



**INSTITUTO POLITECNICO NACIONAL
UNIDAD PROFESIONAL INTERDISCIPLINARIA EN
INGENIERIA Y TECNOLOGIAS AVANZADAS**

Práctica 4

Replicación

**INGENIERIA TELEMÁTICA
BASE DE DATOS DISTRIBUIDAS**

PROFESOR: De la Cruz Sosa Carlos

ALUMNOS:

Legorreta Rodríguez María Fernanda

Macías Galván Arturo Daniel

Palacios Reyes Leslie Noemi

**GRUPO: 3TM3
SEMESTRE 2025/2**

1. Introducción

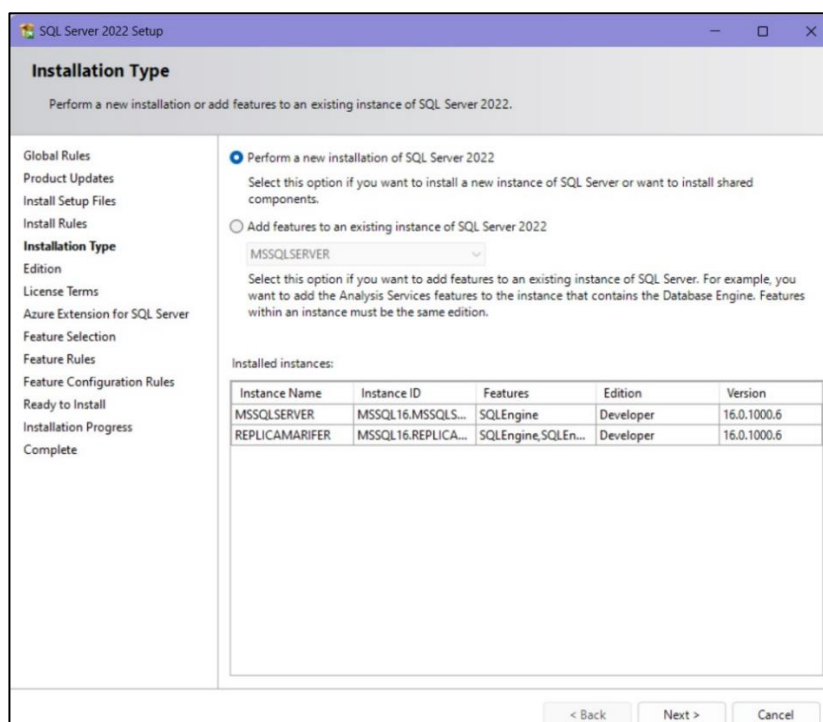
La replicación de base de datos se usa para garantizar la disponibilidad, escalabilidad y tolerancia a fallos en los sistemas de información. En la actualidad, las bases de datos replicadas son una herramienta muy útil debido a que permiten que múltiples copias de una misma información estén disponibles en distintos servidores, mejorando el rendimiento de las consultas, reduciendo los tiempos de respuesta y asegurando la continuidad del servicio ante fallas del sistema principal.

En esta ocasión, trataremos de replicar la base de datos *covid* que hemos estado utilizando a lo largo de todo el curso de Bases de Datos Distribuidas, por lo que implementaremos una arquitectura distribuida basada en replicación asincrónica, gracias a esto inicialmente se realizará una replicación transaccional completa para transferir todo el conjunto de datos a otro nodo, y posteriormente se aplicará replicación mediante instantáneas para mantener actualizada la información.

El enfoque adoptado es de replicación completa, ya que se desea disponer de una copia íntegra de la tabla de datos *covid* en uno o más nodos de la red, permitiendo así la ejecución de consultas distribuidas y la evaluación del rendimiento del sistema en un entorno simulado de bases de datos distribuidas.

2. Estrategias de conectividad

Al igual que en la práctica anterior para la replicación de base de datos tendremos instancias en diferentes nodos que deben estar conectados en red, además de que cada una de estas debe permitir la replicación que se activa desde la configuración.



SQL Server 2022 Setup

Feature Selection

Select the Developer features to install.

Global Rules

Product Updates

Install Setup Files

Install Rules

Installation Type

Edition

License Terms

Azure Extension for SQL Server

Feature Selection

Feature Rules

Instance Configuration

Server Configuration

Database Engine Configuration

Analysis Services Configuration

Integration Services Scale Out ...

Integration Services Scale Out ...

Feature Configuration Rules

Ready to Install

Installation Progress

Complete

Looking for Reporting Services? [Download it from the web](#)

Features:

Instance Features

☒ Database Engine Services

☒ SQL Server Replication

☒ Machine Learning Services and Language Ext

☒ Full-Text and Semantic Extractions for Search

☒ Data Quality Services

☐ PolyBase Query Service for External Data

☒ Analysis Services

Shared Features

☒ Data Quality Client

☒ Integration Services

☒ Scale Out Master

☒ Scale Out Worker

☒ Master Data Services

Redistributable Features

Feature description:

The configuration and operation of each instance feature of a SQL Server instance is isolated from other SQL Server instances. SQL Server instances can operate side-by-side on the same computer.

Prerequisites for selected features:

Already installed:

Windows PowerShell 3.0 or higher

Microsoft Visual C++ 2017 Redistributable

Microsoft Visual Studio Tools for Application

Disk Space Requirements

Drive C: 3178 MB required, 244063 MB available

Select All

Unselect All

Instance root directory:

C:\Program Files\Microsoft SQL Server\

...

Shared feature directory:

C:\Program Files\Microsoft SQL Server\

...

