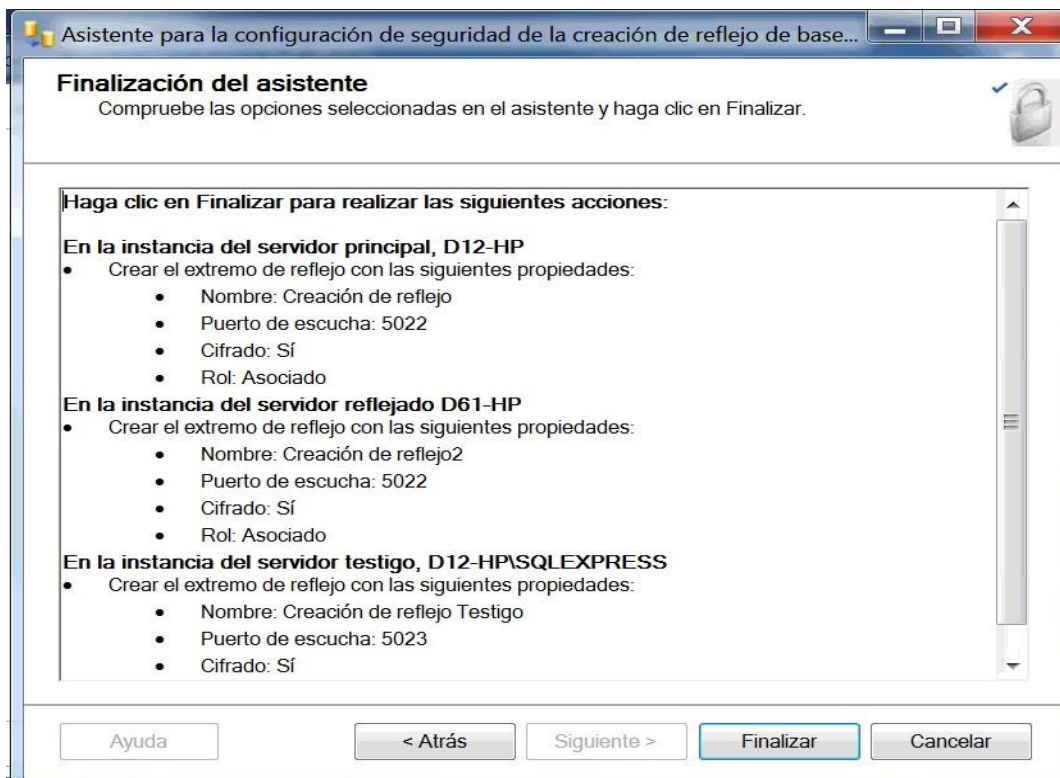


Figura 33 Finalización de asistente

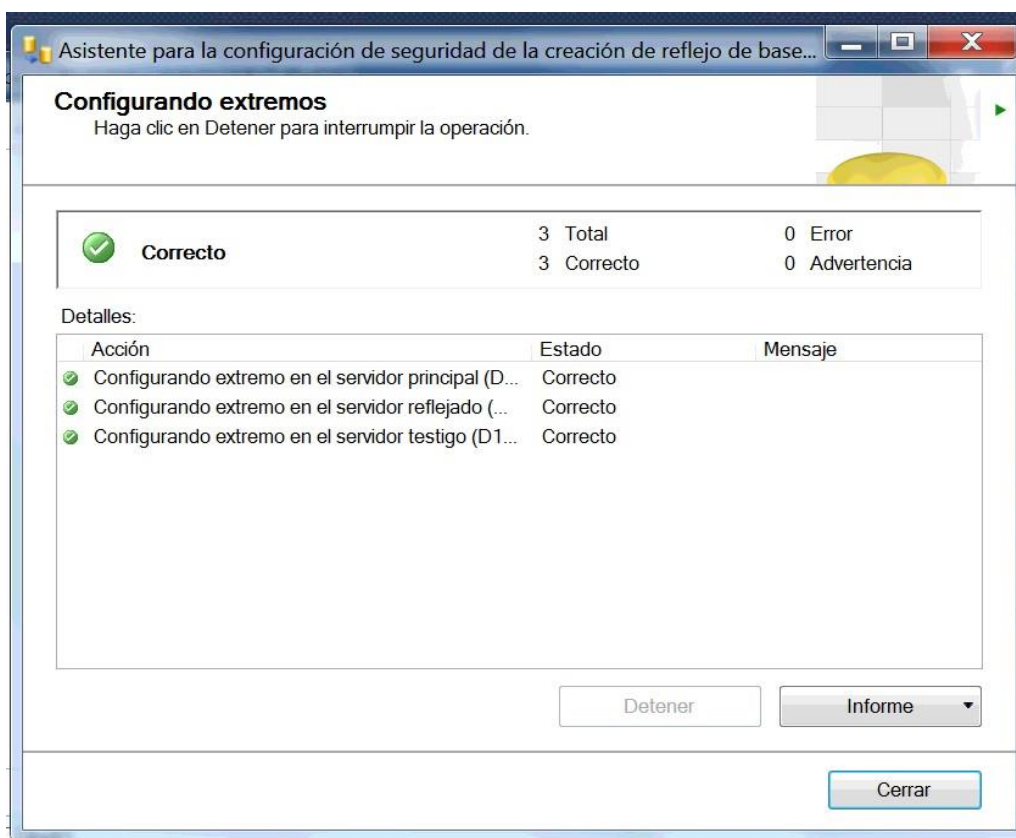


Figura 34 Configurando extremos

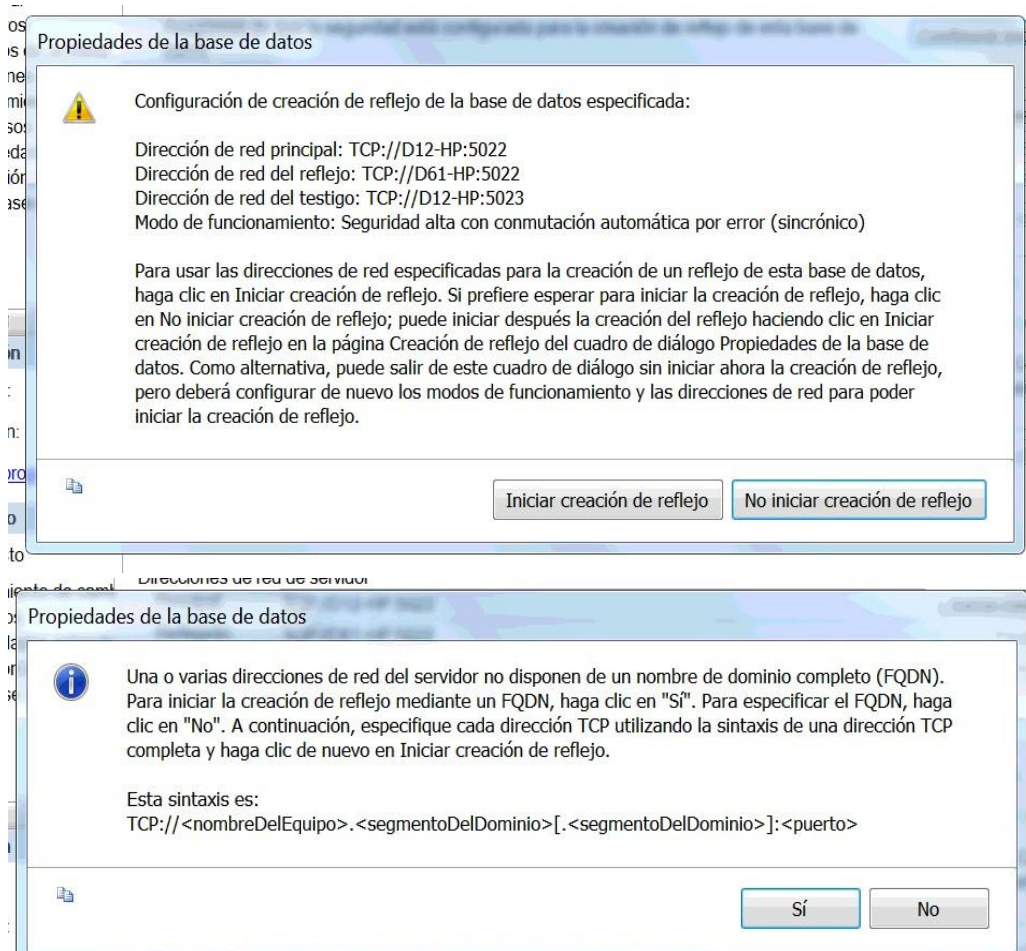


Figura 35 Proceso de creación

Después de esto debería de iniciar el espejeo, y en el nombre del base de datos nos aparecerá la leyenda de que está sincronizada para la base de datos principal y también en el espejo.

Server network addresses

Principal:	TCP://D12-HP:5022	Start Mirroring Pause Remove Mirroring Failover
Mirror:	TCP://D12-HP:5023	
Witness:		

Note: Use fully-qualified TCP addresses. For example:
TCP://svr5.corp.abc.com:5022

Operating mode

☐ High performance (asynchronous) -- Commit changes at the principal and then transfer them to the mirror.
☒ High safety without automatic failover (synchronous) -- Always commit changes at both the principal and mirror.
☐ High safety with automatic failover (synchronous) -- Requires a witness server instance. Commit changes at both the principal and mirror if both are available. The witness controls automatic failover to the mirror if the principal becomes unavailable.

Status: Synchronized: the databases are fully synchronized

Figura 36 High safety

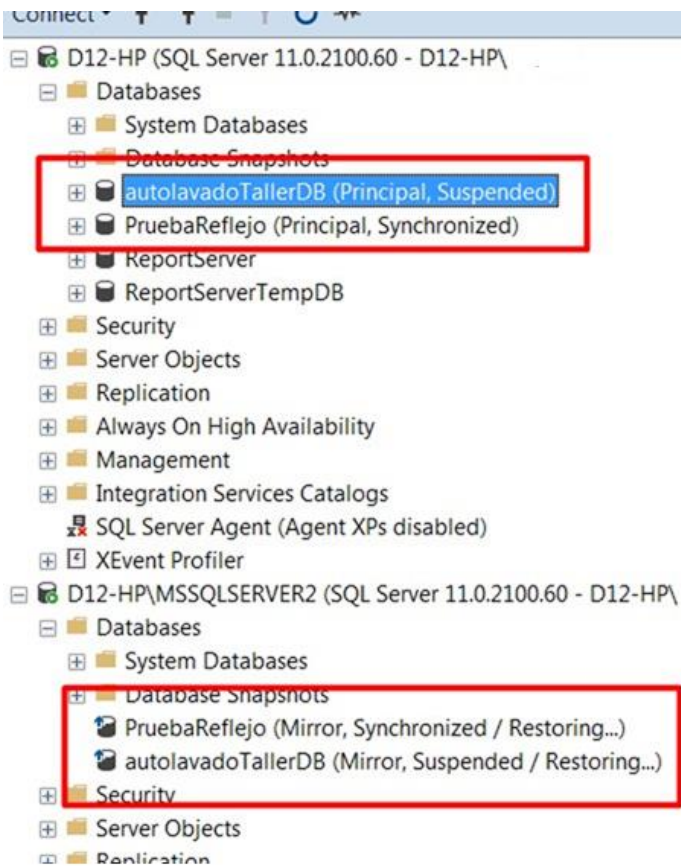


Figura 37 Prueba de funcionamiento

5.2 Réplica (replication).

La **réplica** o **replication** en SQL Server es un conjunto de tecnologías para copiar y distribuir datos y objetos de bases de datos de una base de datos a otra, para luego sincronizar ambas bases de datos. Es una solución de alta disponibilidad y escalabilidad que permite a los usuarios trabajar con una copia de datos, mientras que el servidor de origen no se ve afectado.

A diferencia del mirroring, que solo funciona a nivel de base de datos completa, la réplica permite un control más granular, pudiendo replicar tablas, vistas, o incluso solo ciertos datos dentro de una tabla.

Componentes clave

1. **Publicador (Publisher):** Es el servidor de origen que hace que los datos y los objetos estén disponibles para la réplica. El publicador identifica qué datos han cambiado y los envía al distribuidor.
2. **Distribuidor (Distributor):** Actúa como una oficina de correos.  Recibe los datos replicados del publicador y los almacena en su base de datos de distribución (distribution database). Luego, los distribuye a los suscriptores.
3. **Suscriptor (Subscriber):** Es el destino de la réplica. Recibe los datos del distribuidor y los almacena en una base de datos local. El suscriptor puede ser de sólo lectura o de lectura/escritura, dependiendo del tipo de réplica.

Tipos de réplica

SQL Server ofrece varios tipos de réplica, cada uno adaptado a diferentes necesidades de negocio.

1. **Réplica de Instantáneas (Snapshot Replication):**
 - **Funcionamiento:** Toma una "fotografía" completa de la base de datos o de los objetos seleccionados en un momento dado y la distribuye a los suscriptores.
 - **Uso:** Ideal para bases de datos que no cambian con frecuencia. No es una réplica en tiempo real, ya que solo se actualiza cuando se crea una nueva instantánea.
2. **Réplica Transaccional (Transactional Replication):**
 - **Funcionamiento:** Replica los cambios de datos (inserciones, actualizaciones, eliminaciones) a medida que ocurren. Los datos se replican en tiempo casi real.

- **Uso:** Es el tipo más común para la alta disponibilidad. Se utiliza para sincronizar bases de datos en tiempo real, como en sistemas de comercio electrónico o en entornos de reporting.

3. Réplica de Mezcla (Merge Replication):

- **Funcionamiento:** Permite que los cambios se realicen tanto en el publicador como en los suscriptores. Cuando las bases de datos se conectan, los cambios se fusionan para que todas las copias estén sincronizadas.
- **Uso:** Ideal para entornos de informática móvil o para usuarios que necesitan trabajar sin conexión. Es más complejo de gestionar debido a los posibles conflictos de datos.

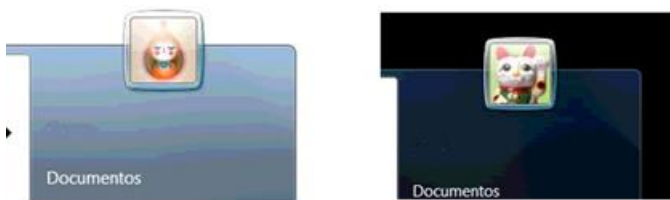


Figura 38 Cuenta Administrador

Para la replicación necesitamos crear preferentemente la misma cuenta administrador en ambos equipos (mismo nombre y contraseña de preferencia).

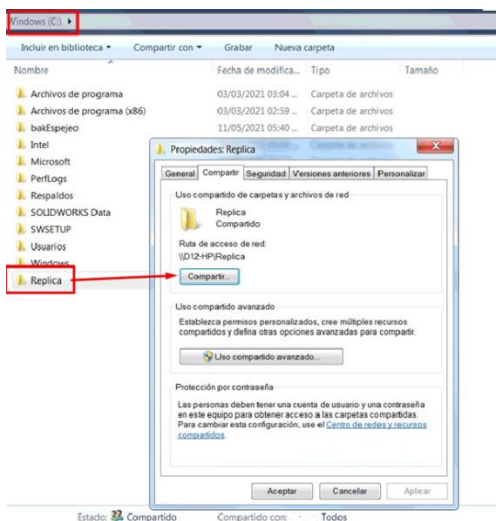


Figura 39 Carpeta compartida

Crearemos una carpeta compartida que funcionará como intermediario entre ambas máquinas para que el espejeo ocurra a través de aquí. (Compartir la carpeta a todos o a la segunda maquina específicamente.)

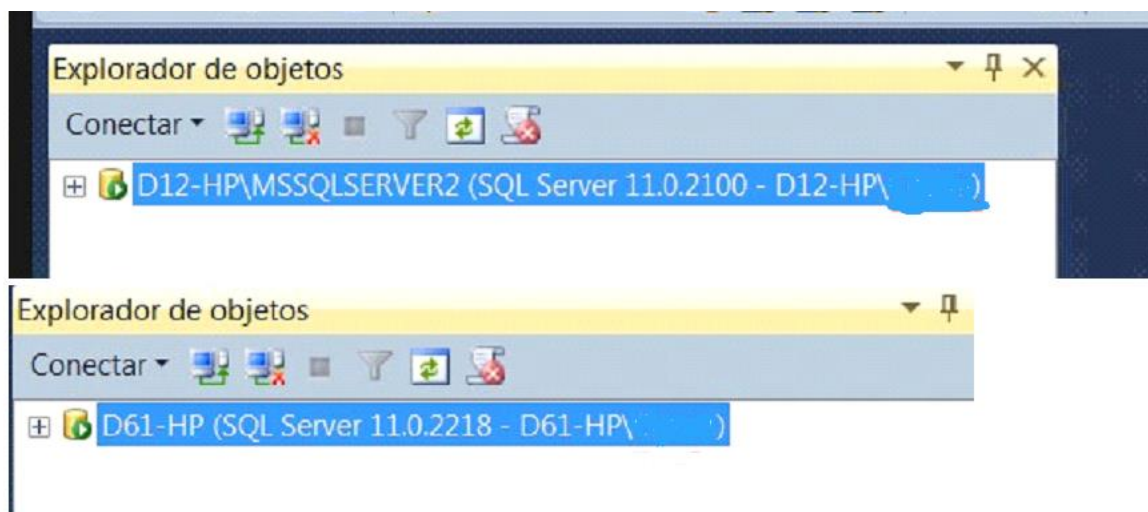


Figura 40 Conexión ambos motores

Desde la computadora principal nos conectamos a ambos motores de cada máquina respectivamente (difusor y suscriptor).

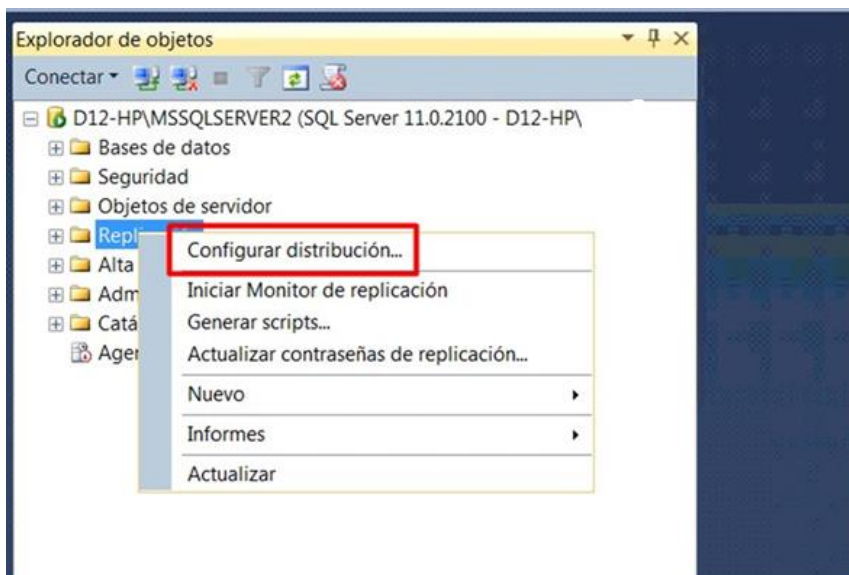


Figura 41 Configuración de distribución

Desde la máquina principal configuramos la distribución.

