



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

Laboratorio de:

Bases de datos distribuidas

Práctica No.:

4

Grupo No.:

9

Integrantes:

- Fernando Eliceo Huilca Villagómez
- Gregory Leandro Salazar Domínguez
- Mateo Nicolás Simbaña Guarnizo

Tema:

Implementación de la replicación en la base de datos distribuida de una red de gimnasios

Objetivos:

1. Identificar las relaciones a replicar en el entorno distribuido mediante el esquema de replicación definido en la fase de diseño.
2. Configurar los componentes necesarios para cada replicación, como el Distribuidor, el Publicador y el Suscriptor, de acuerdo con el tipo de replicación designado en cada caso.
3. Probar el funcionamiento de las tablas replicadas a través de operaciones CRUD, con la finalidad de validar que las configuraciones se hayan aplicado de manera correcta.

Marco teórico:

1. Criterios para decidir sobre la replicación de datos

Por lo general, las bases de datos distribuidas son replicadas, por lo que es importante comprender los beneficios y las posibles dificultades asociadas a su implementación.

1.1 Motivos para replicar

Se replica una base de datos o una tabla por las siguientes razones [1]:



- **Disponibilidad del sistema:** La replicación elimina puntos de falla únicos, permitiendo que los datos sean accesibles desde varios sitios, incluso si algunos están fuera de servicio.
- **Rendimiento:** La replicación reduce la sobrecarga de comunicación al acercar los datos a los puntos de acceso, mejorando el tiempo de respuesta.
- **Escalabilidad:** La replicación facilita el crecimiento del sistema en términos geográficos y de número de sitios sin afectar los tiempos de respuesta.
- **Requerimientos de la aplicación:** Algunas aplicaciones requieren mantener múltiples copias de los datos como parte de sus especificaciones operativas.

1.2 Motivos para no replicar

Aunque la replicación de datos ofrece beneficios, también plantea algunas dificultades, que se detallan a continuación [1]:

- **Transparencia de replicación:** Mientras los usuarios interactúan con los datos lógicos, el sistema debe gestionar varias copias físicas. Esto puede complicar la sincronización de las réplicas y garantizar que todas se mantengan consistentes, a pesar de que los usuarios perciban que solo existe una copia.
- **Problemas de consistencia:**
 - **Actualizaciones costosas:** Mantener las réplicas sincronizadas es complejo y puede generar una sobrecarga considerable, especialmente en sistemas con actualizaciones frecuentes.
 - **Impacto en la disponibilidad:** Si el protocolo de replicación no se gestiona adecuadamente, las actualizaciones a las réplicas pueden afectar el rendimiento y reducir la disponibilidad del sistema.

2. Estrategias de gestión de actualizaciones

La replicación utiliza diferentes estrategias de gestión de actualizaciones, que se clasifican según el momento en que se propagan las actualizaciones a las copias (replicación ansiosa o perezosa) y el lugar donde se realizan las actualizaciones (centralizada o distribuida) [1].

2.1 Replicación ansiosa

En la replicación ansiosa, las actualizaciones se aplican a todas las réplicas durante la transacción. La transacción no termina hasta que todas las copias estén sincronizadas, lo que garantiza consistencia mutua. Sin embargo, esto



puede afectar el rendimiento y disponibilidad, puesto que, si una réplica está inactiva, la transacción no se puede completar [1].

2.2 Replicación perezosa

La replicación perezosa permite que la transacción finalice tan pronto como se actualice una réplica, y las actualizaciones a las demás réplicas se realizan de manera asincrónica. Esto mejora el rendimiento al reducir los tiempos de espera, pero puede generar inconsistencia entre las réplicas, lo que podría ocasionar que se lean datos desactualizados [1].

2.3 Centralizada

En la replicación centralizada, las actualizaciones se realizan primero en una copia maestra y luego se propagan a las réplicas, denominadas esclavas. Esto simplifica el proceso de actualización, pero puede sobrecargar el sitio maestro, creando cuellos de botella. Es adecuado en aplicaciones con procesamiento centralizado [1].

2.4 Distribuida

La replicación distribuida permite que las actualizaciones se realicen en las copias locales de cada sitio y luego se propaguen a otras réplicas. Esto distribuye la carga y mejora la disponibilidad, pero puede generar problemas de sincronización y consistencia si las actualizaciones se realizan en diferentes órdenes en distintos sitios [1].