Shared feature directory (x86):

C:\Program Files (x86)\Microsoft SQL Server\

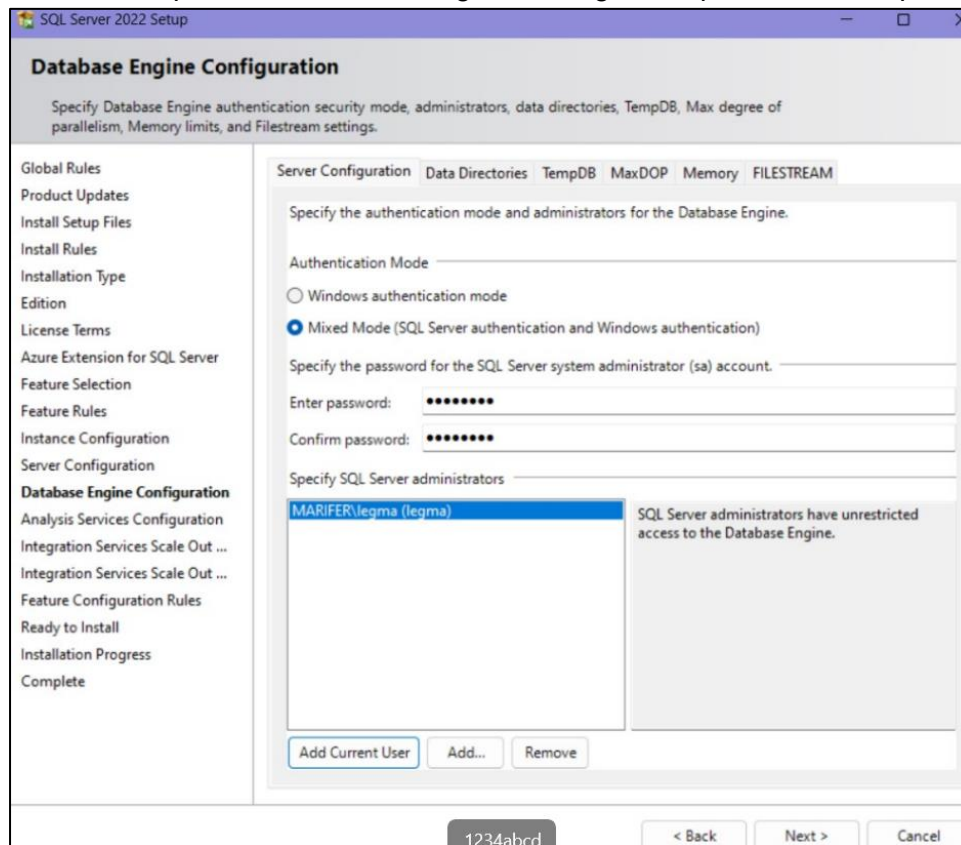
...

< Back

Next >

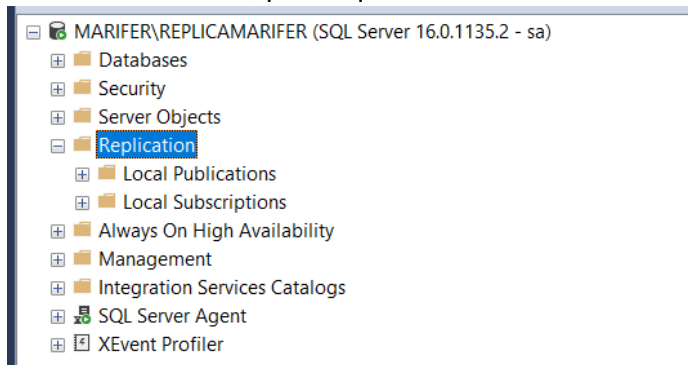
Cancel

Además de que necesitamos configurar la seguridad para nuestra replicación:



3. Configuración de Publicador

Una vez realizada la configuración de las instancias en SQL Server podemos configurar nuestro Publicador, por lo que debemos entrar a la sección de *Replication*