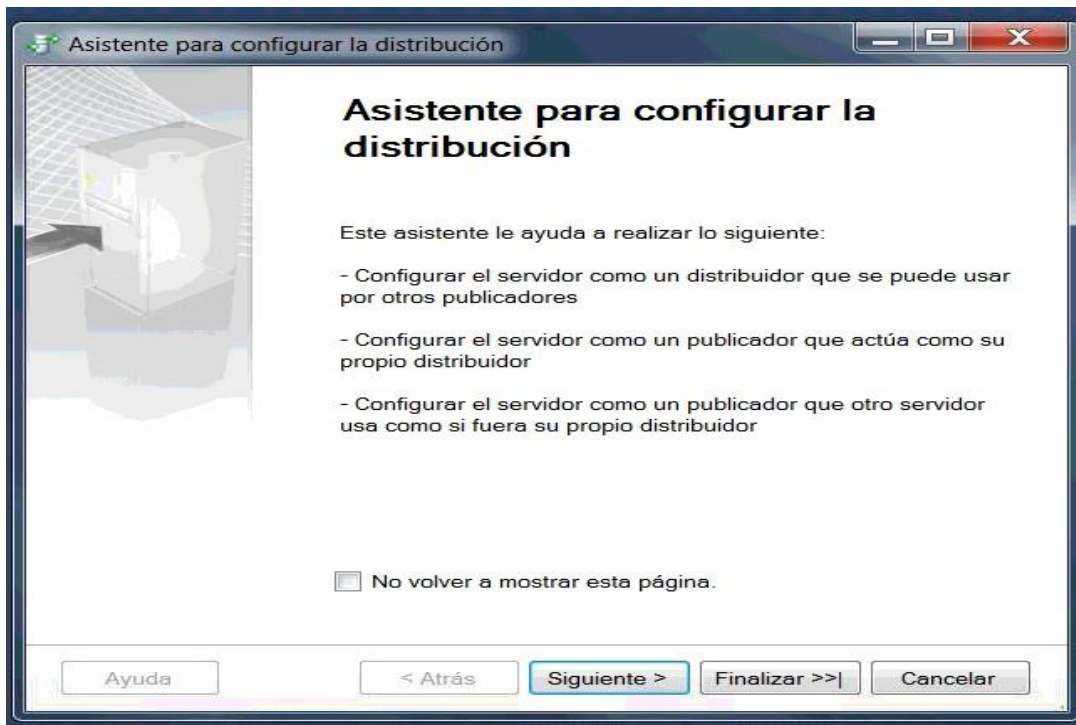


Figura 42 Siguiente

Seleccionamos siguiente.

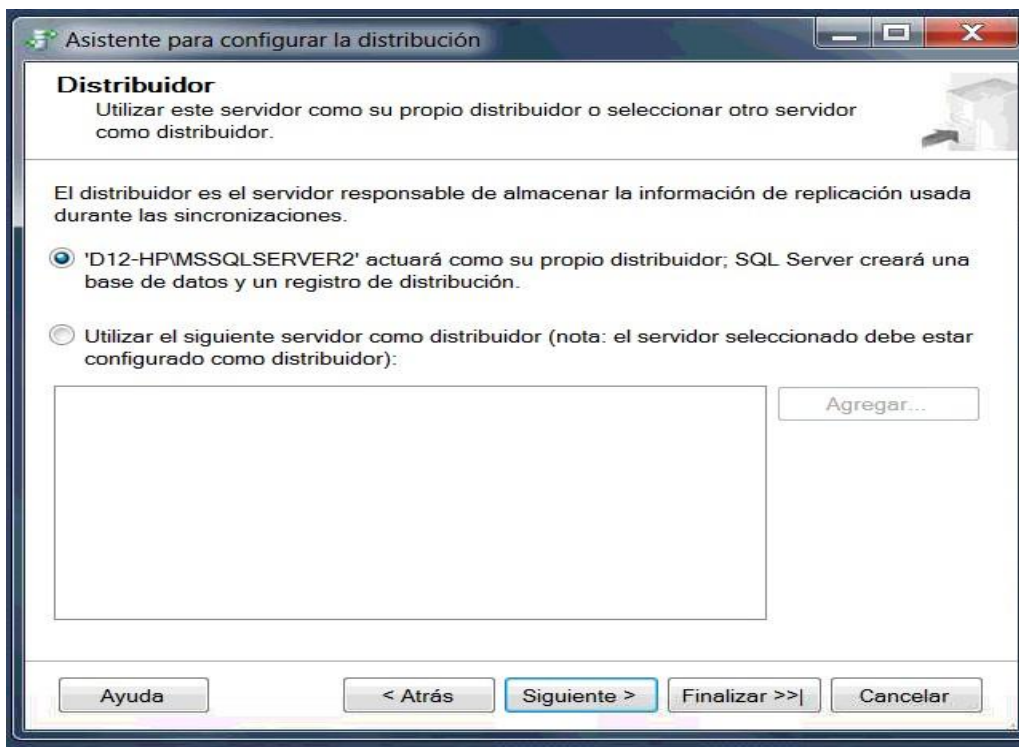


Figura 43 Selección de primera opción

Seleccionamos la primera opción.

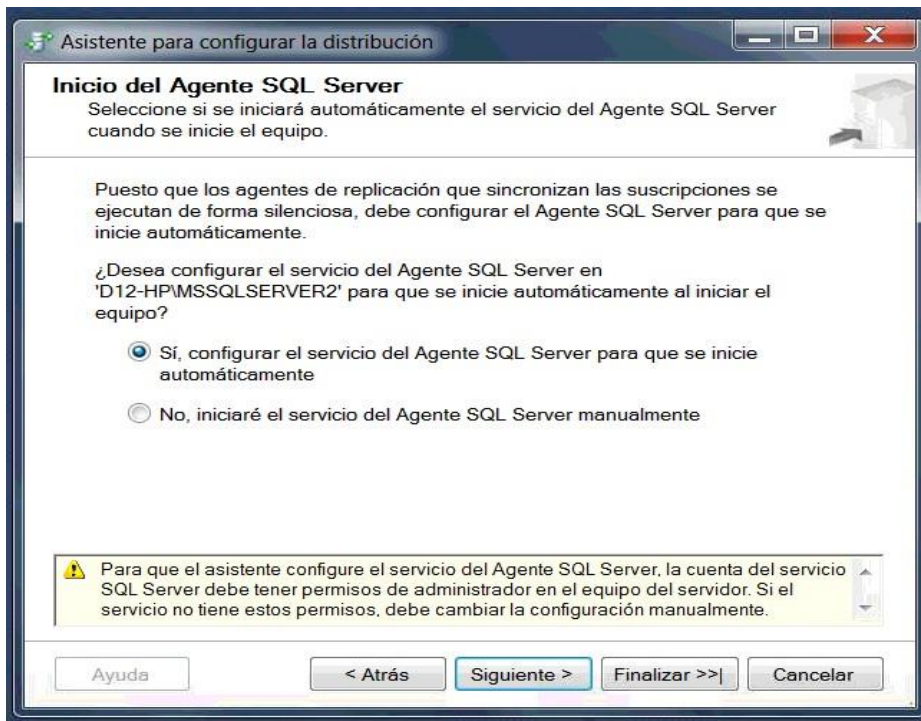


Figura 44 Inicio automático

Configuramos para que inicie automáticamente.

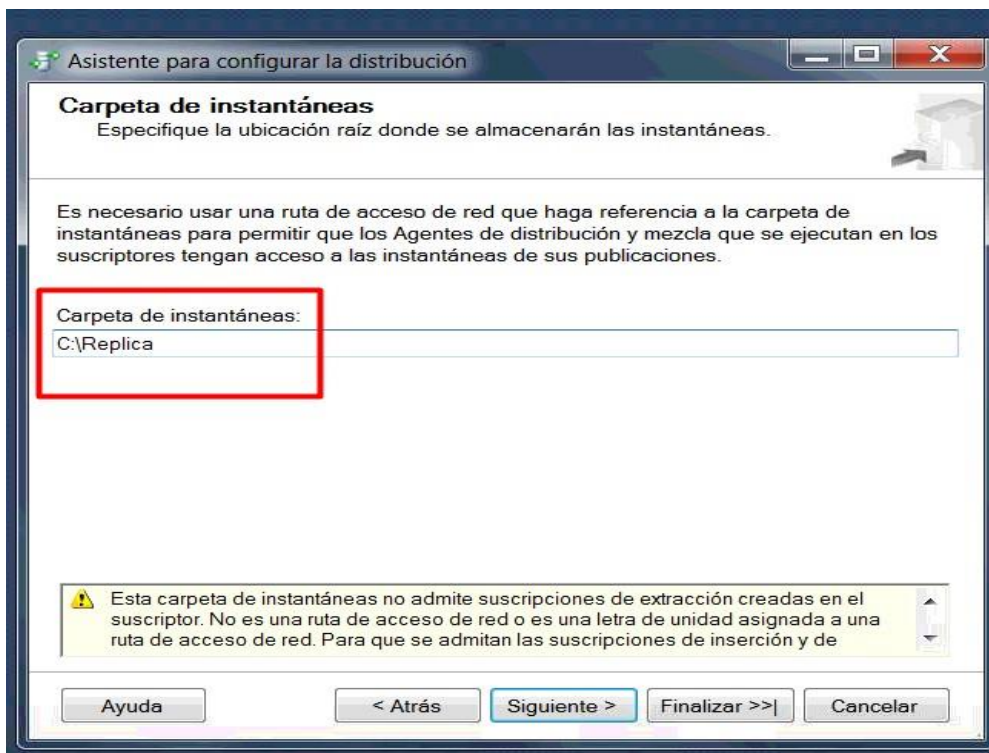


Figura 45 Selección de carpeta compartida

Seleccionamos la carpeta compartida que creamos anteriormente.

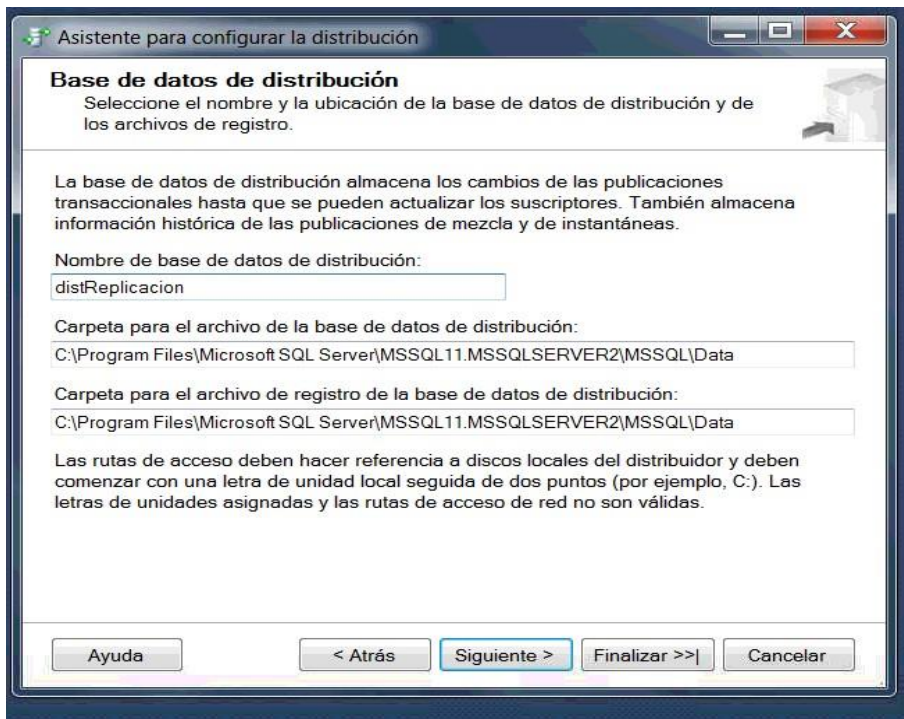


Figura 46 Asignación de nombre

Colocamos un nombre a nuestra replicación.

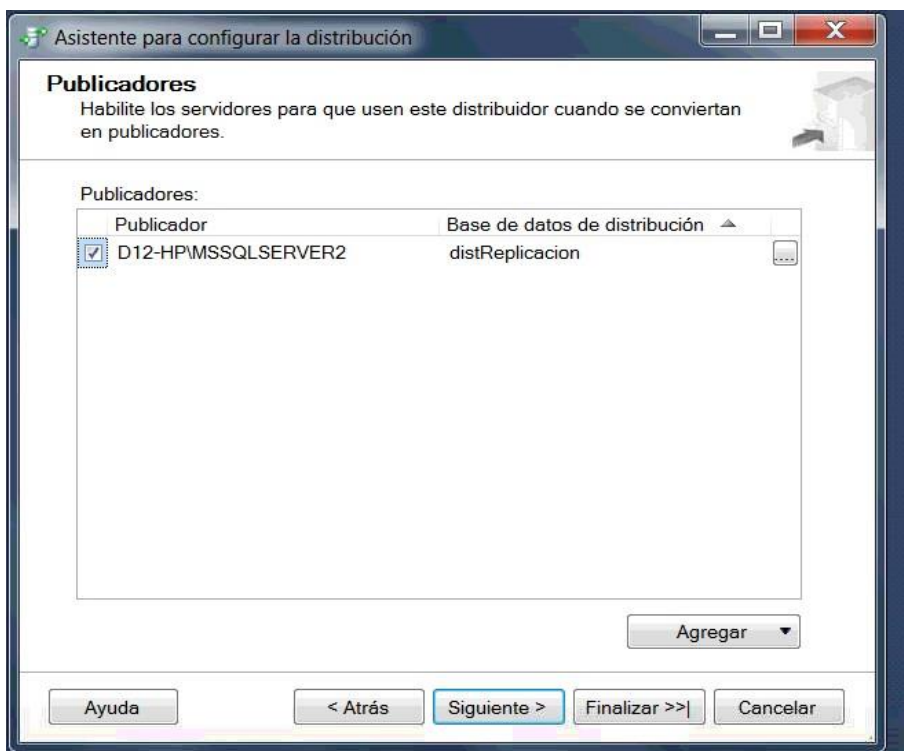


Figura 47 Publicador

Seleccionamos el publicador, le damos siguiente.

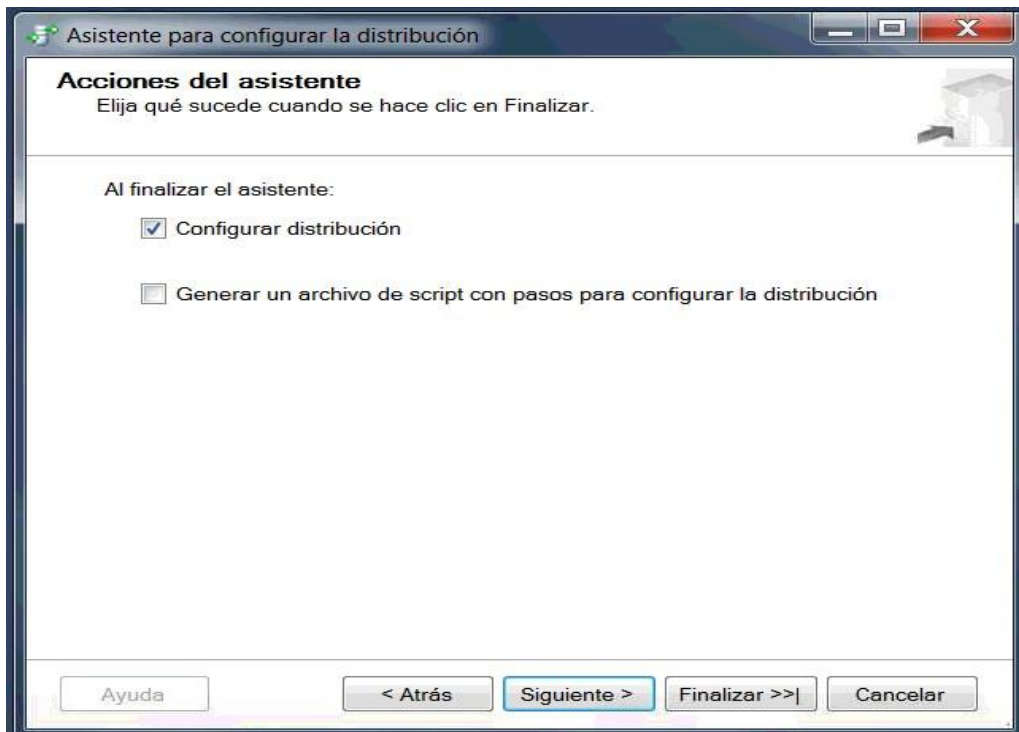


Figura 48 Distribución

Configurar distribución.

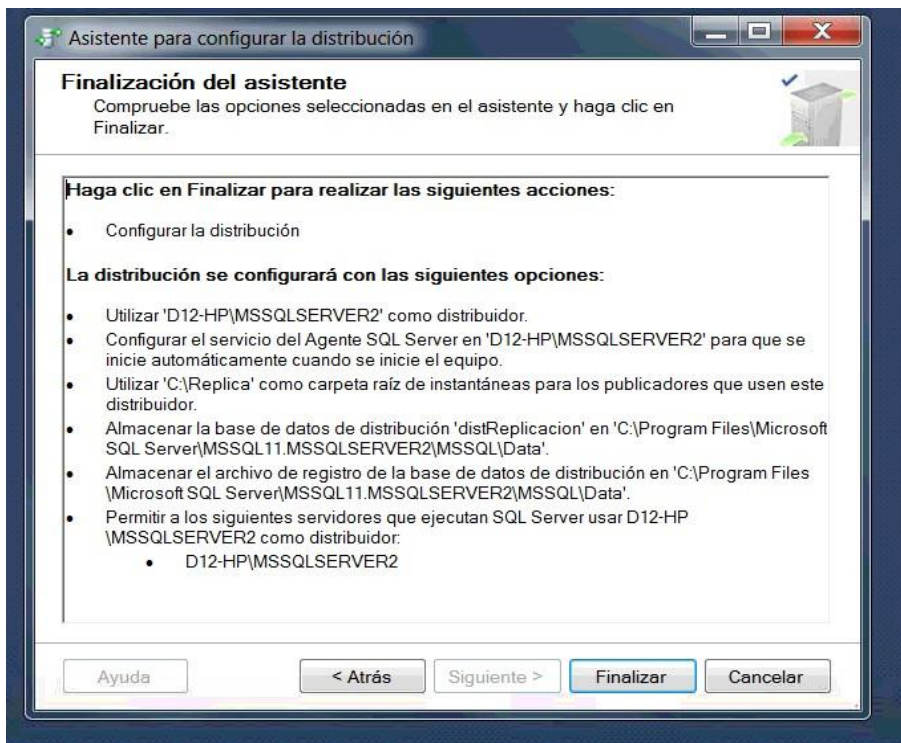


Figura 49 Finalización de configuración

Finalizamos la configuración de distribuidor.