Se pueden combinar las dos formas de clasificación para crear protocolos que se ajusten a las necesidades del sistema, ya sea ansiosa o perezosa, centralizada o distribuida.

3. Elementos que participan en la replicación

La replicación emplea una metáfora de la industria editorial para ilustrar los componentes de una topología de replicación, que incluyen al editor, Distribuidor, Suscriptores, publicaciones, artículos y Suscripciones. La replicación de Microsoft SQL Server en términos de una revista se gestiona de la siguiente manera [2]:

- Un **Publicador** de revistas produce una o más **publicaciones**.
- Una **publicación** contiene **artículos**.
- El Publicador distribuye la revista directamente o utiliza un **Distribuidor**.
- Los **Suscriptores** reciben publicaciones a las que se han **suscrito**.



En la Figura 1, se presenta una descripción general de los componentes y procesos involucrados en la replicación.

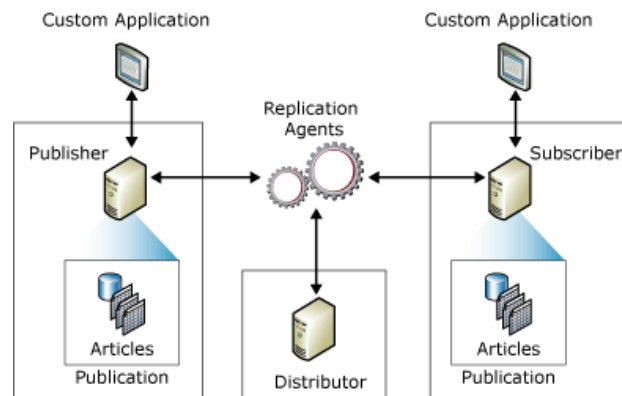


Figura 1. Elementos y procesos que conforman una configuración para la replicación.

3.1 Publicador

El Publicador es una instancia de base de datos que pone los datos a disposición de otros sitios mediante replicación. Un Publicador puede tener una o más publicaciones, cada una de las cuales define un conjunto de objetos y datos relacionados lógicamente para replicar [2].

3.2 Distribuidor

El Distribuidor es una instancia de base de datos que almacena los datos de replicación asociados con uno o más Publicadores. Cada Publicador tiene una base de datos de distribución que guarda el estado de la replicación, metadatos de la publicación y, a veces, actúa como cola para los datos entre el Publicador y los Suscriptores [2].

Cuando el Publicador y el Distribuidor están en el mismo servidor, se llama Distribuidor local; si están en servidores distintos, se lo conoce como Distribuidor remoto [2].

3.3 Suscriptor

Un Suscriptor es una instancia de base de datos que recibe los datos replicados. Puede recibir datos de varios Publicadores y publicaciones. Según el tipo de replicación seleccionado, el Suscriptor también puede enviar los cambios de datos de vuelta al Publicador o redistribuir los datos a otros Suscriptores [2].

3.4 Artículo



Un artículo representa un objeto de base de datos que forma parte de una publicación. Una publicación puede incluir diferentes tipos de artículos, como tablas, vistas, procedimientos almacenados y otros objetos. Cuando las tablas se publican como artículos, se pueden aplicar filtros para limitar las columnas y filas de los datos que se envían a los Suscriptores [2].

3.5 Publicación

Una publicación es un conjunto de uno o más artículos de una base de datos. Agrupar varios artículos en una publicación facilita la especificación de un conjunto de objetos y datos de base de datos relacionados lógicamente que se replican como una unidad [2].

3.6 Suscripción

Una suscripción es una solicitud para recibir una copia de una publicación por parte de un Suscriptor. La suscripción define qué publicación se recibirá, dónde y cuándo. Existen dos tipos de Suscripciones: push y pull [2].

3.6.1 Suscripciones push

En una suscripción push, el Distribuidor actualiza directamente los datos en la base de datos del Suscriptor sin que este tenga que solicitarlo [3].

3.6.2 Suscripciones pull

En una suscripción pull, el Suscriptor consulta de forma periódica al Distribuidor para verificar si hay nuevos cambios disponibles, y si los hay, actualiza los datos en su propia base de datos de suscripción [3].

4. Tipos de replicación

Microsoft SQL Server proporciona los siguientes tipos de replicación para usarlos en las aplicaciones distribuidas:

4.1 Replicación instantánea

La replicación de instantáneas distribuye los datos tal como aparecen en un momento específico, sin realizar un seguimiento de las actualizaciones posteriores. Se genera y envía una instantánea completa a los usuarios [4].