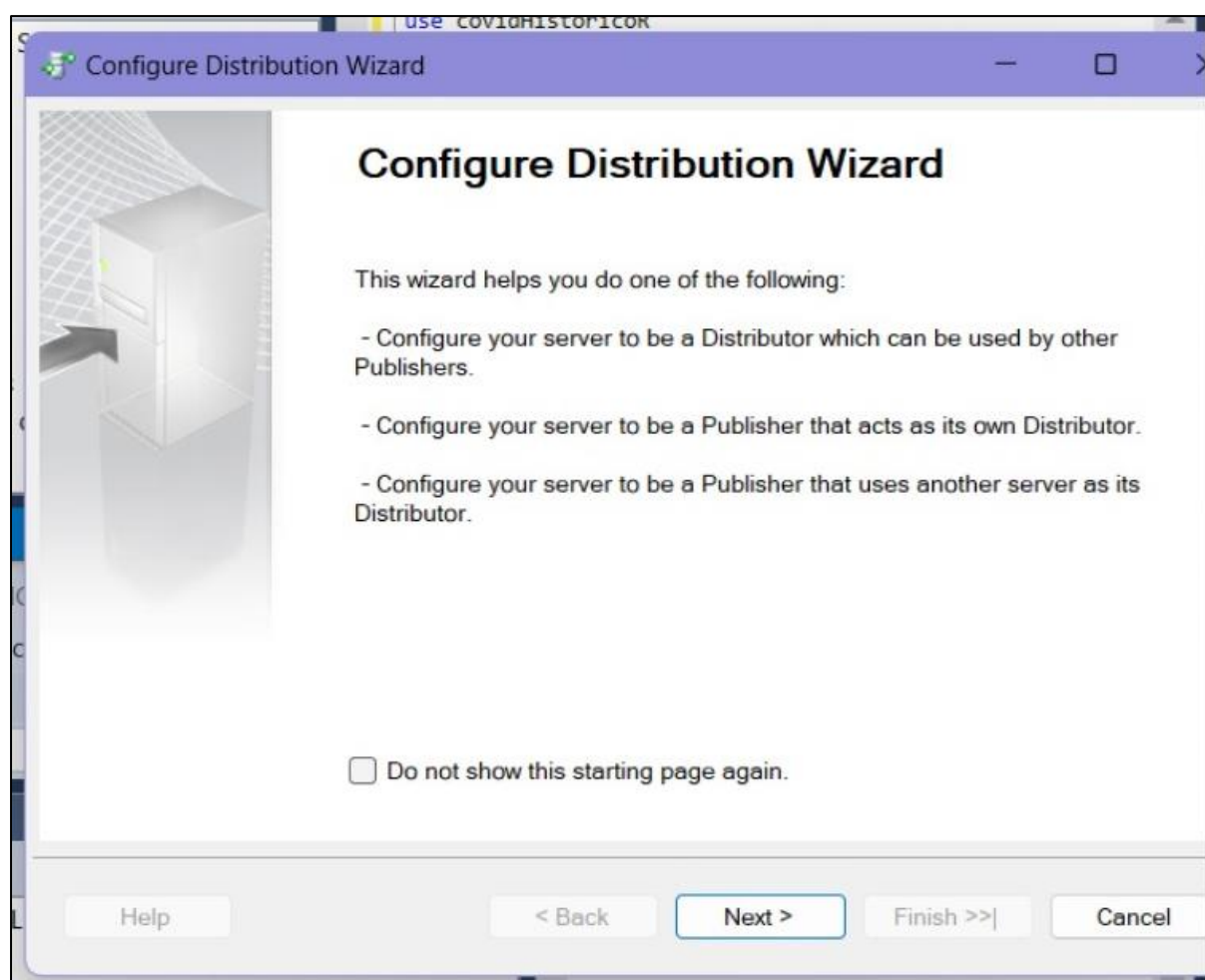


Para esta ocasión necesitamos conocer los componentes que actúan en la replicación de base de datos:

- **SQL Server (MSSQLSERVER)** Es el motor principal de base de datos. Gestiona las bases de datos, consultas, transacciones, etc.

- SQL Server Agent Ejecuta tareas automatizadas como los agentes de replicación (Snapshot, Log Reader, Distribution).

 SQL Server Agent (PUBLICADORMF)	Executes job...	
 SQL Server Agent (REPLICAMARIFER)	Executes job...	
 SQL Server Analysis Services (PUBLICADORMF)	Supplies onli...	En eje
 SQL Server Analysis Services CEIP (PUBLICADORMF)	CEIP service f...	En eje



Configure Distribution Wizard

Publishers

Enable servers to use this Distributor when they become Publishers.

Publishers:

Publisher	Distribution Database
<input checked="" type="checkbox"/> MARIFER\PUBLICADORMF	publicadorCovid

Add

Help < Back Next > Finish >>| Cancel

Configure Distribution Wizard

Configuring...

Click Stop to interrupt the operation.

3 Remaining 3 Total 0 Error 0 Success 0 Warning

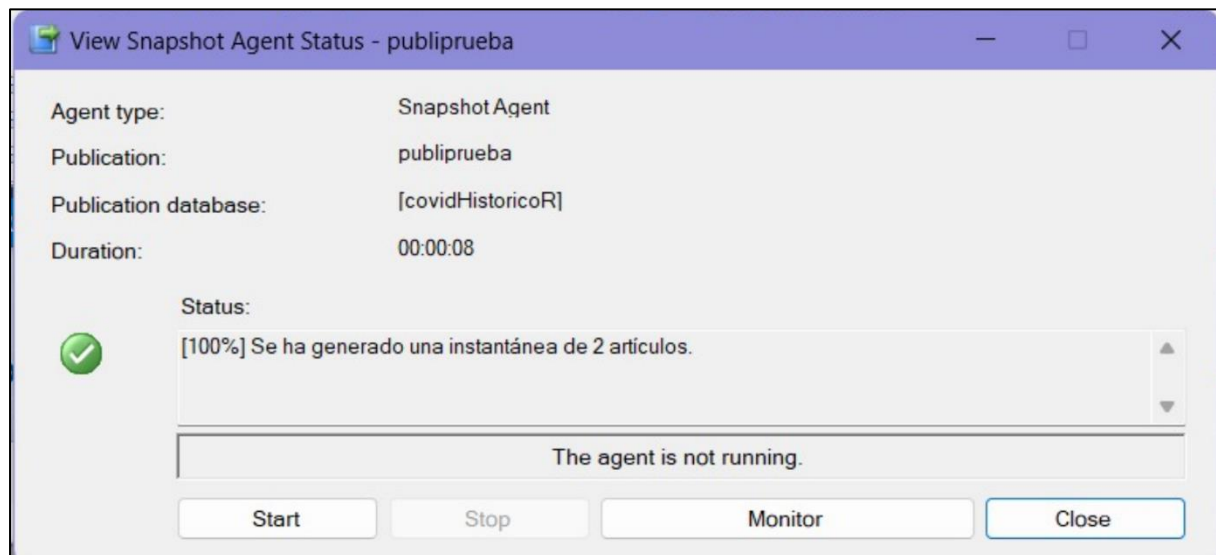
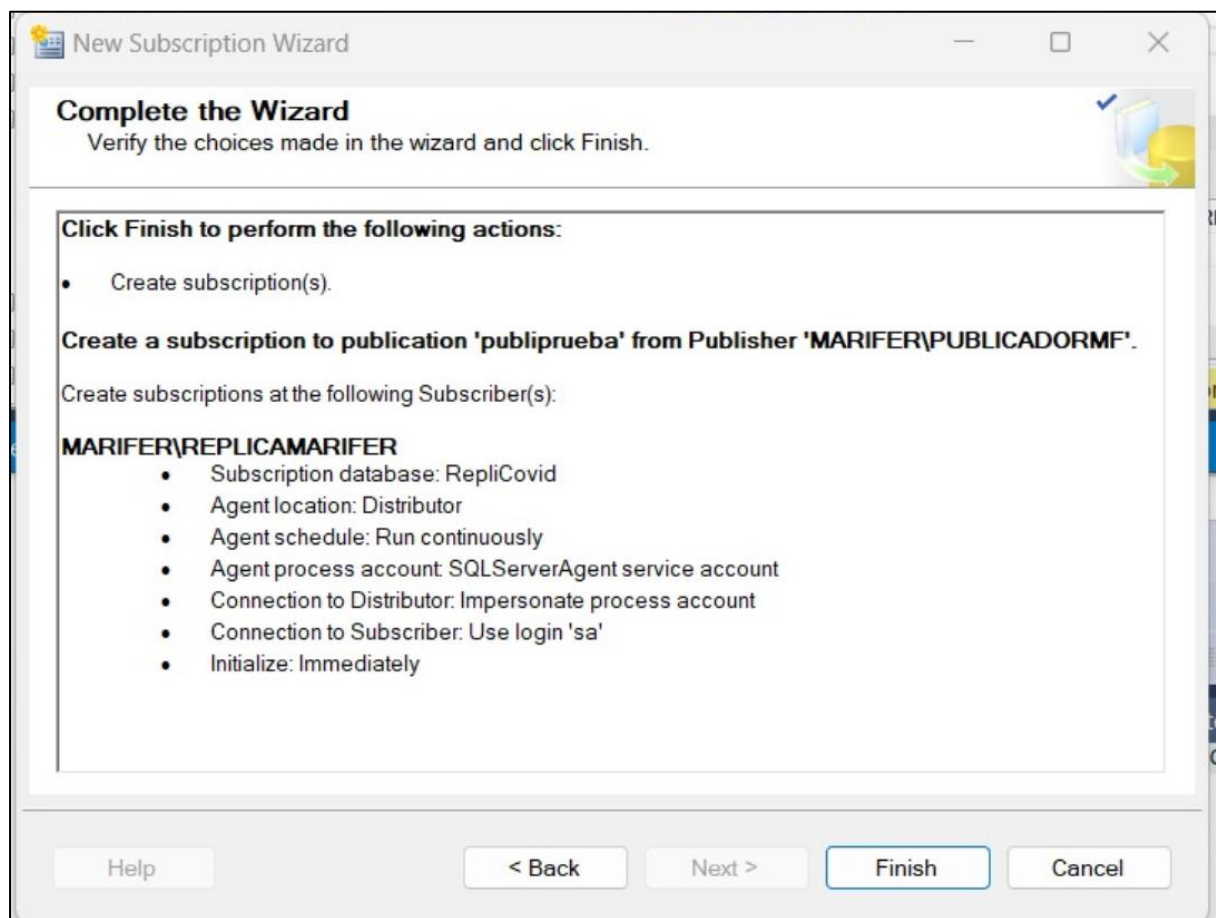
Details:

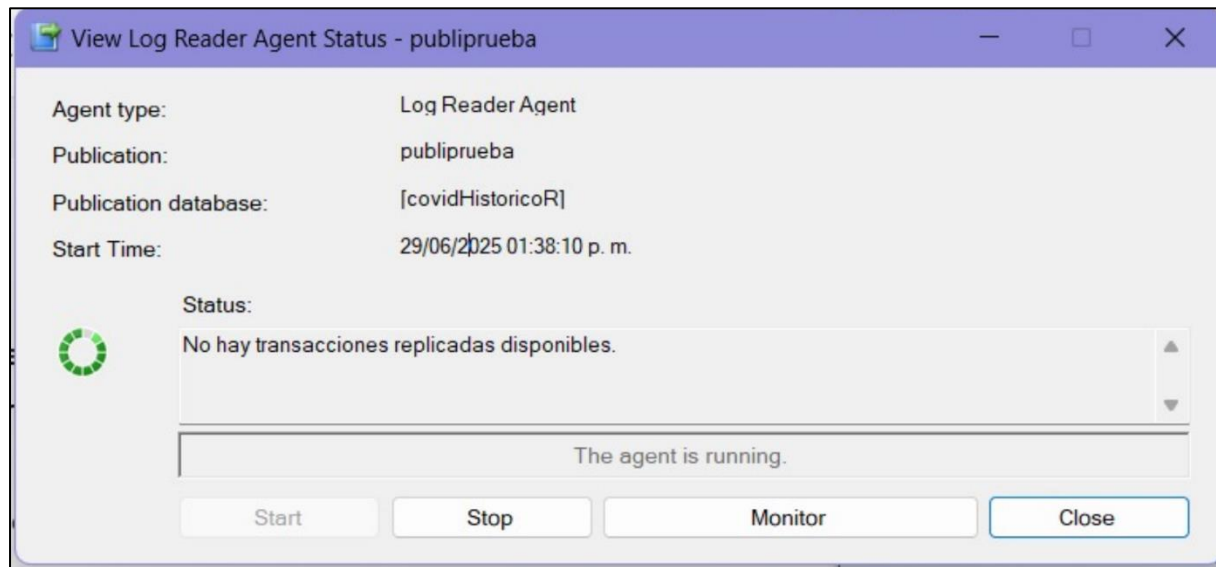
Action	Status	Message
Configuring the Distributor	In progress...	
Enabling Publisher 'MARIFER\PUBLICADORMF'		
Configuring SQL Server Agent to start automa...		