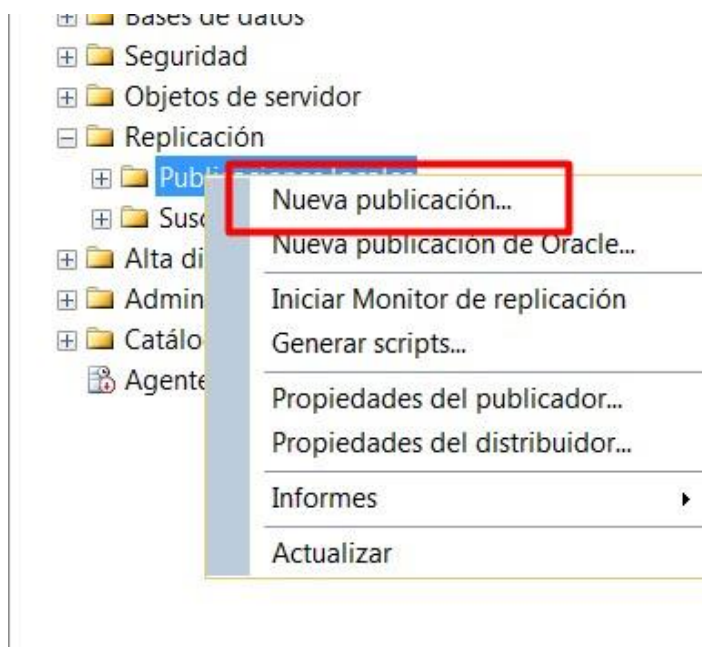


Figura 50 Nueva publicación

Ahora podemos crear una nueva publicación.

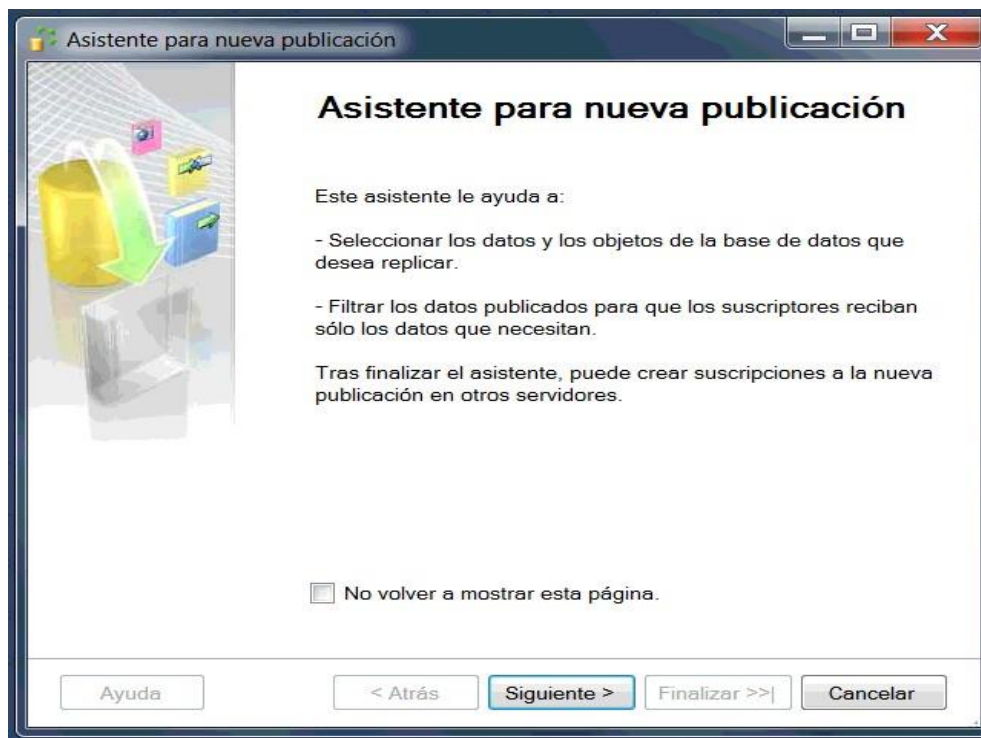


Figura 51 Asistente de configuración

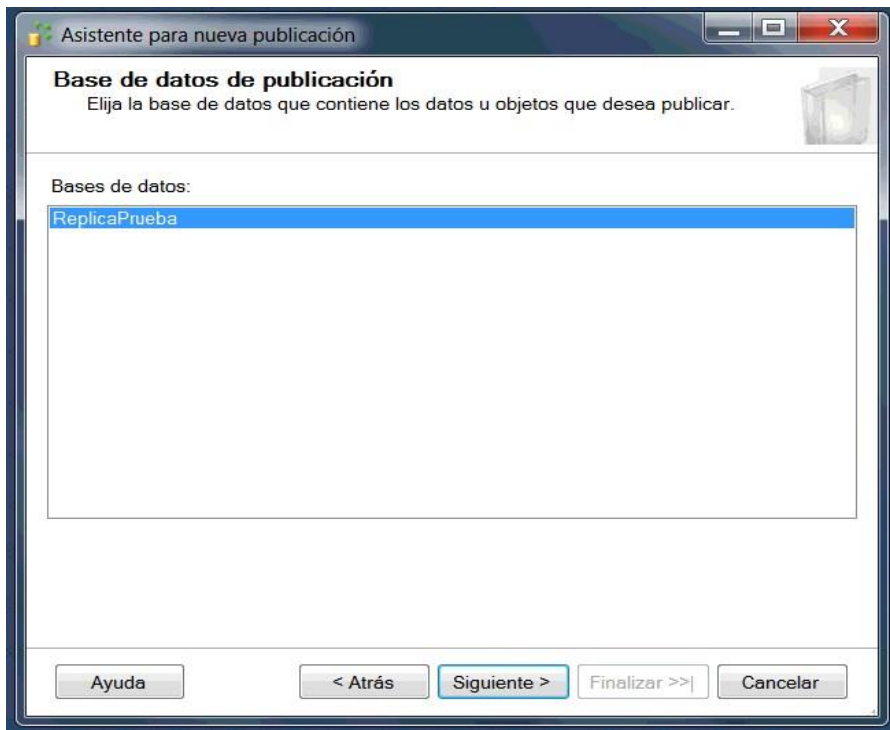


Figura 52 Selección de BD

Seleccionamos la base de datos a la que haremos espejeo en la segunda máquina.

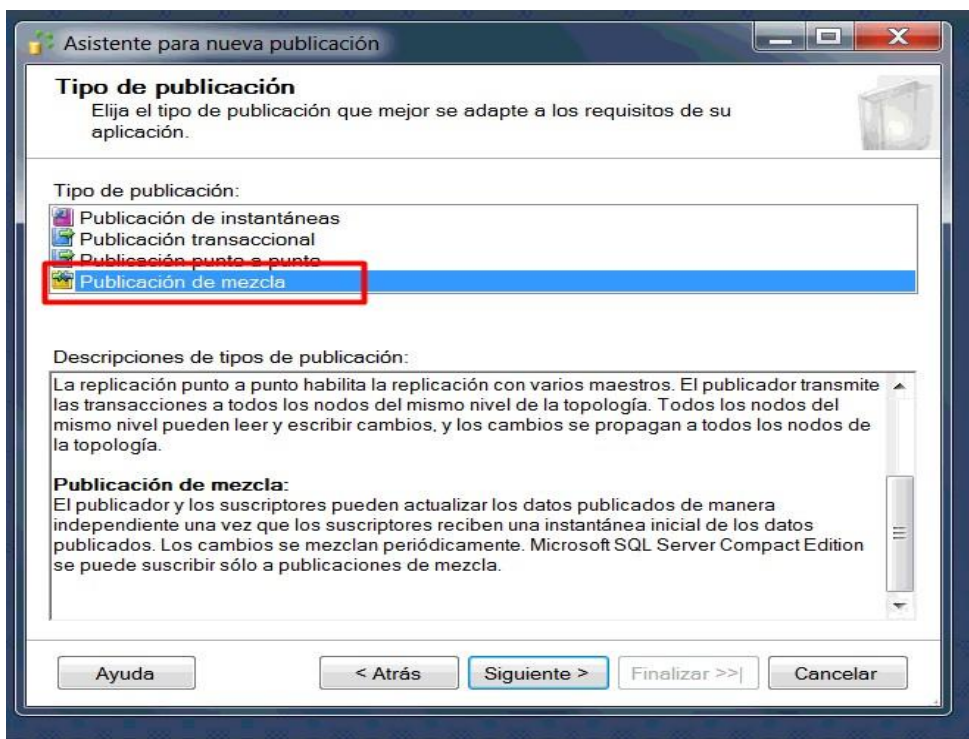


Figura 53 Publicación de mezcla

Seleccionamos publicación de mezcla.

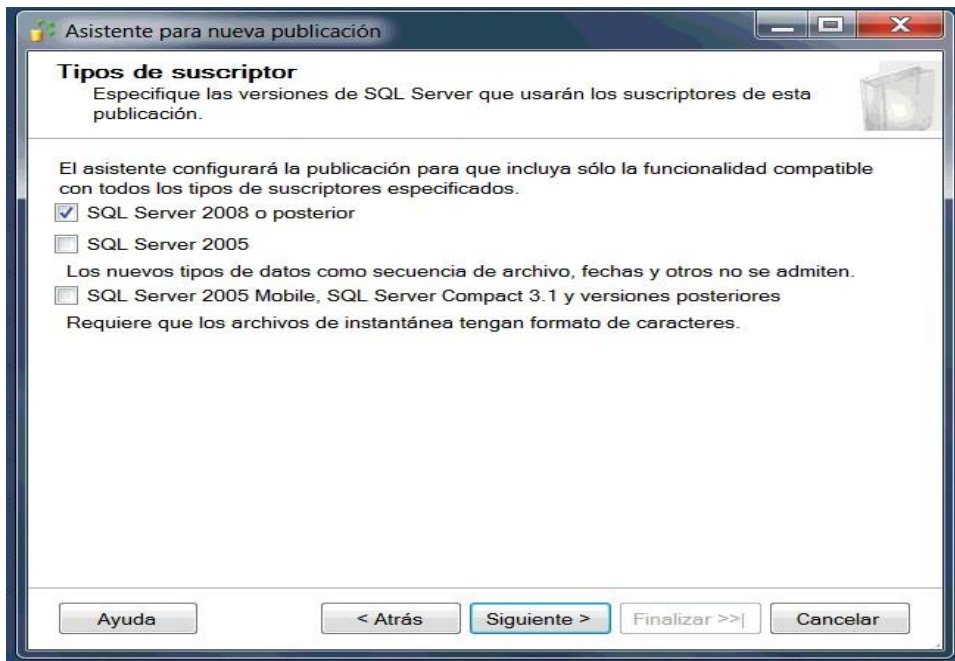


Figura 54 Sql 2008

Seleccionamos sql 2008 o posterior.

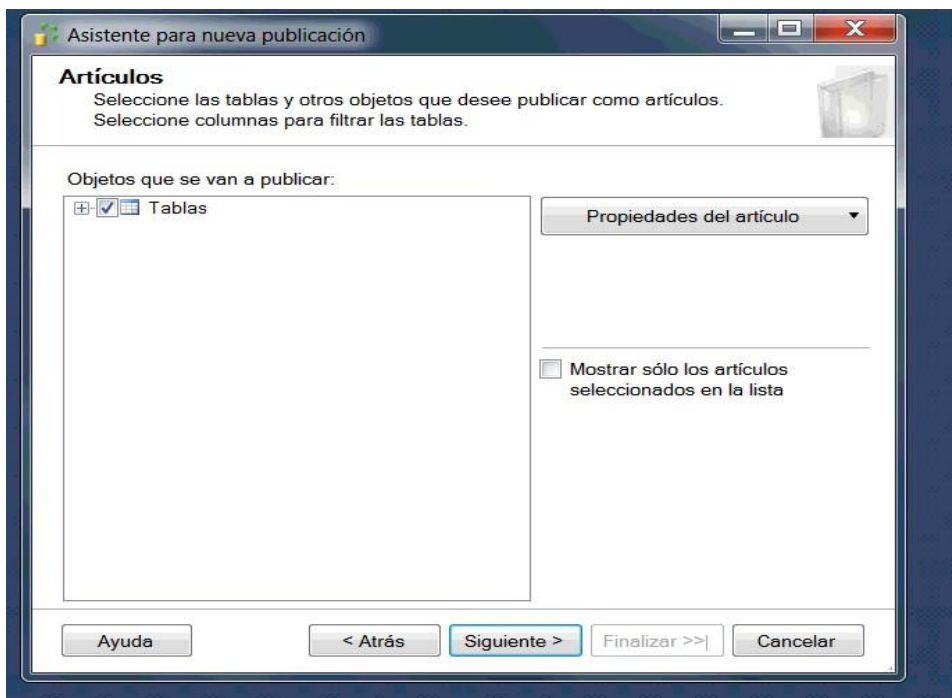


Figura 55 Selección de tablas

Nos aseguramos que todas las tablas deseadas estén seleccionadas.

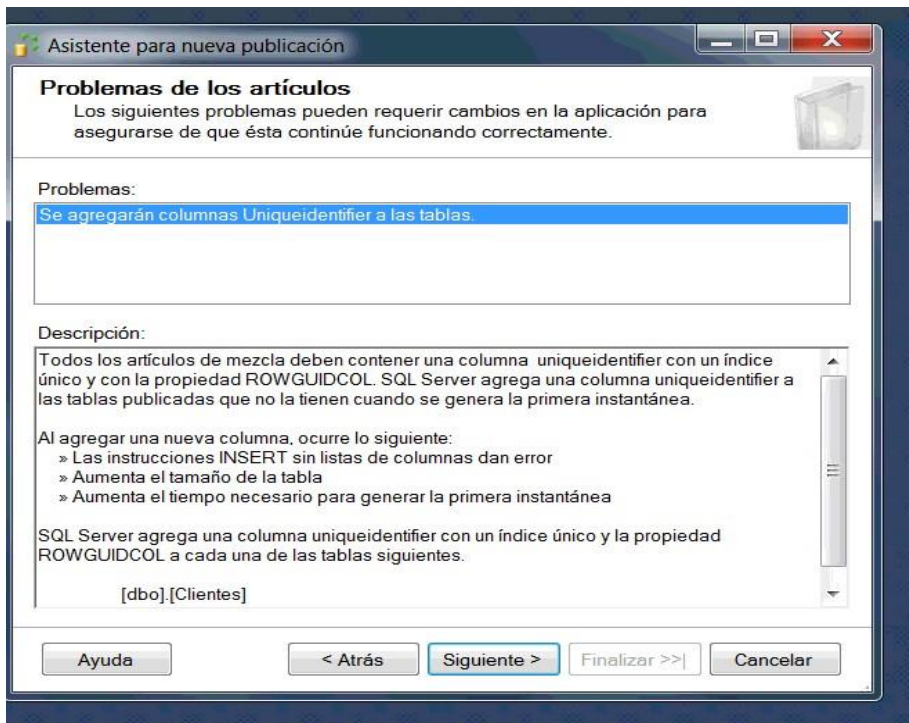


Figura 56 Continuación

Siguiente



Figura 57 Agregar filtros

Si se desea, se agregan filtros

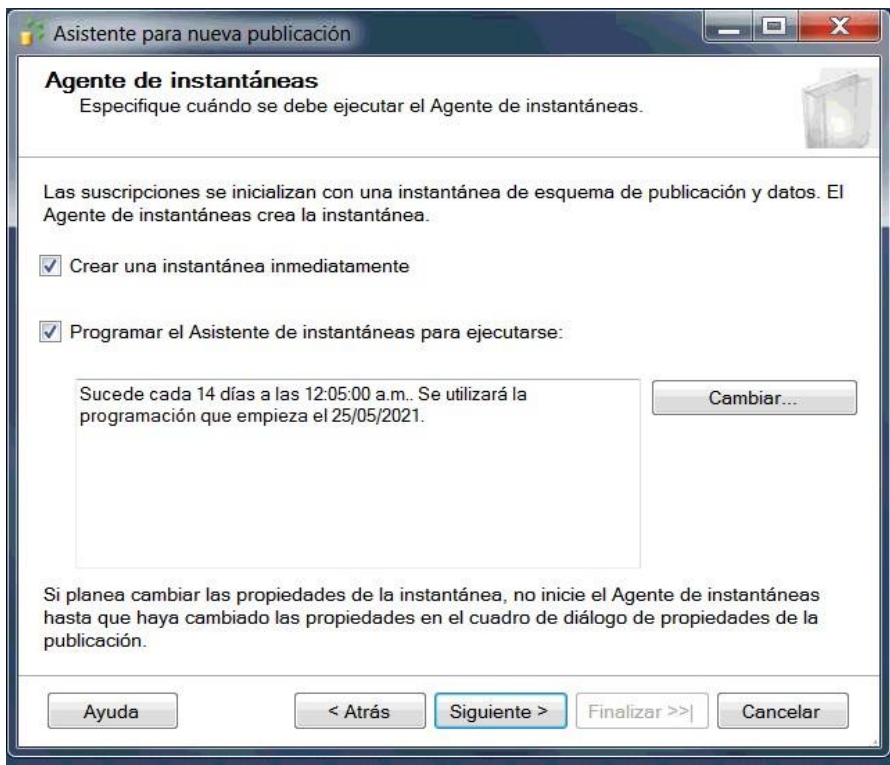


Figura 58 Replica

Importante seleccionar cada cuándo queremos que la replicación suceda, por defecto sucede cada 14 días, entonces para ver efectos inmediatos podemos cambiar el tiempo a 1 segundo, 1 minuto, etc.