Este tipo de replicación es ideal cuando los cambios en los datos son poco frecuentes. Por otro lado, es más lenta que la replicación transaccional, dado que implica mover varios registros de un extremo a otro en cada intento, sin embargo, es útil para realizar una sincronización inicial entre el Publicador y el Suscriptor [4].

4.2 Replicación transaccional

En la replicación transaccional, los usuarios reciben copias completas iniciales de la base de datos y luego actualizaciones conforme los datos cambian. Los datos se copian en tiempo real desde el Publicador a la base de datos receptora (Suscriptor), manteniendo el mismo orden en que ocurren en el Publicador, lo que garantiza la consistencia transaccional [4].

Este tipo de replicación es común en entornos de servidor a servidor, ya que no solo replica los cambios, sino que asegura que cada uno se realice de manera consistente y precisa [4].

4.3 Replicación de mezcla

En la replicación por fusión, los datos de dos o más bases de datos se combinan en una sola base de datos. Es el tipo de replicación más complejo, debido a que permite que tanto el Publicador como el Suscriptor realicen cambios de manera independiente [4].

Se utiliza principalmente en entornos de servidor a cliente y permite que los cambios se envíen de un Publicador a múltiples Suscriptores [4].

4.4 Replicación punto a punto

La replicación punto a punto es una solución que mejora la escalabilidad y la disponibilidad al mantener copias de los datos en varios servidores, llamados nodos [5].

Esta replicación, basada en la replicación transaccional, actualiza los datos de manera coherente en casi tiempo real. Esto permite distribuir las lecturas de los clientes entre los diferentes nodos, lo que ayuda a manejar un mayor volumen de solicitudes [5].

5. Topologías de replicación

Microsoft SQL Server admite las siguientes topologías de replicación:



5.1 Publicador central

Una de las topologías de replicación más utilizadas. En este caso, un servidor se configura como Publicador y Distribuidor, mientras que otro servidor o servidores se configuran como Suscriptor o Suscriptores [6].

Esta es la configuración que se utilizó en la implementación de la replicación. En la Figura 2, se ilustra cómo opera esta topología.

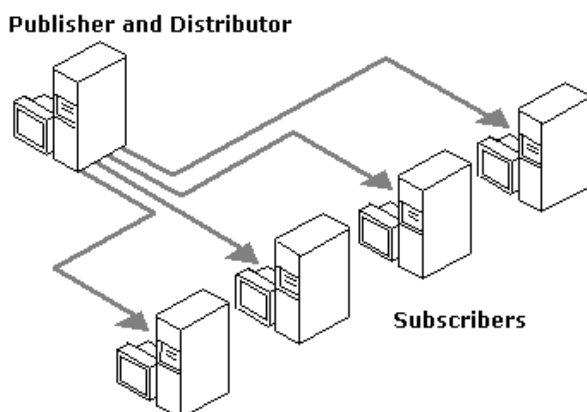


Figura 2. Funcionamiento de la topología Publicador central.

5.2 Suscriptor central

Esta es una topología común en el almacenamiento de datos. Varios servidores o bases de datos replican su información a un único servidor central en una o más bases de datos [6].

5.3 Distribuidor central

En esta topología, varios Publicadores utilizan un único Distribuidor, que se encuentra en un servidor distinto al de los Publicadores. Esta es una de las topologías de replicación menos utilizadas, puesto que presenta un único punto de fallo [6].

5.4 Publicador central con Distribuidor remoto

En esta topología, la base de datos de distribución se encuentra en un servidor distinto al del Publicador. Se utiliza para mejorar el rendimiento cuando aumenta la actividad de replicación o los recursos del servidor o la red se ven limitados. Reduce la carga del Publicador, pero incrementa el tráfico de red [6].

5.5 Suscriptor Publicador



En esta topología, dos servidores publican los mismos datos. Un servidor envía los datos a un Suscriptor, quien luego los distribuye a otros Suscriptores. Es útil cuando se necesita enviar datos a través de una conexión lenta o costosa [6].

Desarrollo de la práctica:

1. Esquema Global Conceptual

En la Figura 3, se presenta el Esquema Global Conceptual a partir del grafo relacional centralizado de la empresa FITEC.