Stop Report

Close

Moves to the last wizard page and takes the default values for all

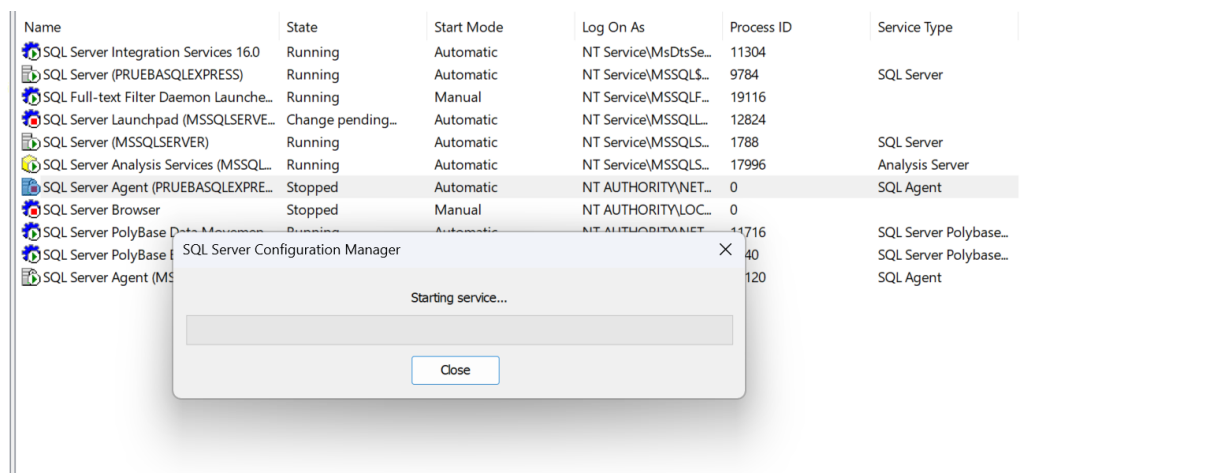




Intento dos (Leslie Noemi)

Después de tener lista mi instancia con SQL Server Developer Edition, comencé con la configuración del entorno de replicación. En un principio, todo parecía estar en orden: la publicación se generó correctamente y las bases estaban bien sincronizada, hasta que llegó el momento de trabajar con los agentes.

Me encontré con un detalle importante: el SQL Server Agent no estaba reconociendo correctamente los servicios necesarios para la replicación. Esto provocó que los agentes (especialmente el Snapshot Agent) no se ejecutaran como debían, afectando la inicialización de las suscripciones.



Configuración de Usuario con Permisos de Administración

Llegado a este punto del proceso, fue necesario crear un nuevo usuario en la base de datos para asegurar el correcto funcionamiento de los agentes y permitir una gestión más controlada de la replicación.

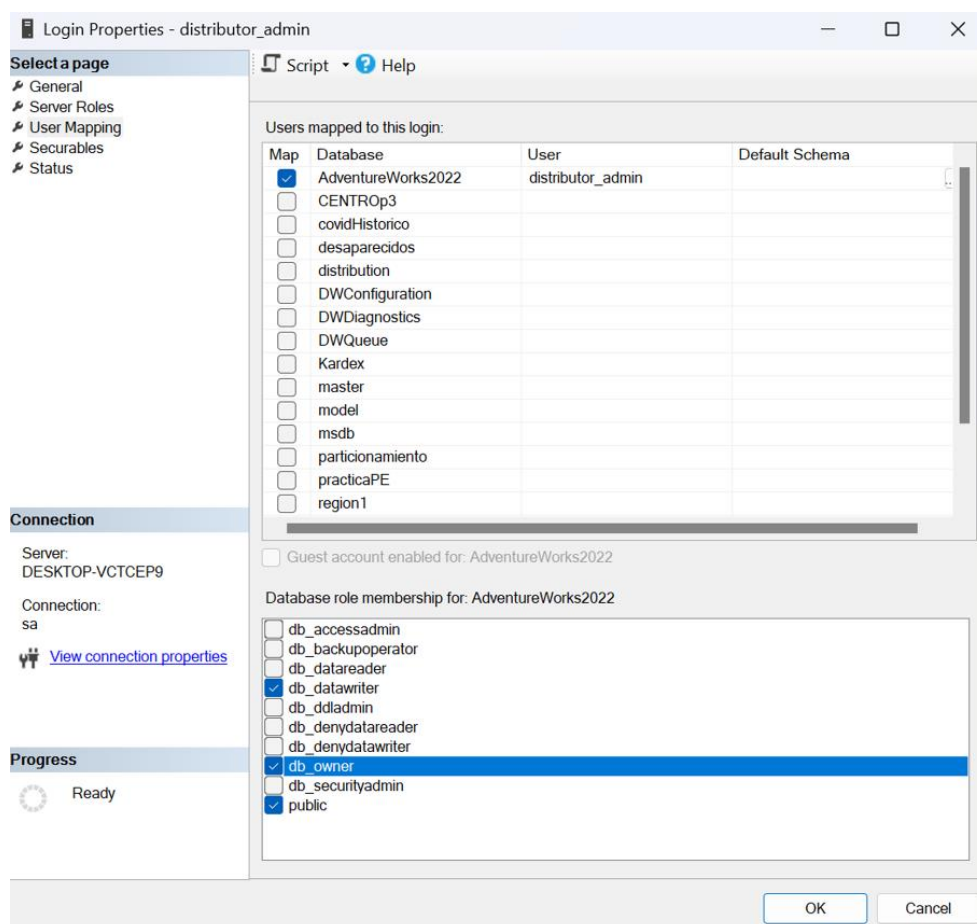
Este usuario fue asignado con el rol de sysadmin desde la sección de servidor (Server Roles) dentro de SQL Server Management Studio. Esto fue crucial, ya que, sin estos privilegios, el usuario no habría tenido acceso suficiente para ejecutar tareas como inicializar el Snapshot Agent, administrar publicaciones y suscripciones, o manipular los objetos replicados.

Además, en la pestaña de User Mapping, se seleccionó específicamente la base de datos AdventureWorks2022, que era la que se quería manipular. Dentro de esta base, se le otorgaron roles adicionales como:

- db_owner
- db_datareader
- db_datawriter

Esto permitió que el usuario pudiera leer, escribir y administrar completamente la base replicada, garantizando así que todos los procesos asociados a la replicación tuvieran los permisos necesarios para ejecutarse sin restricciones.

La siguiente imagen muestra cómo se configuró este usuario en SQL Server Management Studio, detallando tanto su asignación de roles a nivel de servidor como los permisos específicos a nivel de base de datos.