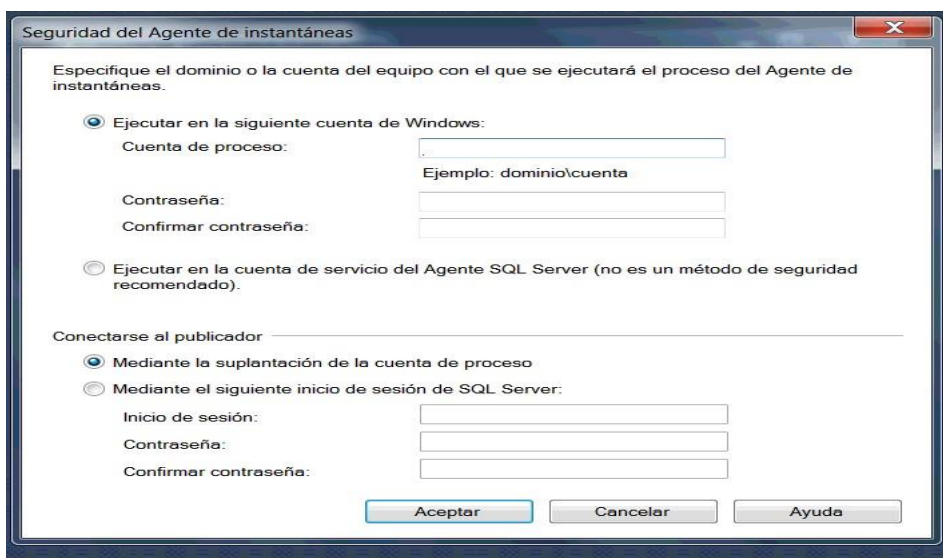


Figura 59 Cuenta de windows

Hay que agregar la cuenta de Windows responsable de contener el proceso.

Seguridad del Agente de instantáneas

Especifique el dominio o la cuenta del equipo con el que se ejecutará el proceso del Agente de instantáneas.

☒ Ejecutar en la siguiente cuenta de Windows:

Cuenta de proceso: D12-HP\

Ejemplo: dominio\cuenta

Contraseña: *****

Confirmar contraseña: *****

☐ Ejecutar en la cuenta de servicio del Agente SQL Server (no es un método de seguridad recomendado).

Conectarse al publicador

☒ Mediante la suplantación de la cuenta de proceso

☐ Mediante el siguiente inicio de sesión de SQL Server:

Inicio de sesión:

Contraseña:

Confirmar contraseña:

Aceptar Cancelar Ayuda

Figura 60 Maquina principal

En este caso la cuenta de chapa de la maquina 12 (la principal).

Asistente para nueva publicación

Seguridad del agente

Para cada agente, especifique la cuenta con la que se ejecutará y la configuración de conexión.

Agente de instantáneas: D12-HP\

Configuración de seguridad...

Ayuda < Atrás Siguiente > Finalizar >>| Cancelar

Figura 61 Continuación del proceso

Siguiente.

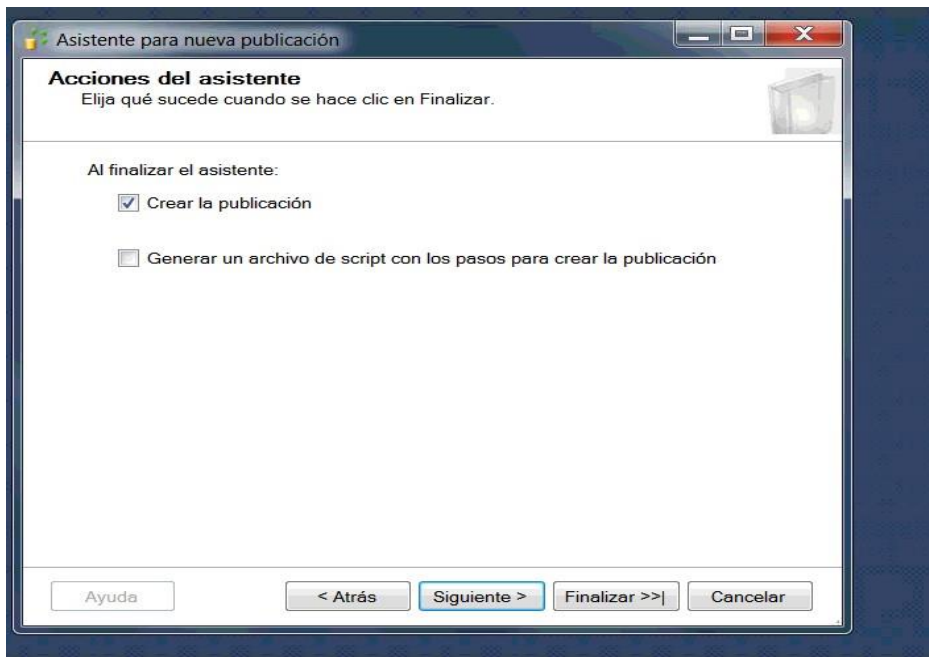


Figura 62 Crear publicación

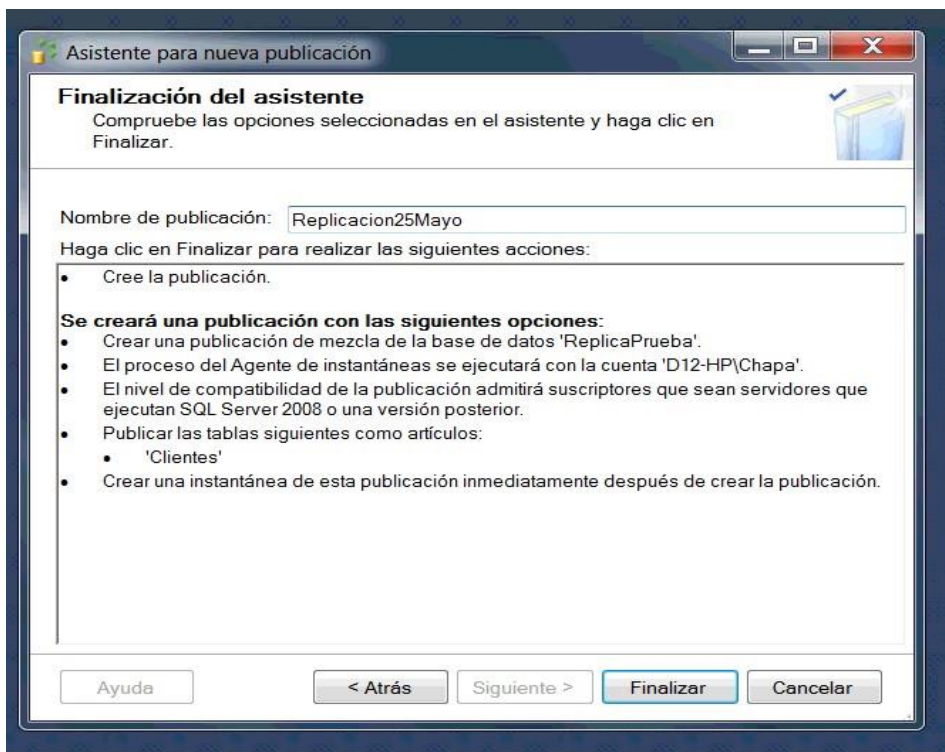


Figura 63 Finalización de publicación

Finalizamos la publicación.

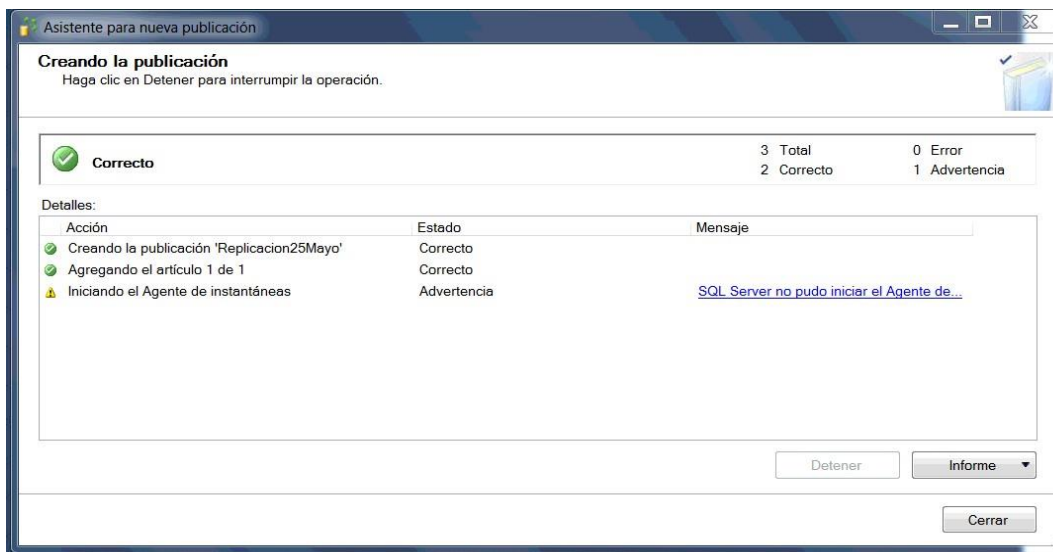


Figura 64 Pantalla de confirmación

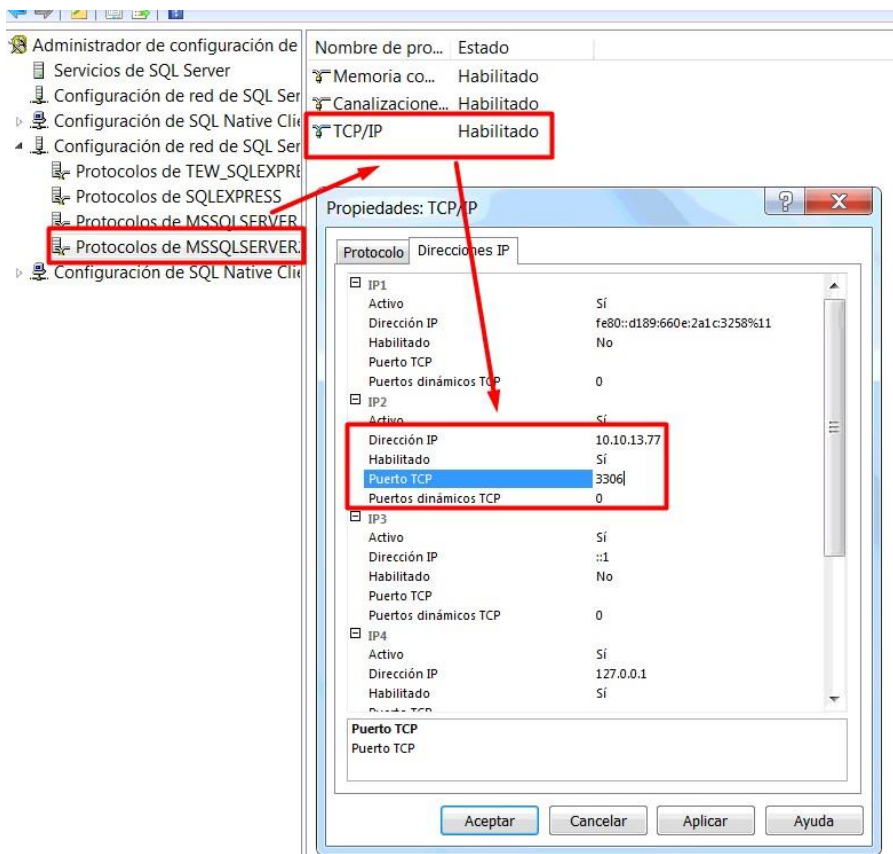


Figura 65 Verificación de configuración ambas máquinas

Asegurar que en ambas máquinas la configuración de ambos motores en sus propiedades de red tenga el mismo puerto tcp.

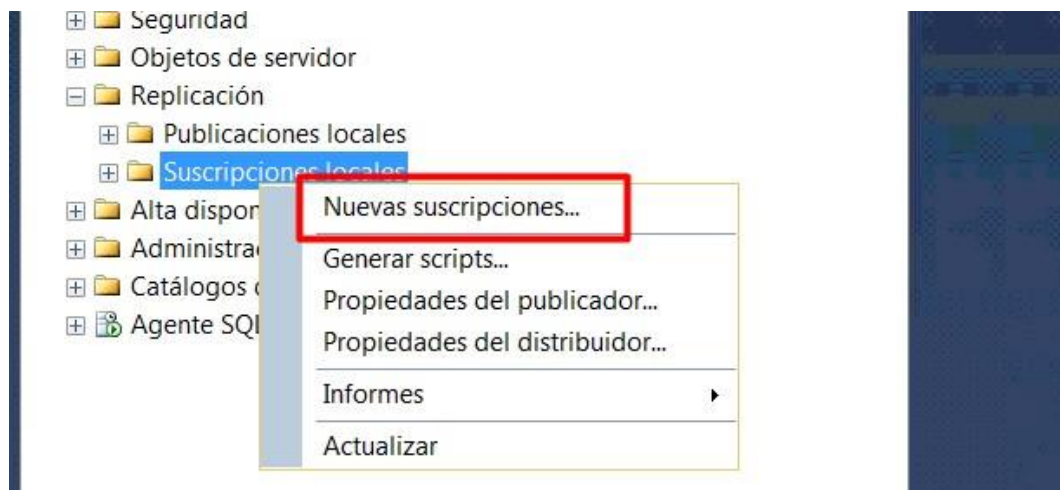


Figura 66 Nueva suscripción

Desde máquina 2 (D61), crearemos una nueva suscripción.

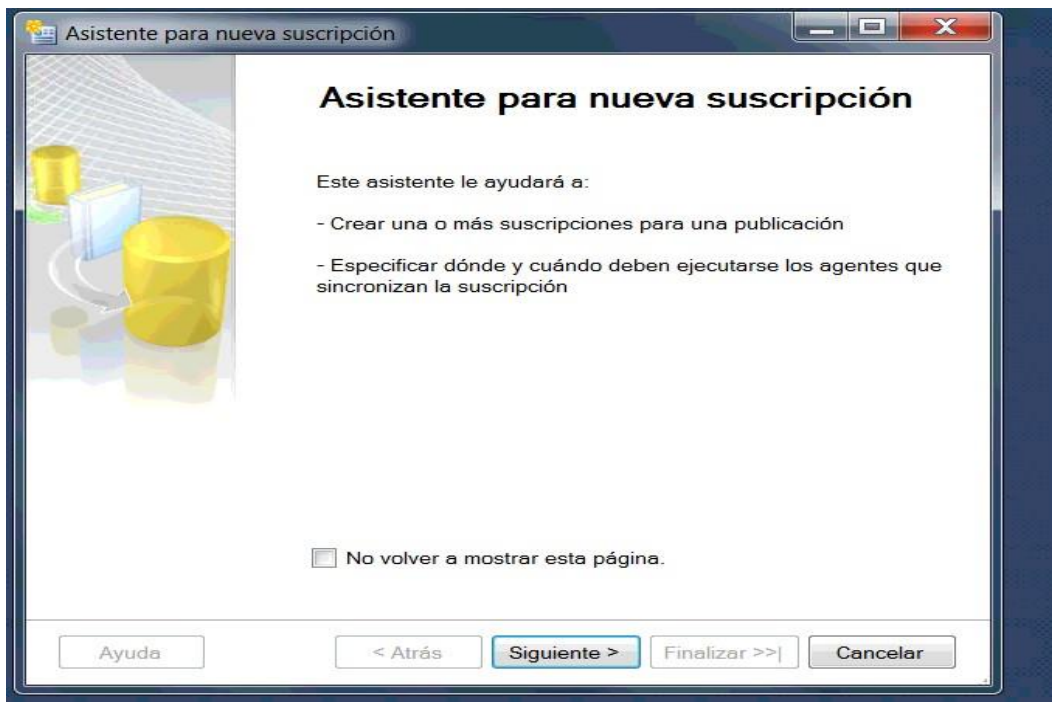


Figura 67 Asistente

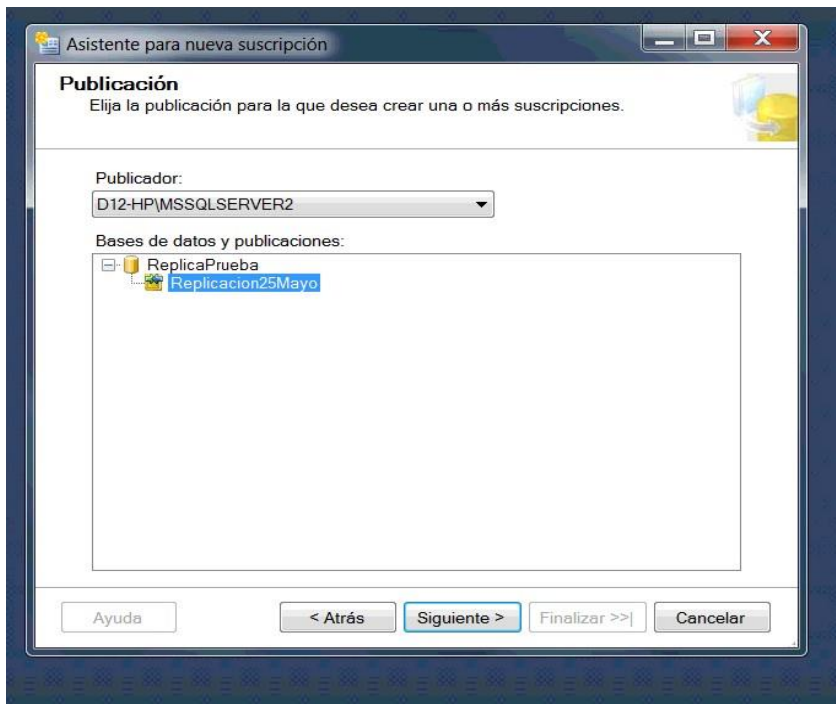


Figura 68 Máquina principal y selección de publicación anterior

Nos conectamos a la máquina principal y seleccionamos la publicación que creamos anteriormente.