Configuración de la Replicación Transaccional

Una vez finalizada la configuración del usuario con permisos adecuados, procedimos a trabajar directamente con la replicación, específicamente utilizando el modelo de replicación transaccional.

Este tipo de replicación se caracteriza por ser altamente automatizada, lo cual facilita mucho su implementación. A diferencia de otras modalidades como la replicación de tipo snapshot o merge, aquí no es necesario realizar acciones manuales constantes, ya que las transacciones se capturan y envían automáticamente desde el publicador hacia el suscriptor casi en tiempo real.

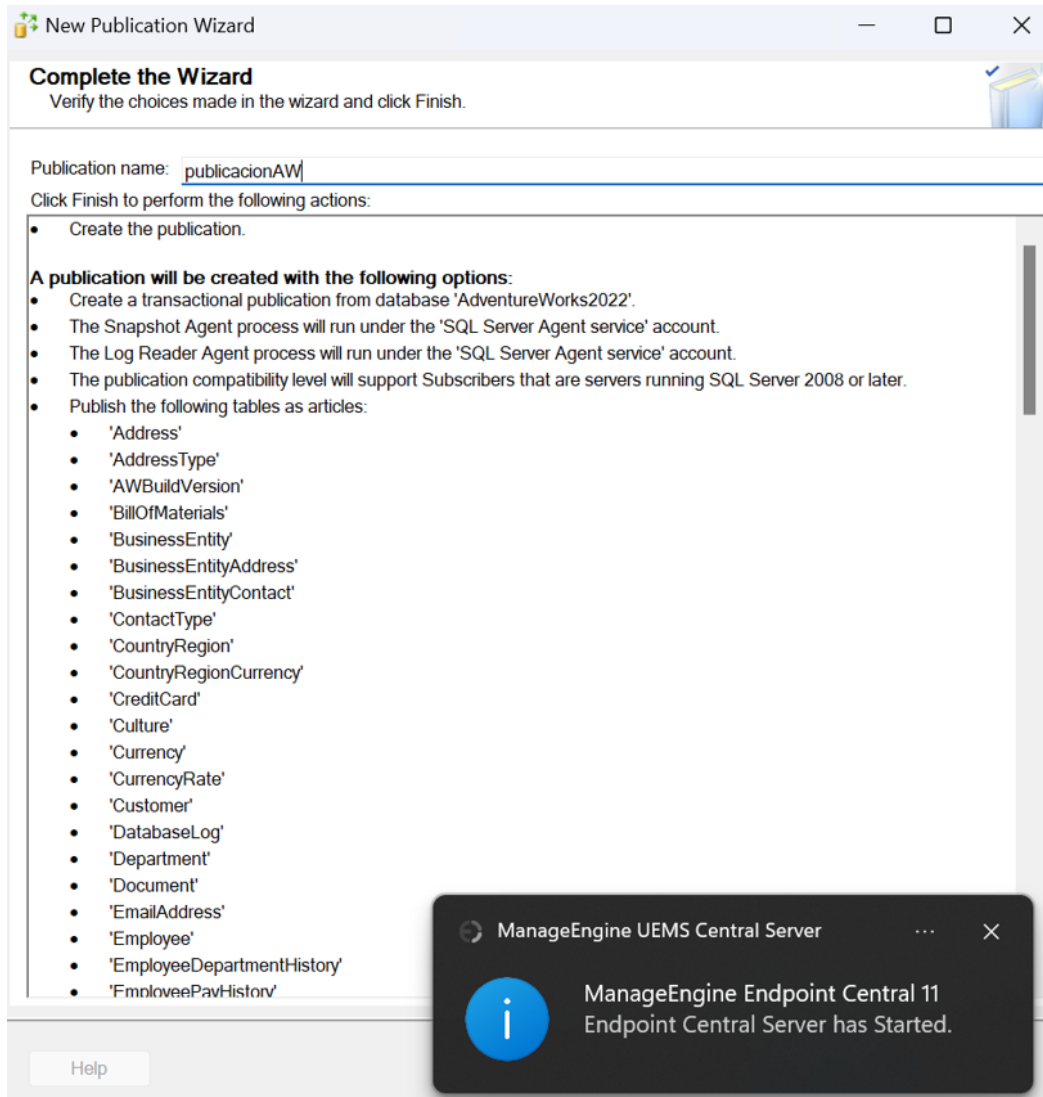
Para esta práctica, seleccioné como base de datos origen a AdventureWorks2022, ya que me pareció adecuada debido a la cantidad y diversidad de tablas, vistas y procedimientos almacenados que contiene. Esto ofrecía un escenario más completo para simular una replicación realista y robusta.

Durante la configuración de la publicación, el asistente de SQL Server permite seleccionar de forma previa qué objetos se desean replicar. En mi caso, opté por incluir todas las tablas, vistas y procedimientos almacenados disponibles en la base de datos. Este enfoque completo me permitió observar cómo se comportan diferentes tipos de objetos dentro del proceso de replicación.

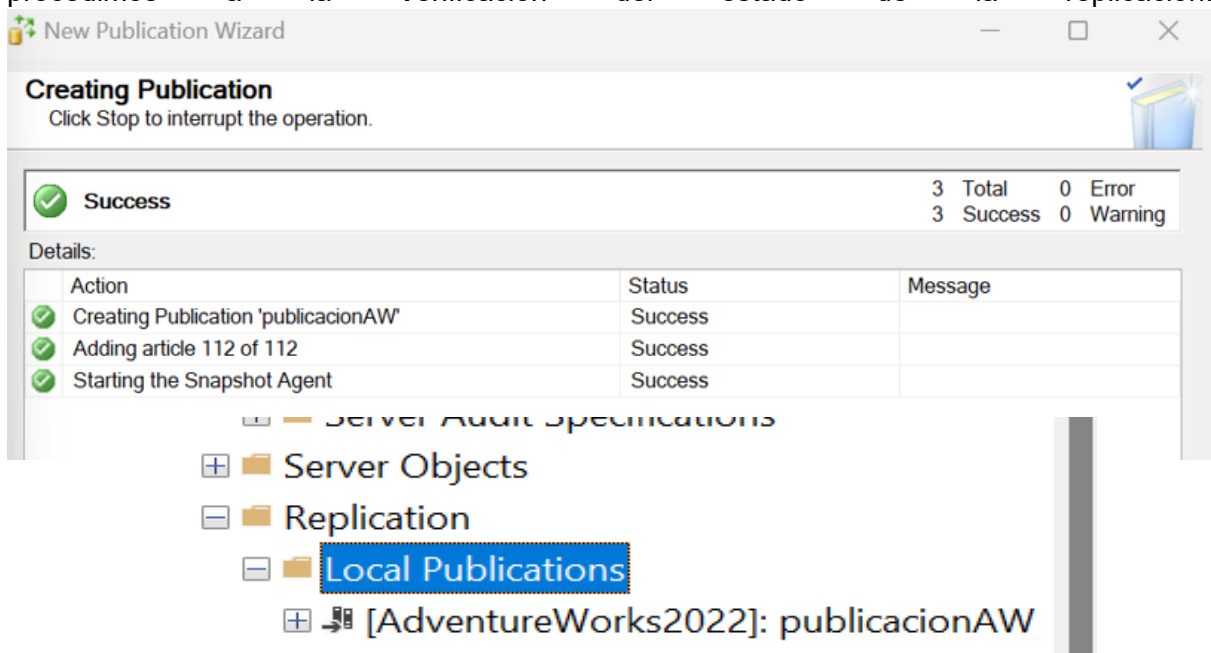
Al finalizar esta configuración, el sistema genera automáticamente un checklist detallado con todos los objetos seleccionados para replicación. Este resumen es útil para confirmar qué elementos serán enviados al suscriptor y validar que todo esté correctamente configurado.

A continuación, se muestra una imagen que refleja ese checklist, donde se puede apreciar claramente el conjunto de objetos replicados seleccionados

.

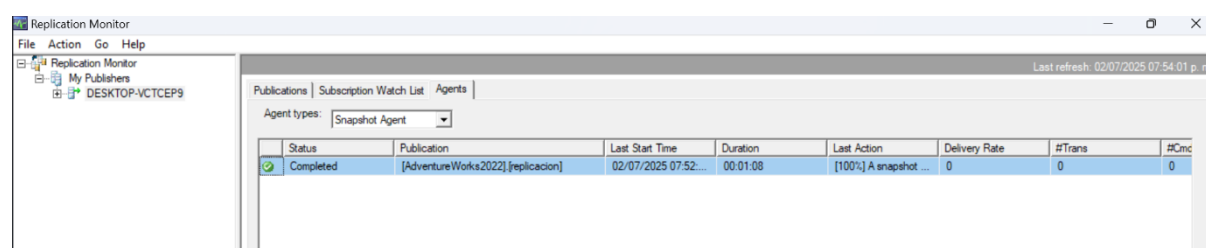


Una vez finalizada la configuración de la publicación y la selección de los objetos a replicar, procedimos a la verificación del estado de la replicación.

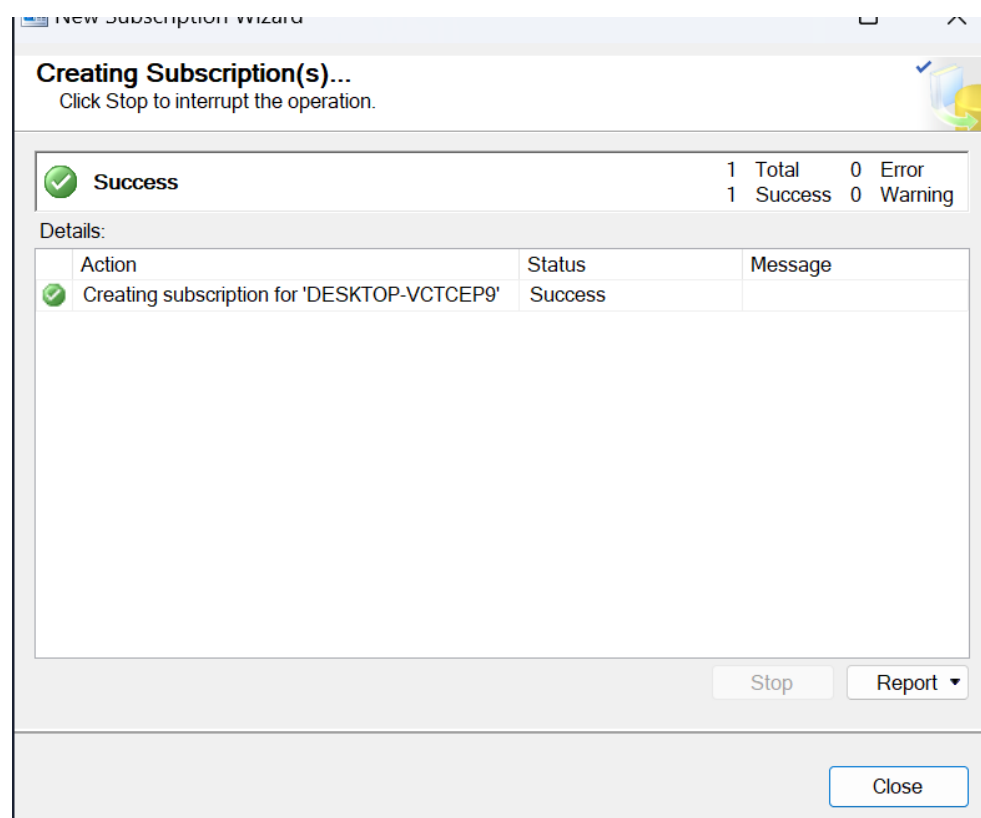


Verificación del Estado del Agente antes de la Suscripción Antes de proceder con la creación de la suscripción, es indispensable asegurarse de que el Snapshot Agent esté funcionando correctamente. Esto se debe a que dicho agente es el encargado de generar la “captura” inicial de los objetos y datos definidos en la publicación, y sin esa captura, la suscripción no puede ser inicializada correctamente.