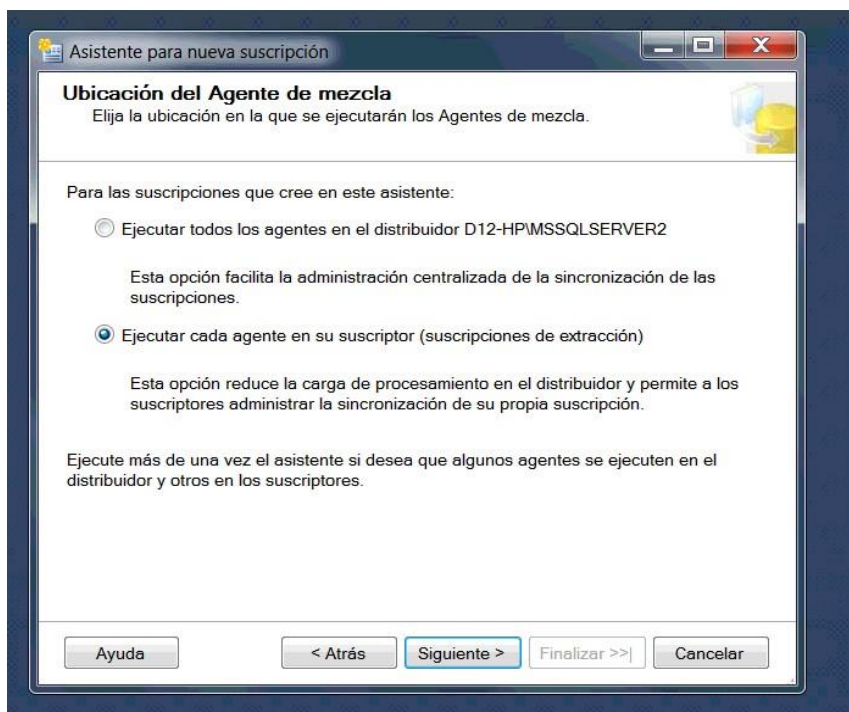


Figura 69 Ejecutar cada agente en su suscriptor

Seleccionamos la segunda opción.

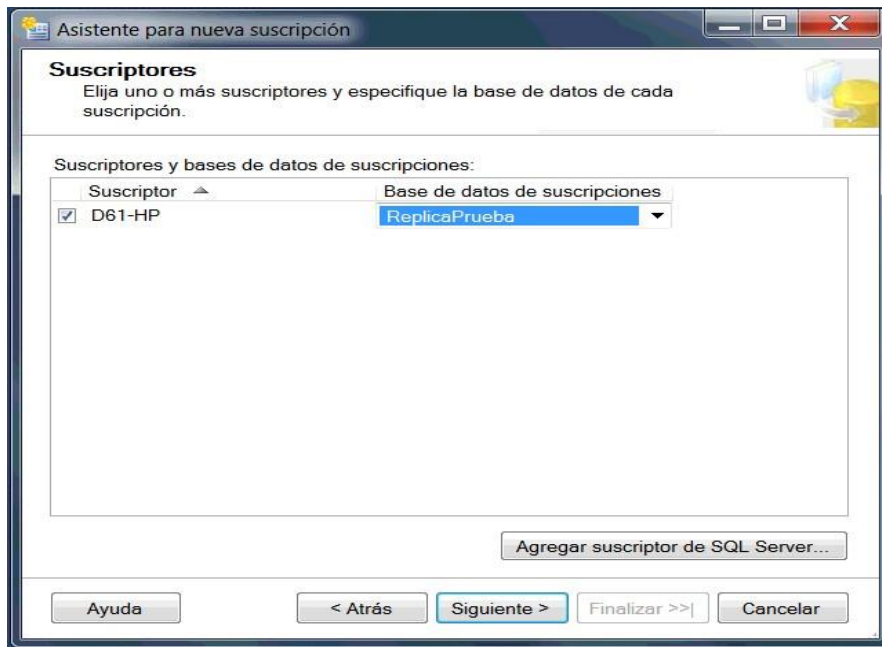


Figura 70 Backup anterior

En ese paso podemos crear una nueva base de datos, o seleccionar una que contenga una backup del original.

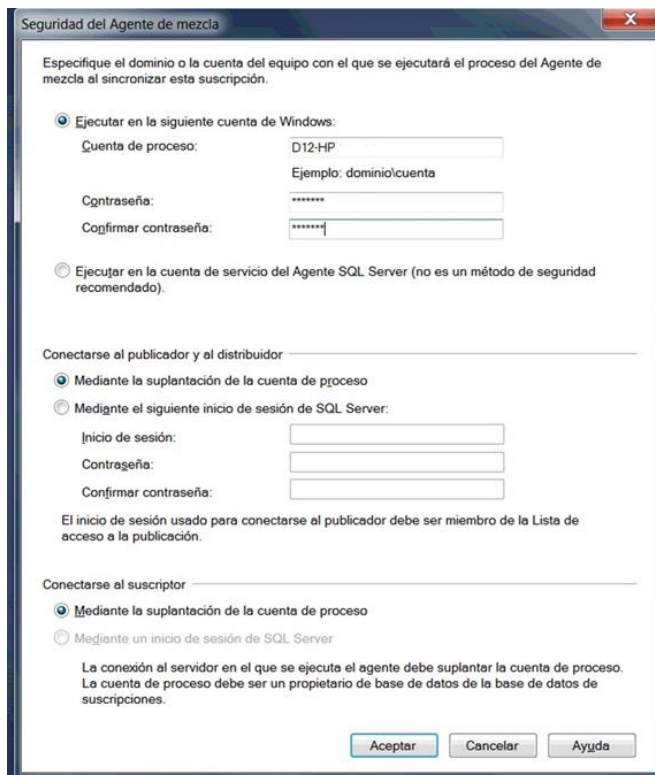


Figura 71 Windows distribuidor

Colocamos la cuenta de Windows del distribuidor.

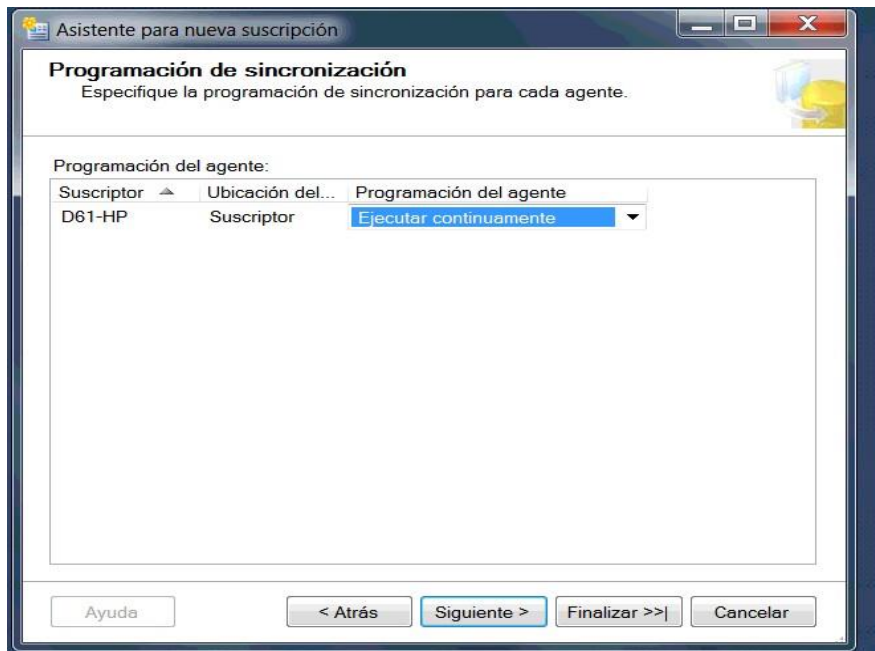


Figura 72 Ejecutar continuamente

Seleccionamos una programación del agente como 'Ejecutar continuamente.'

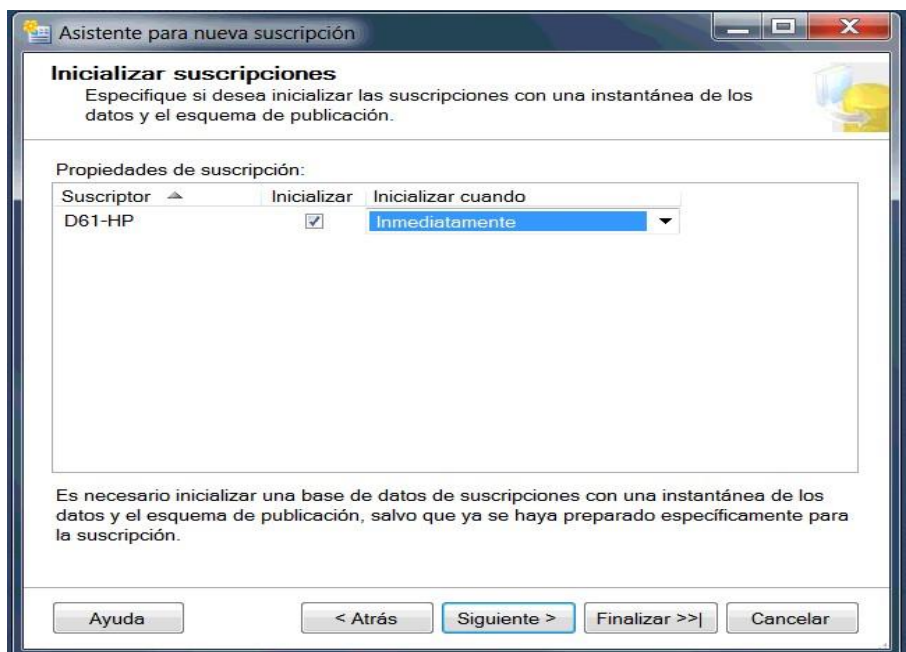


Figura 73 Inicialización de suscripción

La suscripción que se inicialice inmediatamente.

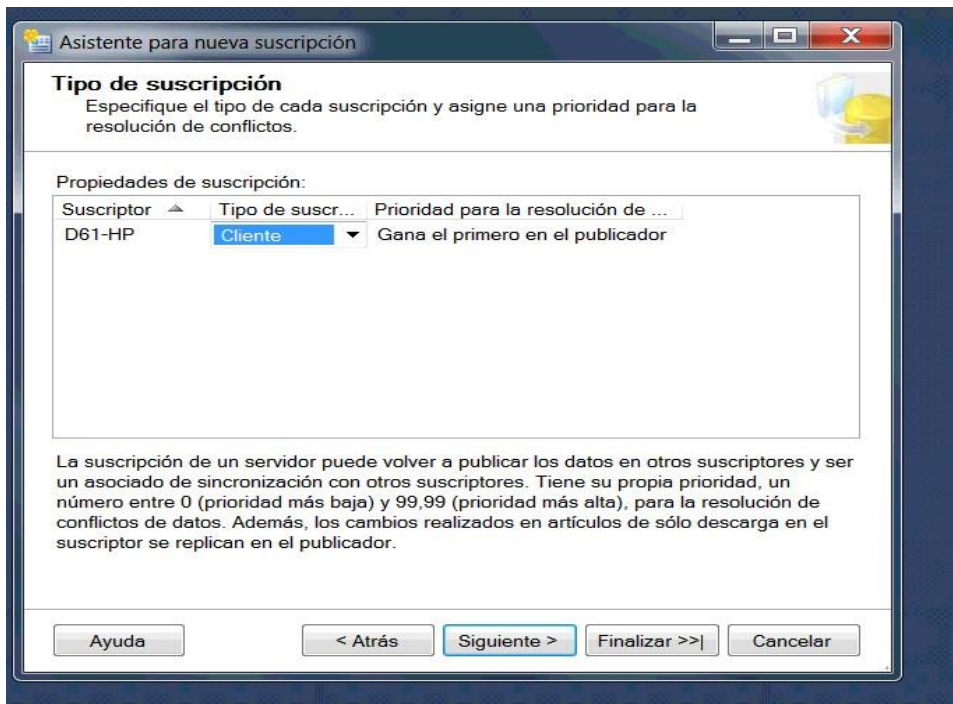


Figura 74 Tipo cliente

El tipo que sea de cliente.

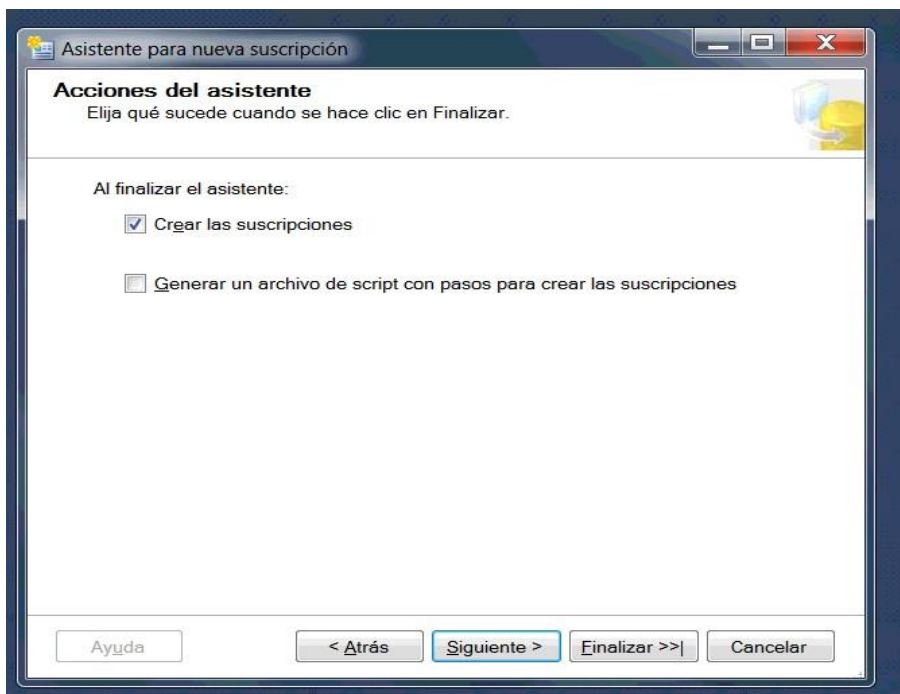


Figura 75 Crear suscripción

Creamos la suscripción.

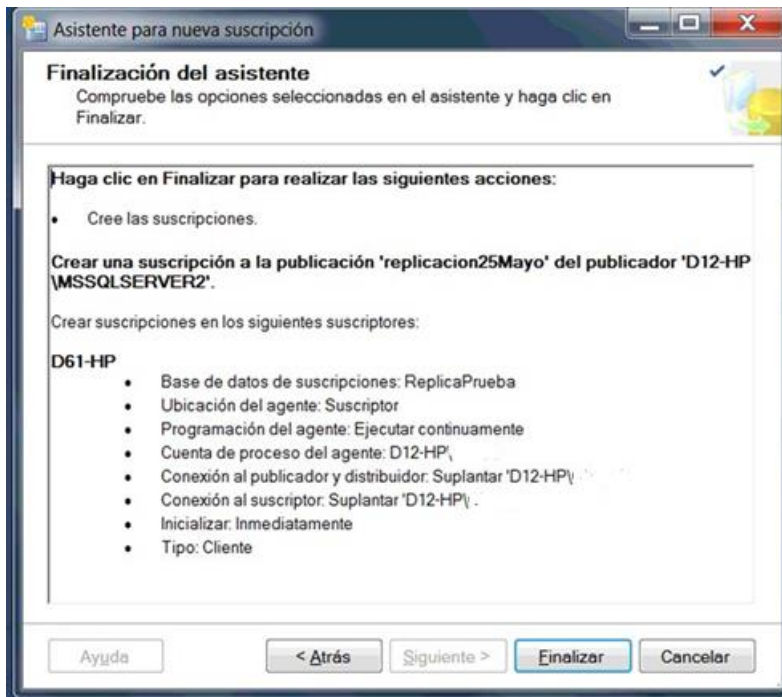


Figura 76 Finalización de suscripción

Finalizamos la suscripción.

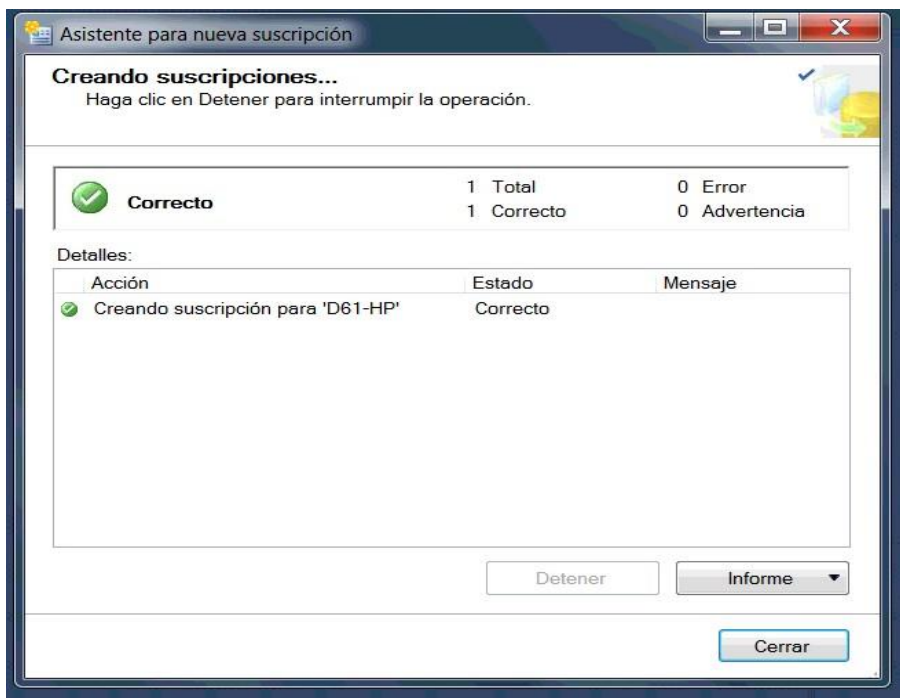


Figura 77 Verificación de creación

La suscripción ha sido un éxito. Ahora cada que ocurra una transacción en la base de datos (ya sea en la principal o en la secundaria), el cambio se aplicará en ambas

bases de datos. El tiempo depende del que se definió el momento de crear la publicación.

5.3 Métodos de respaldo de un SGBD.

en SQL Server, cada uno con un propósito específico para la recuperación de la base de datos. Los principales tipos de respaldo son:

1. Respaldo completo (Full Backup)

El respaldo completo es el método más fundamental. Crea una copia de seguridad de toda la base de datos, incluyendo todos los datos, los objetos (tablas, vistas, etc.) y una parte del registro de transacciones para garantizar la consistencia.

- **Uso:** Es la base para todos los demás tipos de respaldo. Se recomienda realizar respaldos completos regularmente para tener un punto de recuperación estable.
- **Ventaja:** Es el más fácil de restaurar. Con solo el respaldo completo, puedes recuperar la base de datos a un punto en el tiempo.
- **Desventaja:** Consume más tiempo y espacio en disco, ya que copia la base de datos completa cada vez.

2. Respaldo diferencial (Differential Backup)

Un respaldo diferencial solo copia los datos que han cambiado desde el último respaldo **completo**.

- **Uso:** Se usa para reducir el tamaño de las copias de seguridad y el tiempo de respaldo, especialmente en bases de datos grandes con muchos cambios.
- **Ventaja:** Es más rápido y pequeño que un respaldo completo.
- **Desventaja:** Para restaurar la base de datos, necesitas el último respaldo completo más el último respaldo diferencial. Si el respaldo completo se corrompe, no puedes restaurar la base de datos.