Para ello, se debe verificar que el agente esté corriendo y que su estado sea “sin errores”, lo cual se puede hacer desde el Replication Monitor o directamente desde el árbol de replicación en SQL Server Management Studio. Si el agente no se encuentra en ejecución o presenta fallos, es probable que la replicación no funcione como se espera, lo cual puede ocasionar errores en la sincronización.



Hasta aquí, todo va bien, incluso cuando creamos a la suscripción, todo parece estar bajo control.

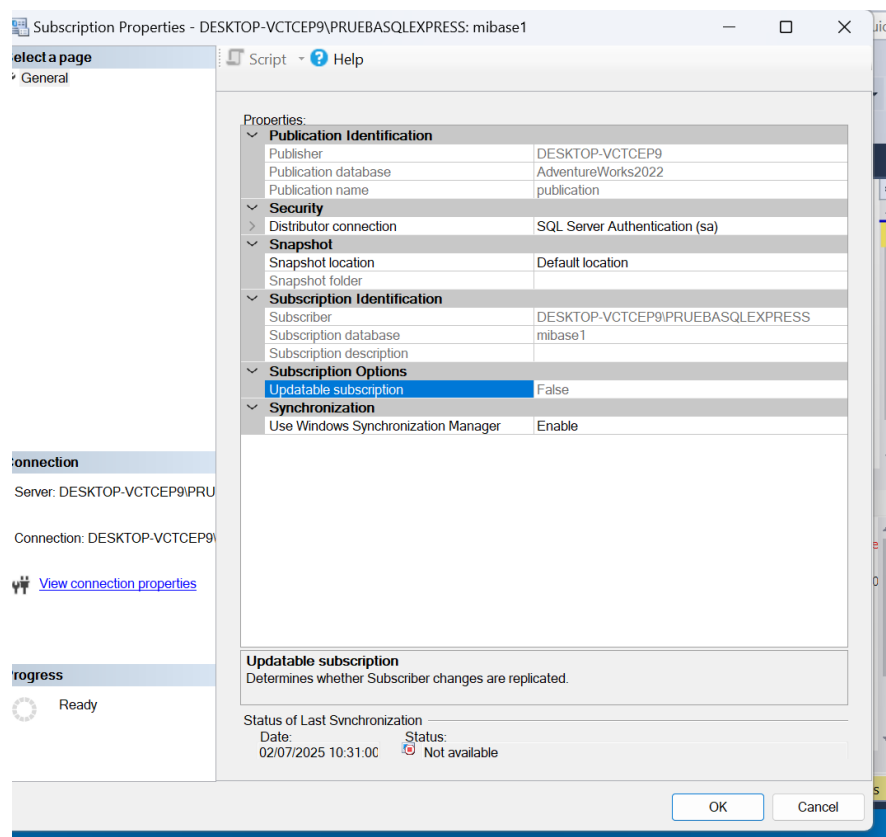


Problemas al Replicar en la Otra Instancia

Sin embargo, al intentar conectar la publicación con una segunda instancia para aplicar la suscripción, nos llevamos una gran sorpresa: las réplicas de las tablas no se habían transferido correctamente. A pesar de que toda la configuración inicial parecía estar en orden y el agente se encontraba activo, los objetos que seleccionamos para replicar no aparecían en la instancia suscriptora.

Ante este inconveniente, comenzamos una etapa de diagnóstico. Recurriendo a videos, tutoriales de Microsoft y documentación oficial, seguimos paso a paso las recomendaciones, replicando el proceso al detalle. No obstante, el problema persistía.

Después de una revisión minuciosa, observamos que en ciertos scripts o configuraciones había errores de sintaxis, nombres mal escritos o parámetros transpuestos.



Conclusiones:

- Legorreta Rodríguez María Fernanda:

Para esta práctica me pareció muy interesante el uso que se le puede dar a la replicación de bases de datos, ya que permite mantener sincronizados distintos servidores en escenarios donde se requiere disponibilidad de la información en tiempo real o distribuir la carga de trabajo entre varios sistemas. Me llamó especialmente la atención conocer las diferentes características y tipos de replicación que existen, como la replicación transaccional, de mezcla y de instantánea, cada una con aplicaciones específicas dependiendo del entorno.

Además, fue muy interesante ver cómo funciona SQL Server en este proceso, gracias a la intervención de varios componentes importantes como los agentes de replicación. Estos agentes son los encargados de realizar tareas clave, como la verificación de las publicaciones, la distribución de los datos y la sincronización entre el publicador y el suscriptor.

- Macias Galván Daniel Arturo

Durante este proceso, Arturo logró identificar la importancia de la precisión en la configuración de los elementos de replicación, especialmente al trabajar con múltiples instancias. Esta experiencia reforzó su capacidad de análisis y diagnóstico técnico, enfrentando errores no documentados y aprendiendo a validar cada paso de forma sistemática. La replicación no solo fue un ejercicio técnico, sino también un reto de atención al detalle y persistencia.

- Palacios Reyes Leslie Noemi

Esta práctica representó un aprendizaje valioso sobre cómo pequeños errores (como rutas incorrectas, nombres mal escritos o permisos insuficientes) pueden interrumpir completamente un proceso de replicación que en teoría estaba bien estructurado. A través de la investigación, prueba y error, fortaleció su comprensión del entorno de SQL Server y desarrolló una mayor confianza en su capacidad para enfrentar problemas reales con herramientas de diagnóstico y documentación oficial.