3. Respaldo del registro de transacciones (Transaction Log Backup)

Un respaldo del registro de transacciones copia todos los cambios registrados desde el último respaldo del registro de transacciones. Esto solo es posible si la base de datos está en el **modo de recuperación completa o de registro masivo**.

- **Uso:** Permite la recuperación a un punto específico en el tiempo, lo que se conoce como **recuperación puntual (Point-in-Time Recovery)**. Es crucial

para bases de datos con alta actividad y que no pueden permitirse la pérdida de datos.

- **Ventaja:** Son muy rápidos y pequeños, lo que permite realizarlos con alta frecuencia (cada 15 o 30 minutos).
- **Desventaja:** No puedes restaurar solo un respaldo del registro de transacciones. Debes restaurar el último respaldo completo, seguido del último respaldo diferencial (si lo hay), y luego la secuencia de respaldos del registro de transacciones.

4. Respaldo solo para copia (Copy-Only Backup)

Este tipo de respaldo no afecta la secuencia de copias de seguridad. Es decir, no interrumpe el ciclo de respaldos diferenciales o del registro de transacciones.

- **Uso:** Es útil cuando necesitas una copia de seguridad para un propósito específico (por ejemplo, para realizar pruebas o enviar a otro servidor) sin afectar los trabajos de respaldo regulares.

Resumen de la estrategia de respaldo

Una estrategia de respaldo común y robusta combina estos métodos:

1. Un **respaldo completo** semanal.
2. **Respaldos diferenciales** diarios.
3. **Respaldos del registro de transacciones** cada 15-30 minutos.

Esta combinación ofrece un equilibrio óptimo entre tiempo de respaldo, espacio en disco y la capacidad de recuperar la base de datos con una pérdida de datos mínima.

Ejemplo:

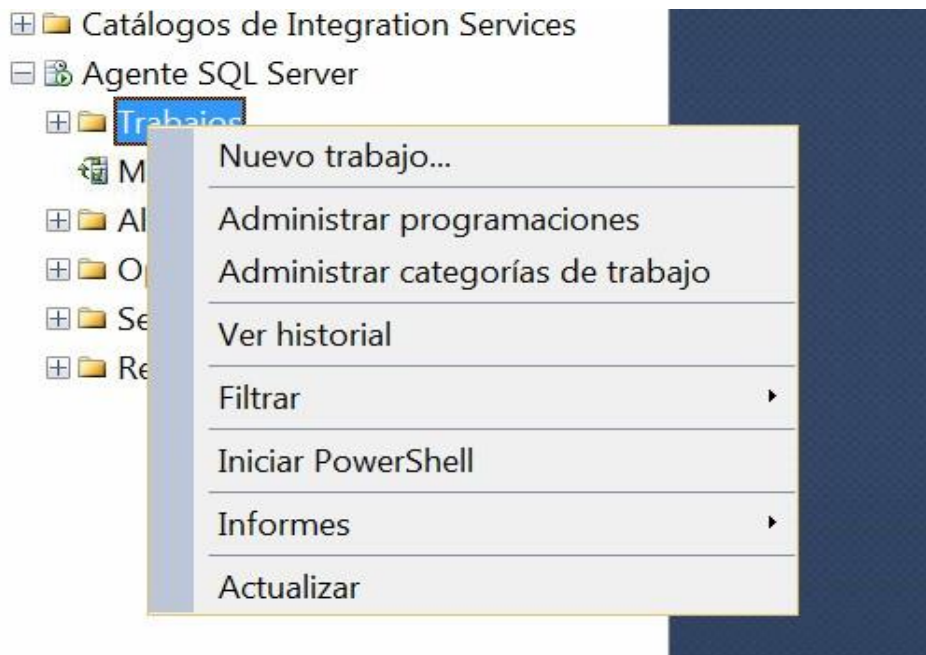


Figura 78 Nuevo trabajo

En el agente de SQL Server en la dirección de trabajos hacemos un nuevo trabajo.

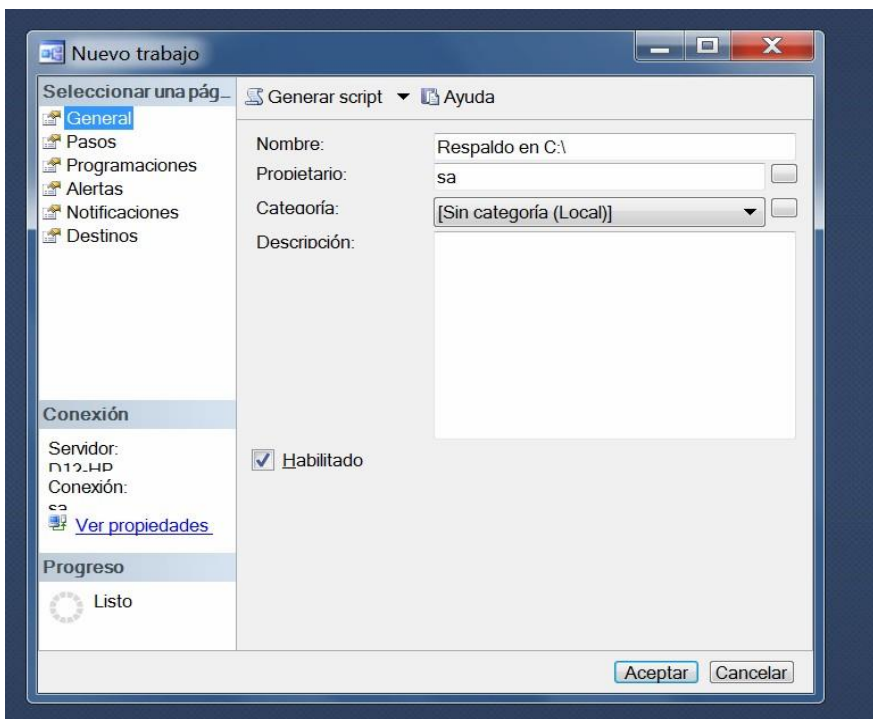


Figura 79 Nombre de trabajo

Agregamos un nombre al trabajo.

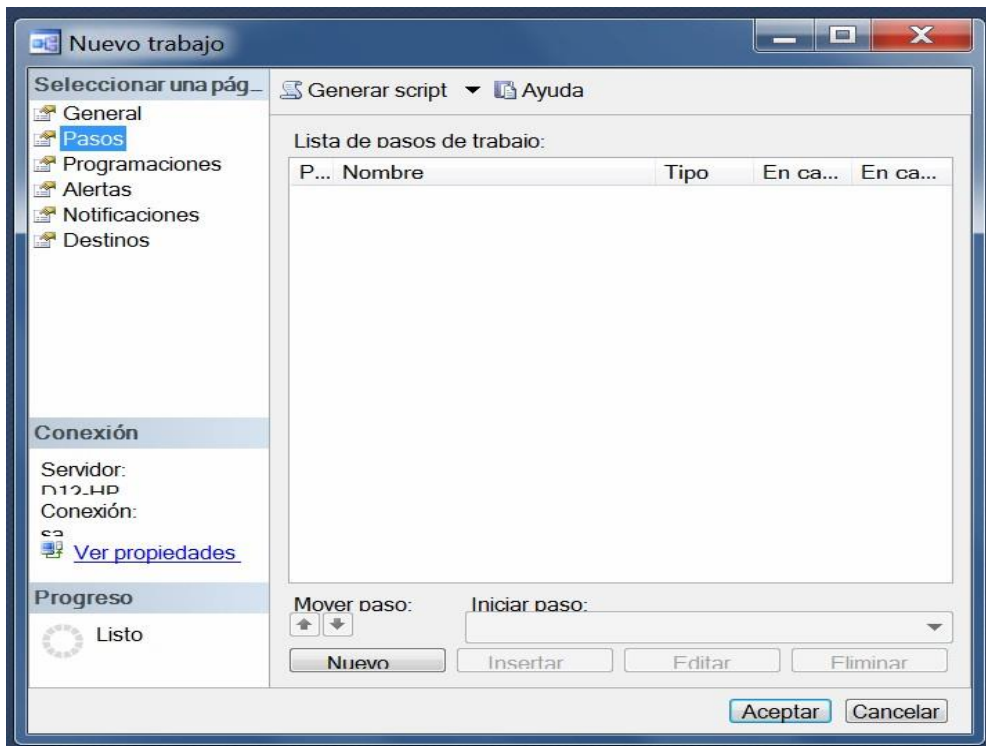


Figura 80 Comando de respaldo

En la sección de pasos agregaremos un paso que será el comando de respaldo.

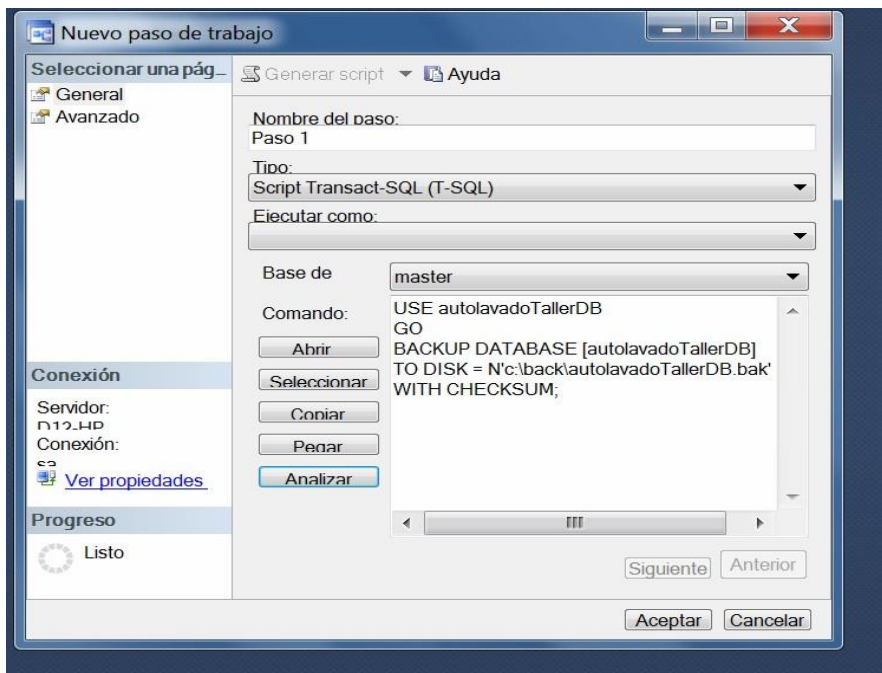


Figura 81 Nombre paso 1

Como nombre 'Paso 1' y dentro del campo de texto comando ingresamos el query para realizar un backup de una base de datos. Aceptamos.

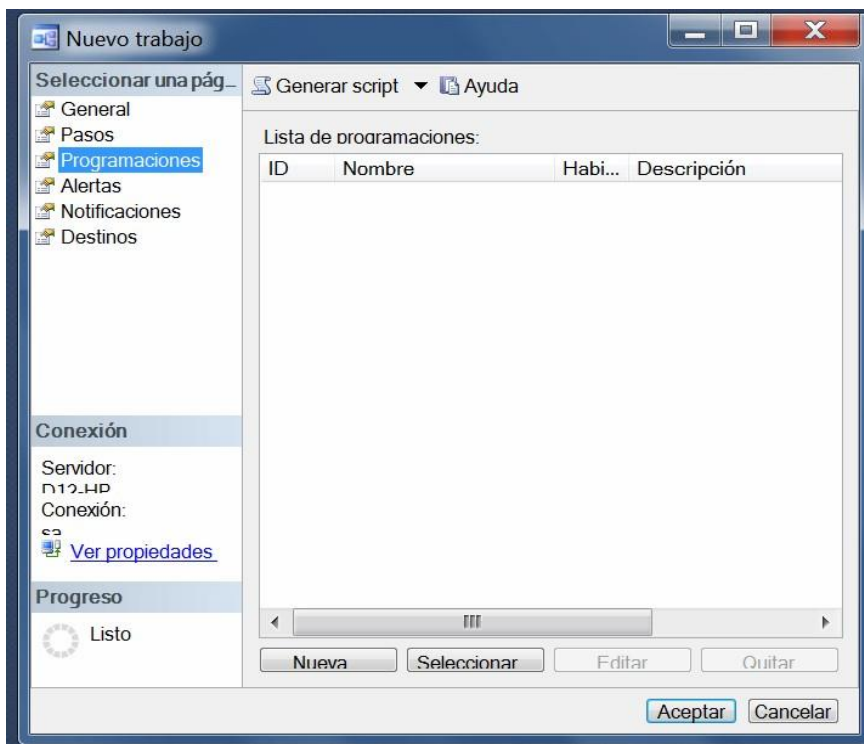


Figura 82 Nueva programación

En la sección Programaciones agregaremos una nueva programación.

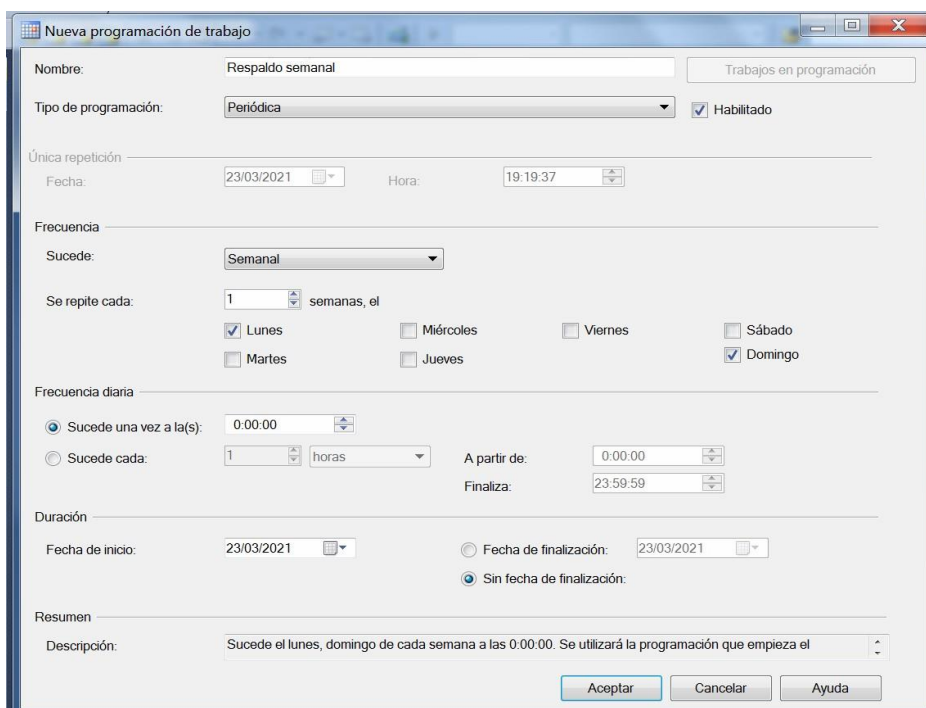


Figura 83 Elección de día

La programación es como desees, en este caso será cada lunes la programación. Y eso es todo lo mínimo para programar un respaldo automático.

Respaldo con un plan de mantenimiento.

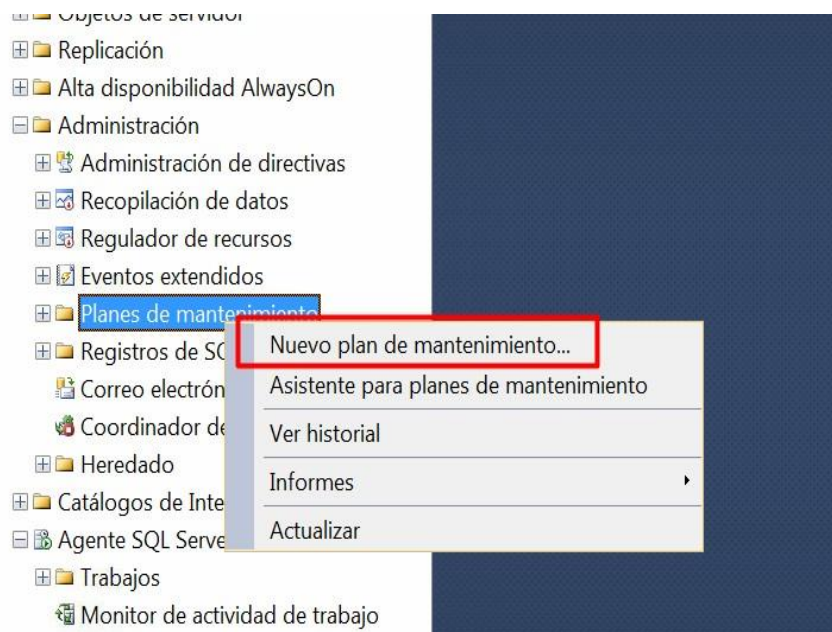


Figura 84 Plan de mantenimiento

En Administración están los planes de mantenimiento, agregamos un nuevo plan.

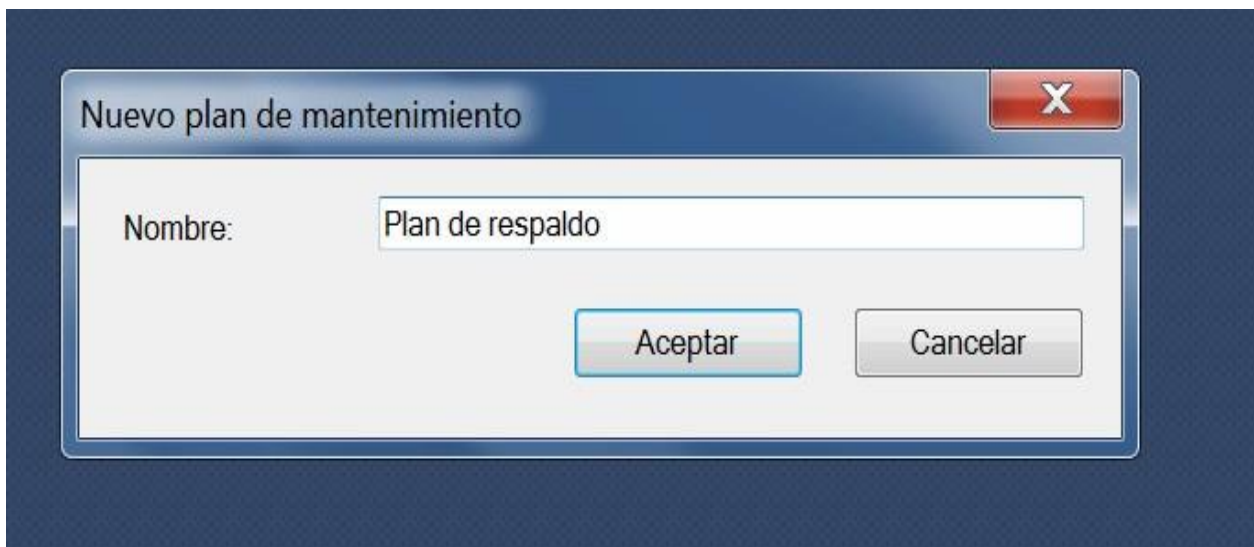


Figura 85 Nombre de plan de mantenimiento

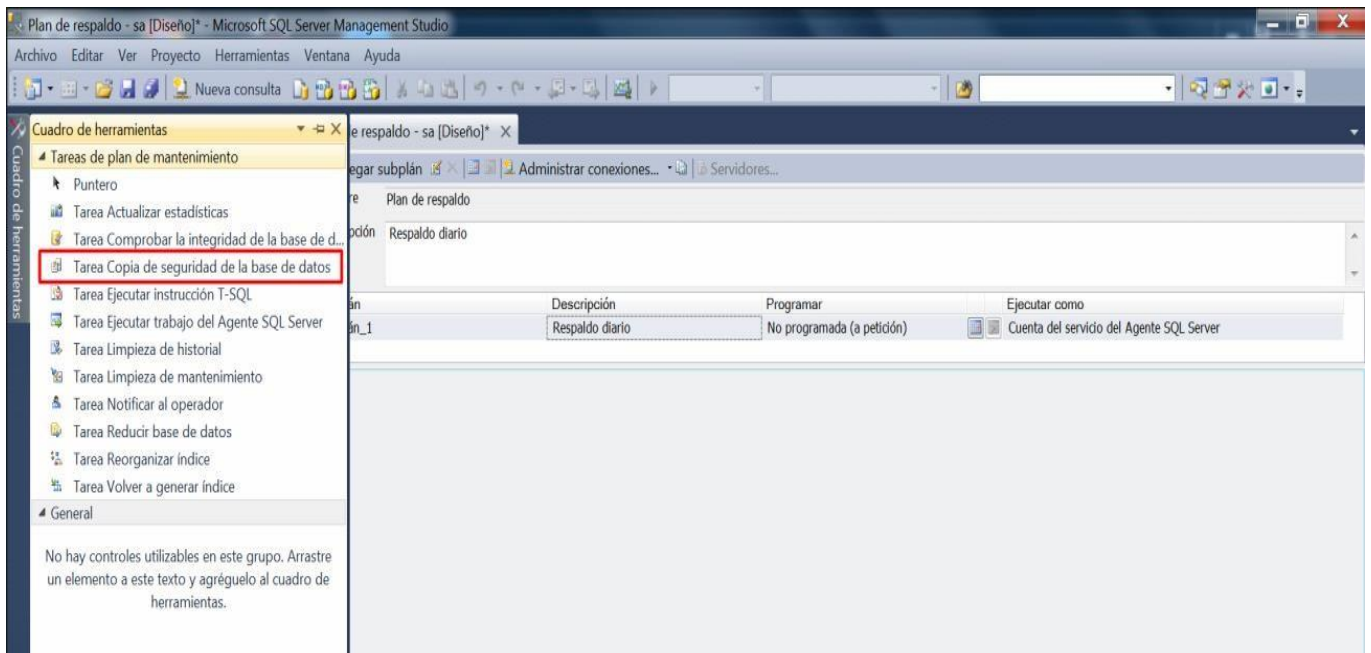


Figura 86 Tarea Copia de seguridad

Aquí abrimos la barra de tareas y agregamos una Tarea Copia de seguridad de la base de datos.

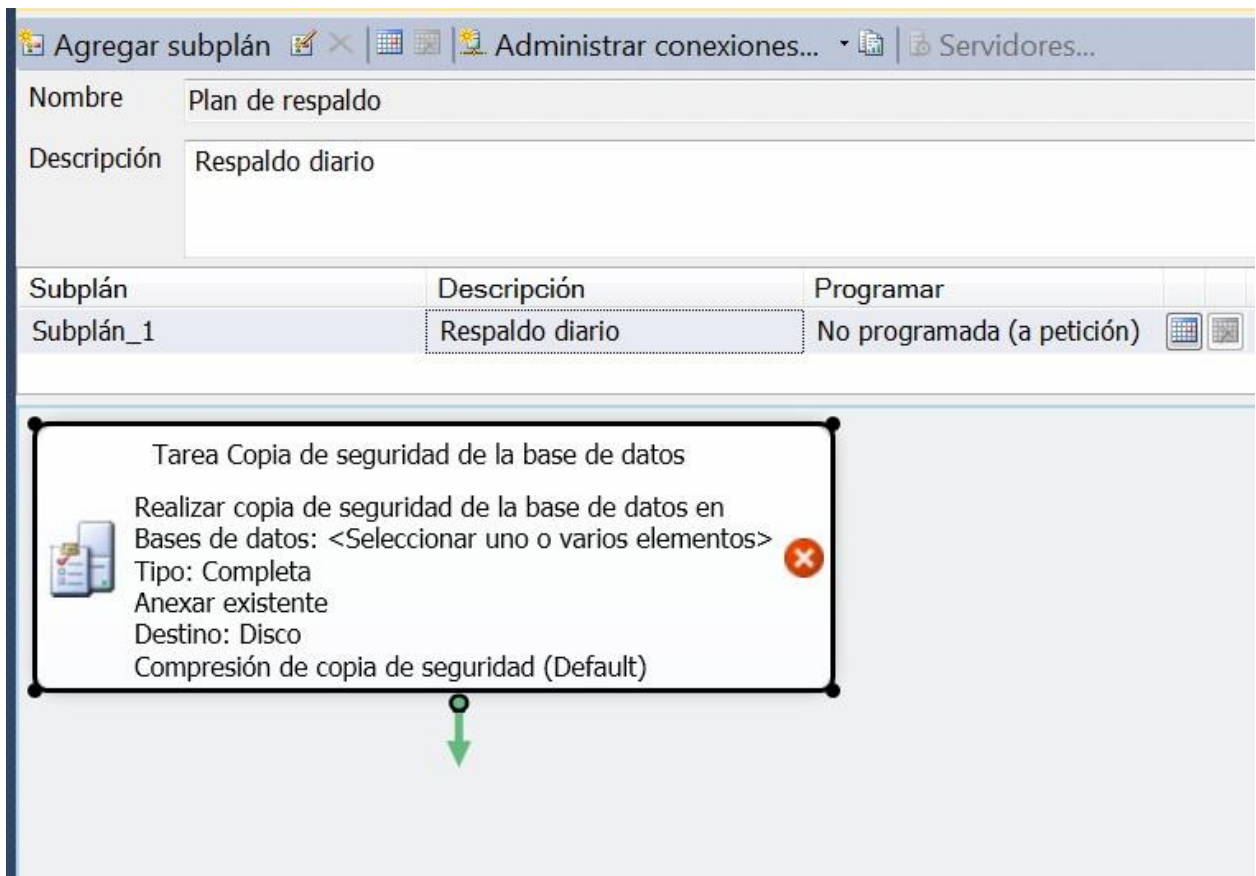
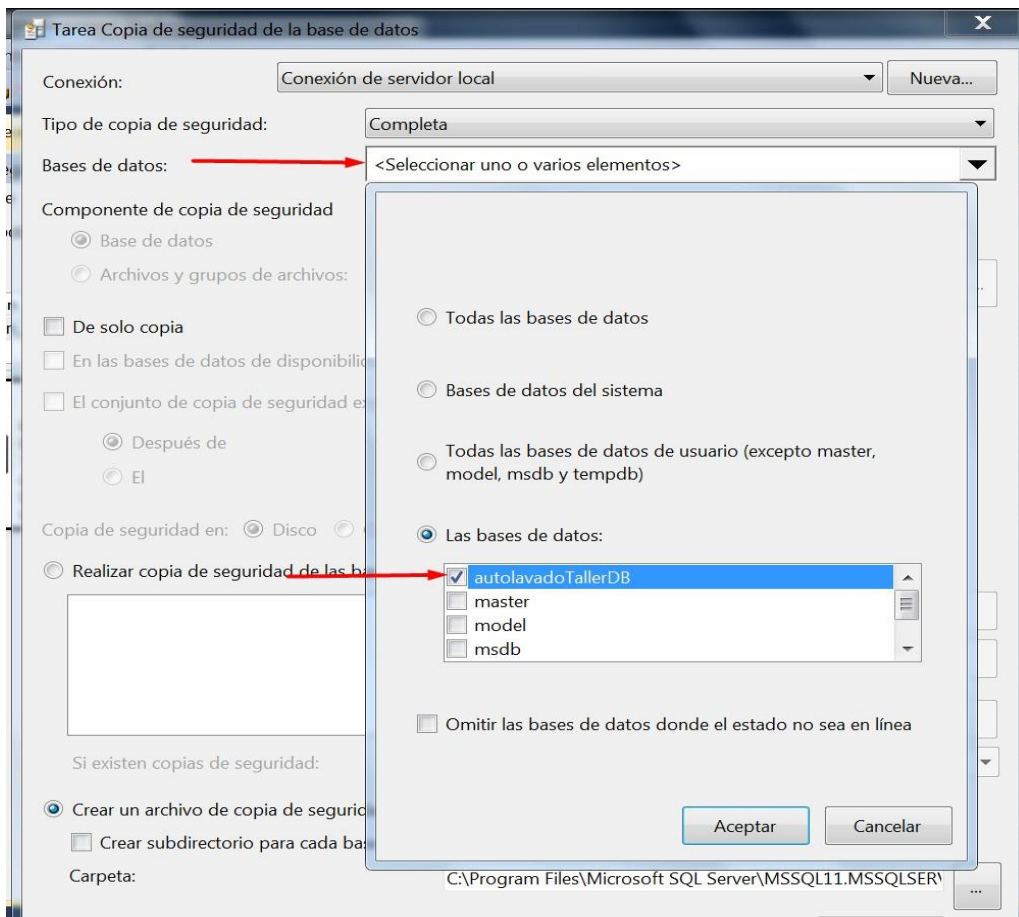
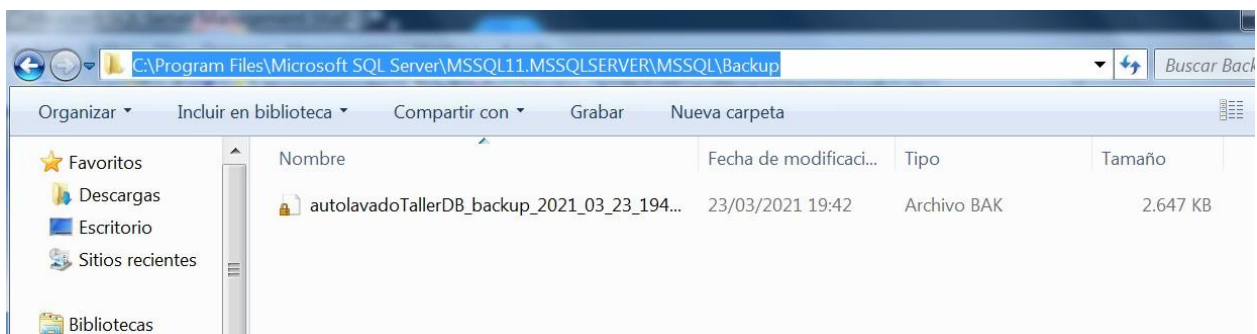


Figura 87 Doble clic

Damos doble click en el recuadro que se agregó.



Aquí seleccionamos la base de datos y también la carpeta donde se guardará el archivo .bak



5.4 Métodos de recuperación de un SGBD.

SQL Server ofrece varios **métodos de recuperación** que se utilizan para restaurar una base de datos a un estado consistente después de un fallo. La recuperación se

basa en la combinación de copias de seguridad (backups) y el registro de transacciones.

1. Recuperación completa (Full Recovery)

Este es el método más común y seguro. Permite recuperar la base de datos a cualquier punto en el tiempo, con una pérdida mínima de datos. Se utiliza en bases de datos críticas que no pueden permitirse la pérdida de información.

- **Requisitos:** Se debe utilizar el **modelo de recuperación completa** en la base de datos.
- **Proceso:**
 1. Restaurar el último **respaldo completo**.
 2. Restaurar el último **respaldo diferencial** (opcional, si existe).
 3. Restaurar la secuencia de **respaldos del registro de transacciones** en el orden en que se crearon, hasta el punto de fallo.
- **Ventaja:** Permite la **recuperación puntual (point-in-time recovery)**.

2. Recuperación a un punto de fallo

Este método se aplica cuando la base de datos se corrompe o el servidor cae. El objetivo es llevar la base de datos a un estado consistente, justo antes del fallo.

- **Proceso:**
 1. Restaurar el último **respaldo completo**.
 2. Restaurar el último **respaldo diferencial** (si lo hay).
 3. Restaurar la secuencia de **respaldos del registro de transacciones** hasta el momento exacto del fallo.
 4. SQL Server aplica las transacciones confirmadas y revierte las que estaban en curso, garantizando la consistencia de los datos.

3. Recuperación de la base de datos del sistema

Las bases de datos del sistema, como master, msdb y model, son críticas para el funcionamiento de SQL Server. Su recuperación es un proceso especial.

- **master:** Contiene toda la información de configuración del servidor y las bases de datos del usuario. Su recuperación debe hacerse en modo de usuario único para evitar que otros procesos se conecten.

- **msdb:** Utilizada por el Agente SQL Server. Si se corrompe, se puede perder la información de trabajos, alertas y operadores. Se restaura de forma similar a una base de datos de usuario.
- **model:** Es una plantilla para las nuevas bases de datos. Si se corrompe, no es tan crítica como las otras. Se puede restaurar a partir de una copia de seguridad o reemplazarla con una del mismo servidor.

4. Recuperación en caso de desastre

Cuando hay una pérdida completa del servidor, el proceso es más complejo e implica reconstruir la instancia de SQL Server antes de restaurar las bases de datos.

- **Proceso:**
 1. Instalar una nueva instancia de SQL Server en un servidor de reemplazo.
 2. Restaurar el último respaldo de la base de datos master para recuperar la configuración del servidor.
 3. Restaurar los respaldos de las bases de datos de usuario.
 4. Restaurar los respaldos de msdb y otras bases de datos del sistema.

5.5 Migración de la Base de Datos en sql server

La **migración de bases de datos** en SQL Server es el proceso de mover una base de datos de un servidor a otro. Esto puede ser entre diferentes versiones de SQL Server, de un servidor físico a uno virtual, o de un entorno local a la nube. El proceso garantiza que la base de datos, sus objetos y sus datos permanezcan intactos y funcionales en el nuevo entorno.

1. Métodos de migración

- **Backup y Restore:** Es el método más común y sencillo. Consiste en crear una copia de seguridad (.bak) de la base de datos en el servidor de origen y luego restaurarla en el servidor de destino. Es ideal para migraciones de bajo riesgo y cuando no se requiere una alta disponibilidad durante el proceso.
- **Detención y Conexión (Detach/Attach):** Se detienen los archivos de la base de datos (.mdf y .ldf) del servidor de origen, se copian al nuevo servidor y luego se adjuntan a la nueva instancia de SQL Server. Es un método rápido para bases de datos pequeñas, pero requiere tiempo de inactividad, ya que la base de datos no está disponible durante el proceso de copia.

- **Replicación Transaccional:** Permite mantener una base de datos de destino actualizada con los cambios del origen en tiempo casi real. Es una buena opción para migraciones de bases de datos grandes, ya que el tiempo de inactividad se reduce al mínimo.
- **SQL Server Management Studio (SSMS):** El asistente de importación y exportación de SSMS permite migrar bases de datos y objetos entre servidores. También se pueden usar los asistentes de planes de mantenimiento para generar scripts de migración.
- **Data Migration Assistant (DMA):** Es una herramienta gratuita de Microsoft que ayuda a evaluar y migrar bases de datos. DMA puede detectar problemas de compatibilidad y dar recomendaciones antes de la migración. Es muy útil para migrar a nuevas versiones de SQL Server o a Azure SQL.

2. Proceso de migración

El proceso de migración de una base de datos generalmente consta de las siguientes etapas:

1. **Evaluación:** Analizar la base de datos de origen para identificar posibles problemas de compatibilidad con la versión de destino. Se pueden usar herramientas como DMA para este fin.
2. **Preparación:** Realizar una copia de seguridad completa del origen, detener la base de datos para nuevas transacciones y asegurar que el servidor de destino tenga suficiente espacio en disco.
3. **Migración de la base de datos:** Usar uno de los métodos mencionados para transferir la base de datos.
4. **Verificación:** Probar la base de datos migrada para asegurar que funcione correctamente. Se puede hacer con herramientas de comparación de datos y esquemas, o simplemente ejecutando pruebas de la aplicación.
5. **Actualización de aplicaciones:** Configurar las aplicaciones para que se conecten al nuevo servidor de bases de datos.

CUESTIONARIO TEMA 5

1. **¿Cuál es el principal propósito del espejeo (mirroring) en SQL Server?**
 - a) Mejorar el rendimiento de consultas complejas
 - b) Mantener una copia exacta y en tiempo real de una base de datos para alta disponibilidad

- c) Replicar tablas específicas entre bases de datos
- 2. ¿Cuál es la función principal del servidor testigo en un entorno de espejo?**
- a) Almacenar los datos replicados para los suscriptores
 - b) Monitorizar la conexión entre el servidor principal y espejo y permitir failover automático
 - c) Ejecutar consultas de solo lectura en la base de datos espejo
- 3. En el modo de alta seguridad (High-Safety Mode) del espejo, ¿qué característica lo define?**
- a) Confirmación de transacciones en el servidor principal antes de que el servidor espejo confirme
 - b) Confirmación asíncrona de transacciones para mejorar el rendimiento
 - c) Falta de failover automático incluso con testigo
- 4. ¿Cuál es una desventaja importante del espejo en SQL Server?**
- a) Puede replicar solo tablas específicas, no bases completas
 - b) Solo permite una base de datos especificada por instancia de SQL Server
 - c) No soporta failover automático bajo ningún modo
- 5. ¿Cuál es la diferencia principal entre réplica (replicación) y espejo (duplicación) en SQL Server?**
- a) El espejo funciona a nivel de base de datos completa, la réplica permite replicar objetos específicos
 - b) La réplica solo copia bases de datos completas, el espejo sincroniza tablas individuales
 - c) La réplica no puede funcionar en tiempo real, mientras que el espejo sí
- 6. ¿Cuál de los siguientes no es un componente clave de la réplica en SQL Server?**
- a) Publicador (Editor)
 - b) Distribuidor (Distribuidor)
 - c) Testigo (Testigo)
- 7. ¿Qué tipo de respaldo en SQL Server copia únicamente los datos modificados desde el último respaldo completo?**
- a) Respaldo completo (Full Backup)

- b) Respaldo diferencial (Copia de seguridad diferencial)
 - c) Respaldo del registro de transacciones (Transaction Log Backup)
- 8. ¿Cuál es el principal beneficio del respaldo del registro de transacciones?**
- a) Copiar toda la base de datos para una recuperación rápida
 - b) Permitir la recuperación puntual (point-in-time) con respaldos pequeños y frecuentes
 - c) No afecta la secuencia de respaldos diferenciales
- 9. ¿Cuál es el orden correcto para restaurar una base de datos en una recuperación completa?**
- a) Último respaldo diferencial, último respaldo completo, respaldos del registro de transacciones
 - b) Último respaldo completo, último respaldo diferencial (si existe), respaldos del registro de transacciones.
 - c) Solo el último respaldo completo es necesario
- 10. ¿Cuál es la herramienta recomendada para evaluar problemas de compatibilidad antes de migrar una base de datos a SQL Server?**
- a) SQL Server Management Studio (SSMS)
 - b) Asistente de migración de datos (DMA)
 - c) Replicación Transaccional

Respuestas correctas:

1. b) Mantener una copia exacta y en tiempo real de una base de datos para alta disponibilidad
2. b) Monitorizar la conexión entre el servidor principal y espejo y permitir la conmutación por error automática
3. a) Confirmación de transacciones en el servidor principal antes de que el servidor espejo confirme
4. b) Solo permite una base de datos especificada por instancia de SQL Server
5. a) El espejeo funciona a nivel de base de datos completa, la réplica permite replicar objetos específicos
6. c) Testigo

7. b) Respaldo diferencial
8. b) Permitir la recuperación puntual (point-in-time) con respaldos pequeños y frecuentes
9. b) Último respaldo completo, último respaldo diferencial (si existe), respaldos del registro de transacciones
10. b) Asistente de migración de datos (DMA)

Tema 6 Monitoreo y auditoría

Implementa la auditoría de base de datos para controlar la seguridad de la información. Implementa el monitoreo del rendimiento de un SGBD para verificar su funcionamiento.

6.1 Monitoreo de la Base de Datos

El monitoreo es el proceso de supervisar el rendimiento y el estado de la base de datos en tiempo real. Su objetivo es detectar problemas de manera proactiva antes de que afecten a los usuarios finales o causen una caída del sistema. El monitoreo es una actividad continua y esencial para la salud de cualquier sistema de bases de datos.

- **¿Qué se monitorea?**
 - **Uso de recursos del sistema:** Se vigila la carga de trabajo en el servidor, como el uso del CPU, la memoria (RAM) y el espacio en disco. Un alto consumo de CPU puede indicar que hay consultas ineficientes, mientras que la falta de espacio en disco es una causa común de fallos.
 - **Estadísticas de la base de datos:** Se observan métricas internas del SGBD, como el número de conexiones activas, el tiempo de ejecución de las consultas, la tasa de aciertos del caché de datos y la tasa de bloqueos (*deadlocks*).
 - **Rendimiento de las consultas:** El monitoreo permite identificar las consultas que están tardando más tiempo en ejecutarse. El DBA puede usar esta información para optimizar las consultas, crear o modificar índices, y mejorar el diseño de la base de datos.
- **Herramientas de monitoreo:**
 - **Herramientas nativas del SGBD:** La mayoría de los SGBD (como Oracle, SQL Server, MySQL, PostgreSQL) incluyen herramientas de monitoreo integradas. Por ejemplo, en SQL Server se usa el *SQL Server Profiler* o los *Performance Monitor Counters*.
 - **Herramientas de terceros:** Hay una amplia gama de herramientas comerciales y de código abierto que ofrecen capacidades de monitoreo más avanzadas, con paneles de control visuales, alertas automatizadas y análisis históricos del rendimiento.

6.2 Auditoría de la Base de Datos

La auditoría, a diferencia del monitoreo, se centra en registrar las acciones de los usuarios para tener un registro histórico y poder analizarlas posteriormente. Su propósito principal es la seguridad, el cumplimiento de normativas y la trazabilidad de los datos.

- **¿Qué se audita?**
 - **Accesos a la base de datos:** Se auditan los intentos de inicio de sesión, tanto exitosos como fallidos. Esto es crucial para detectar intentos de fuerza bruta o accesos no autorizados.
 - **Cambios en los datos:** Se registran las operaciones de modificación de datos (INSERT, UPDATE, DELETE) en tablas críticas, lo que permite saber quién, cuándo y qué datos fueron alterados.
 - **Cambios de privilegios:** Es fundamental auditar los comandos GRANT y REVOKE, ya que los cambios en los privilegios de los usuarios pueden representar un riesgo de seguridad.
 - **Ejecución de sentencias DDL:** Se audita la creación, modificación o eliminación de objetos de la base de datos (tablas, vistas, procedimientos almacenados), lo que ayuda a rastrear cambios en la estructura de la base de datos.
- **Tipos de auditoría:**
 - **Auditoría a nivel de sentencia:** Registra cada sentencia SQL ejecutada por un usuario. Es muy detallada, pero puede generar un gran volumen de datos de auditoría.
 - **Auditoría a nivel de esquema o de objeto:** Se enfoca en la auditoría de acciones sobre objetos específicos (por ejemplo, auditar SELECT en la tabla de salarios) o en cambios en la estructura del esquema.
- **Mecanismos de auditoría:**
 - **Características del SGBD:** La mayoría de los SGBD modernos tienen su propio sistema de auditoría. Por ejemplo, Oracle tiene el Oracle Audit Vault and Database Firewall y SQL Server utiliza las especificaciones de auditoría del servidor.
 - **Triggers:** Se pueden usar *triggers* (disparadores) para capturar cambios en los datos. Por ejemplo, un *trigger* en una tabla puede registrar cada vez que se inserta un nuevo registro y quién lo hizo. Sin embargo, su uso excesivo puede afectar el rendimiento.

Relación entre Monitoreo y Auditoría

Aunque son conceptos distintos, el monitoreo y la auditoría se complementan para proporcionar una visión completa del estado de la base de datos.

- **Monitoreo:** Se enfoca en el **rendimiento** y la **disponibilidad**. Responde a la pregunta: *¿El sistema está funcionando bien ahora?*
- **Auditoría:** Se enfoca en la **seguridad** y la **trazabilidad**. Responde a la pregunta: *¿Qué ha pasado en el sistema?*

Para cualquier administrador de bases de datos, tanto el monitoreo como la auditoría son tareas que no se pueden dejar de lado. El monitoreo te ayuda a reaccionar de inmediato cuando hay problemas de rendimiento, evitando dolores de cabeza más adelante. Por otro lado, la auditoría funciona como un archivo histórico donde puedes revisar quién hizo qué, lo cual es clave para detectar fraudes, investigar incidentes de seguridad o cumplir con las reglas y leyes. Un buen DBA debe saber manejar ambas áreas para que la información siempre esté segura, disponible y protegida.

CUESTIONARIO TEMA 6

1. **¿Cuál es el principal objetivo del monitoreo de una base de datos?**
 - a) Registrar todas las acciones de los usuarios para análisis futuros
 - b) Detectar problemas proactivamente antes de que afecten a los usuarios finales
 - c) Modificar el diseño de la base de datos automáticamente
2. **¿Cuál de las siguientes métricas NO es en gran medida monitoreada en una base de datos?**
 - a) Uso de CPU y memoria
 - b) Cambios en privilegios de usuarios
 - c) Tasa de bloqueos (deadlocks)
3. **¿Qué herramienta es nativa de SQL Server para el monitoreo?**
 - a) Oracle Audit Vault
 - b) SQL Server Profiler
 - c) Nagios
4. **¿Cuál es la diferencia principal entre monitoreo y auditoría en bases de datos?**
 - a) El monitoreo registro acciones de usuarios; la auditoría supervisa el rendimiento

- b) El monitoreo se enfoca en seguridad; la auditoría en disponibilidad
 - c) El monitoreo para supervisar el rendimiento en tiempo real; la auditoría registra acciones para análisis histórico
- 5. ¿Qué tipo de auditoría registra cada sentencia SQL ejecutada por un usuario?**
- a) Auditoría a nivel de esquema
 - b) Auditoría a nivel de sentencia
 - c) Auditoría por triggers
- 6. ¿Para qué se utilizan los triggers en la auditoría de bases de datos?**
- a) Para optimizar consultas y mejorar índices
 - b) Para capturar cambios en los datos y registrar quién realizó la acción
 - c) Para medir el uso de CPU y memoria
- 7. ¿Cuáles de las siguientes acciones se auditan para controlar cambios en la estructura de la base de datos?**
- a) Uso del espacio en disco
 - b) Ejecución de sentencias DDL (CREATE, ALTER, DROP)
 - c) Consultas lentas
- 8. ¿Cuál es un riesgo asociado al uso excesivo de triggers para auditoría?**
- a) Generar datos de auditoría insuficientes
 - b) Afectar negativamente el rendimiento del sistema
 - c) No registrar cambios en privilegios de usuarios
- 9. ¿Qué aspecto se vigila para identificar consultas ineficientes durante el monitoreo?**
- a) Número de conexiones activas
 - b) Tiempo de ejecución de las consultas
 - c) Intentos fallidos de inicio de sesión
- 10. ¿Por qué es fundamental que un DBA domine tanto el monitoreo como la auditoría?**
- a) Para garantizar que el sistema funcione sin necesidad de mantenimiento
 - b) Para asegurar integridad, disponibilidad y confidencialidad de los datos
 - c) Para eliminar la necesidad de herramientas externas

Respuestas correctas:

1. b) Detectar problemas proactivamente antes de que afecten a los usuarios finales
2. b) Cambios en privilegios de usuarios
3. b) Analizador de SQL Server
4. c) El monitoreo del rendimiento en tiempo real; la analisis registro acciones para analisis historico
5. b) Auditoría a nivel de sentencia
6. b) Para capturar cambios en los datos y registrar quién realizó la acción
7. b) Ejecución de sentencias DDL (CREATE, ALTER, DROP)
8. b) Afectar negativamente el rendimiento del sistema
9. b) Tiempo de ejecución de las consultas
10. b) Para asegurar la integridad, disponibilidad y confidencialidad de los datos

Bibliografía

1. Transición a una carrera profesional en bases de datos: Guía para principiantes,
<https://learnsql.es/blog/transicion-a-una-carrera-profesional-en-bases-de-datos-guia-para-principiantes/>
2. Perfil y Salario de un Perfil de Administrador de Bases de Datos en México en 2025,
<https://hireline.io/mx/enciclopedia-de-perfiles-de-tecnologia/administrador-de-bases-de-datos>
3. Funciones esenciales de un administrador de bases de datos - Tiffin University,
<https://global.tiffin.edu/blog/administrador-de-base-de-datos-que-es-y-cuales-son-sus-funciones>
4. Database Administrator Careers : Skills, Qualifications & Job Requirements - ComputerScience.org,
<https://www.computerscience.org/careers/database-administrator/>
5. ¿Qué es un DBA o Administrador de Bases de Datos? - KeepCoding,
<https://keepcoding.io/blog/dba-requisitos-habilidades-y-responsabilidades/>
6. Ofertas de trabajo de dba 2025 | Bolsa de trabajo OCC,
<https://www.occ.com.mx/empleos/de-dba/>
7. www.educaweb.com,
<https://www.educaweb.com/profesion/administrador-bases-datos-724/#:~:text=Perfil%20profesional%20de%20un%20Administrador%20de%20bases%20de%20datos&text=Tener%20capacidad%20de%20organizaci%C3%B3n.,de%20planificaci%C3%B3n%20y%20de%20previsi%C3%B3n.>
8. 10 habilidades blandas y técnicas que todos los científicos de datos deben tener - Tableau,
<https://www.tableau.com/es-es/learn/articles/data-science-skills>
9. ¿Qué habilidades necesitan los profesionales de gestión de datos? - Discover The New,

<https://discoverthenew.ituser.es/predictive-analytics/2022/06/que-habilidades-necesitan-los-profesionales-de-gestion-de-datos>

10. Perfil de ocupación Administradores de Bases de Datos CareerOneStop,
<https://www.careeronestop.org/toolkit/careers/occupations/occupation-profile.aspx?keyword=Database%20Administrators&onetcode=15124200&location=Texas&lang=es>
11. Problemas comunes de crecimiento de bases de datos y soluciones - OneNine,
<https://onenine.com/es/common-database-growth-issues-and-fixes/>
12. Concurrencia de bases de datos | Solix Technologies, Inc.,
<https://www.solix.com/es/kb/database-concurrency/>
13. Control de concurrencia - AppMaster,
<https://appmaster.io/es/glossary/control-de-concurrencia>
14. ¿Qué es la Alta disponibilidad de MySQL? | Pure Storage,
<https://www.purestorage.com/es/knowledge/what-is-mysql-high-availability.html>
15. Principales desafíos de la Seguridad de Datos y cómo abordarlos - Fortra,
<https://www.fortra.com/es/blog/top-6-data-security-challenges-and-how-address-them>
16. 5 Mejores Prácticas de Seguridad en Bases de Datos | DS - DataSunrise,
<https://www.datasunrise.com/es/informacion-profesional/5-mejores-practicas-seguridad-db/>
17. 10 prácticas recomendadas de seguridad de bases de datos para 2025 - OneNine,
<https://onenine.com/es/10-database-security-best-practices-2025/>
18. 10 Desafíos en seguridad de datos para organizaciones y CISOs - Sealpath,
<https://www.sealpath.com/es/blog/desafios-seguridad-datos-empresas/>
19. Explicación de los tipos de backups: Completa, Incremental y Diferencial - NAKIVO,
<https://www.nakivo.com/es/blog/backup-types-explained/>

20. ¿Qué es la copia de seguridad de datos? - AWS,
<https://aws.amazon.com/es/what-is/data-backup/>
21. Mejores prácticas de copia de seguridad y recuperación de datos para el sector industrial - Acronis,
<https://www.acronis.com/es-es/blog/posts/data-backup-and-recovery-best-practices-for-the-manufacturing-industry/>
22. Estrategias de copia de seguridad de datos: Consejos para garantizar la continuidad de negocio y reducir el tiempo de inactividad | Arcserve,
<https://www.arcserve.com/es/blog/data-backup-strategies-tips-ensuring-business-continuity-and-minimizing-downtime>
23. The Role of the DBA in the Age of Cloud Computing - Database Trends and Applications, <https://www.dbta.com/Editorial/Think-About-It/The-Role-of-the-DBA-in-the-Age-of-Cloud-Computing-160009.aspx>
24. Cloud Database vs Traditional Database Advice | WellData,
<https://www.welldata.co.uk/knowledge-base/cloud-database-vs-traditional-database/>
25. ¿Qué es una base de datos en la nube? | Google Cloud,
<https://cloud.google.com/learn/what-is-a-cloud-database?hl=es-419>
26. What Is A Cloud Database vs. A Traditional Database? - Couchbase,
<https://www.couchbase.com/resources/concepts/what-is-a-cloud-database/>
27. The Role of DevOps in Cloud Computing Transformation - Invensis Learning, acceso: agosto 23, 2025, <https://www.invensislearning.com/blog/impact-cloud-computing-in-devops/>
28. ¿Qué es la administración en la nube? - Explicación de los servicios ..., <https://aws.amazon.com/es/what-is/cloud-management/>
29. El Rol del DBA en Migraciones a la Nube: Técnica y Estrategia para un Éxito Sostenible, <https://www.gbsys.com/el-rol-del-dba-en-migraciones-a-la-nube/>
30. The DBA's Role in DevOps - Liquibase,
<https://www.liquibase.com/resources/whitepapers/dba-role-devops>
31. Of DevOps, Databases and DBAs, <https://devops.com/of-devops-databases-and-dbas/>

32. ¿Qué es DevOps? Explicación de DevOps | Microsoft Azure, <https://azure.microsoft.com/es-es/resources/cloud-computing-dictionary/what-is-devops>
33. Herramientas DevOps para la automatización de la implementación de bases de datos, <https://www.quest.com/mx-es/solutions/devops/>
34. Database Administrator: Managing Big Data Analytics | MoldStud, <https://moldstud.com/articles/p-database-administrator-managing-big-data-analytics>
35. DOCTORATE OF BUSINESS ADMINISTRATION - ESCE, <https://www.esce.fr/en/program/esce-dba-international-management-in-big-data-robots/>
36. ¿Cuál es la diferencia entre las bases de datos relacionales y las no relacionales? - AWS, <https://aws.amazon.com/es/compare/the-difference-between-relational-and-non-relational-databases/>
37. NoSQL vs SQL: Las diferencias + cuándo usar cada una - Coursera, <https://www.coursera.org/mx/articles/nosql-vs-sql>
38. Bases de datos SQL vs NoSQL: Diferencias clave e ideas prácticas - DataCamp, <https://www.datacamp.com/es/blog/sql-vs-nosql-databases>
39. Base de datos relacional vs. no relacional: diferencias clave para la gestión de datos, <https://www.intersystems.com/es/recursos/base-de-datos-relacional-vs-no-relacional-diferencias-clave-para-la-gestion-moderna-de-datos/>
40. Comparativa: Herramientas para la Gestión de Bases de Datos en 2024 - Bmind, <https://bmind.com/comparativa-herramientas-para-la-gestion-de-bases-de-datos-en-2024/>
41. DB-Engines Ranking - popularity ranking of database management systems, <https://db-engines.com/en/ranking>
42. SQL vs NoSQL comparison: MySQL, PostgreSQL, MongoDB & Cassandra - Devathon, <https://devathon.com/blog/sql-vs-nosql-mysql-vs-postgresql-vs-mongodb-vs-cassandra/>
43. Las 10 mejores herramientas de consulta de bases de datos para una gestión eficiente de datos - Apidog, <https://apidog.com/es/blog/best-database-query-tools-3/>

44. 10 Cliente NoSQL para la administración y el desarrollo de bases de datos - Geekflare, <https://geekflare.com/es/nosql-client/>
45. Database Administrators and Architects : Occupational Outlook Handbook, <https://www.bls.gov/ooh/computer-and-information-technology/database-administrators.htm>
46. Administradores de Bases de Datos y Redes de Computadora: Salarios, diversidad, industrias e informalidad laboral, <https://www.economia.gob.mx/datamexico/es/profile/occupation/administradores-de-bases-de-datos-y-redes-de-computadora>
47. Salario de DBA en México 2025 - Computrabajo, <https://mx.computrabajo.com/salarios/dba>
48. Bolsa de trabajo para dba en CDMX (Ciudad de México DF) | OCC, <https://www.occ.com.mx/empleos/de-dba/en-ciudad-de-mexico/>