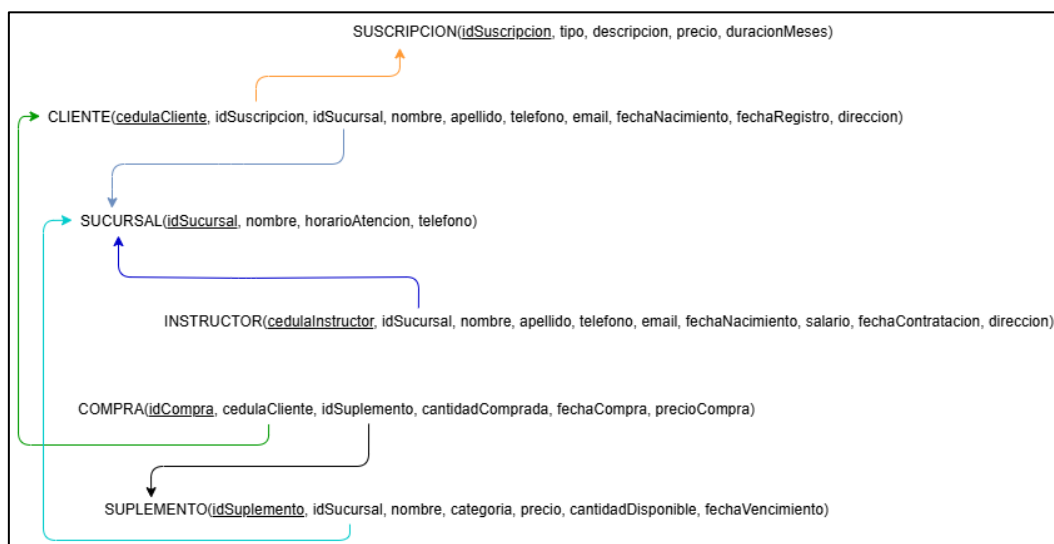


Figura 3. Grafo relacional para la empresa de gimnasios FITEC.

2. Esquema de Replicación

A continuación, se presentan las tablas seleccionadas previamente para su replicación en el sistema distribuido:

Tabla **SUCURSAL**:

- **Justificación:** La tabla se replica porque es referenciada por tablas como **CLIENTE**, **INSTRUCTOR** y **SUPLEMENTO**, específicamente en el campo **idSucursal** para identificar en qué sede se encuentra un cliente, instructor o suplemento. Además, la tabla solo contiene datos informativos que suelen permanecer exactamente iguales durante toda la vida funcional de la sucursal. Por lo tanto, realmente son solo datos de consulta.
- **Nodos involucrados:** QUITO_NORTE, QUITO_SUR
- **Tipo:** Unidireccional
- **Nodo de gestión:** QUITO_SUR



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

Tabla *SUSCRIPCION*:

- **Justificación:** La tabla se replica porque es referenciada por la tabla **CLIENTE** para identificar el tipo de suscripción contratada por cada cliente. Adicionalmente, al contener únicamente tres registros y no presentar cambios frecuentes, se considera una tabla de consulta cuya disponibilidad local en cada sucursal facilitaría el acceso a la información de los planes existentes.
- **Nodos involucrados:** QUITO_NORTE, QUITO_SUR
- **Tipo:** Bidireccional

Tabla *CEDULA_CLIENTE*:

- **Justificación:** La tabla, que en realidad es un fragmento de la tabla **CLIENTE**, a pesar de estar sujeta a registros y modificaciones frecuentes, debe ser replicada para cumplir con el requerimiento de permitir que un cliente realice compras de suplementos en cualquiera de las dos sucursales. De esta manera, se garantiza que las referencias de los clientes estén disponibles en ambas sedes y que la compra se registre en la sede donde se efectuó.
- **Nodos involucrados:** QUITO_NORTE, QUITO_SUR
- **Tipo:** Bidireccional

Como en la fase de diseño se explicó detalladamente la elección del tipo de replicación para cada tabla, esta parte se omitió para dar prioridad a la explicación de la implementación de las replications establecidas.

3. Configuración del Publicador(es)/Distribuidor(es)

a. Tabla *SUCURSAL*

Configuración Distribución en servidor MATEITO

En base al esquema de replicación, primero se realizó la copia de la tabla **SUCURSAL** de la base de datos centralizada **FITEC** en el nodo de gestión **QUITO_SUR** (computador MATEITO) a través del uso de operaciones **SELECT**. Cabe mencionar que no se copió la columna **IDSUCURSAL** como clave primaria y a través de otra query se la tuvo que especificar como primary key.



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

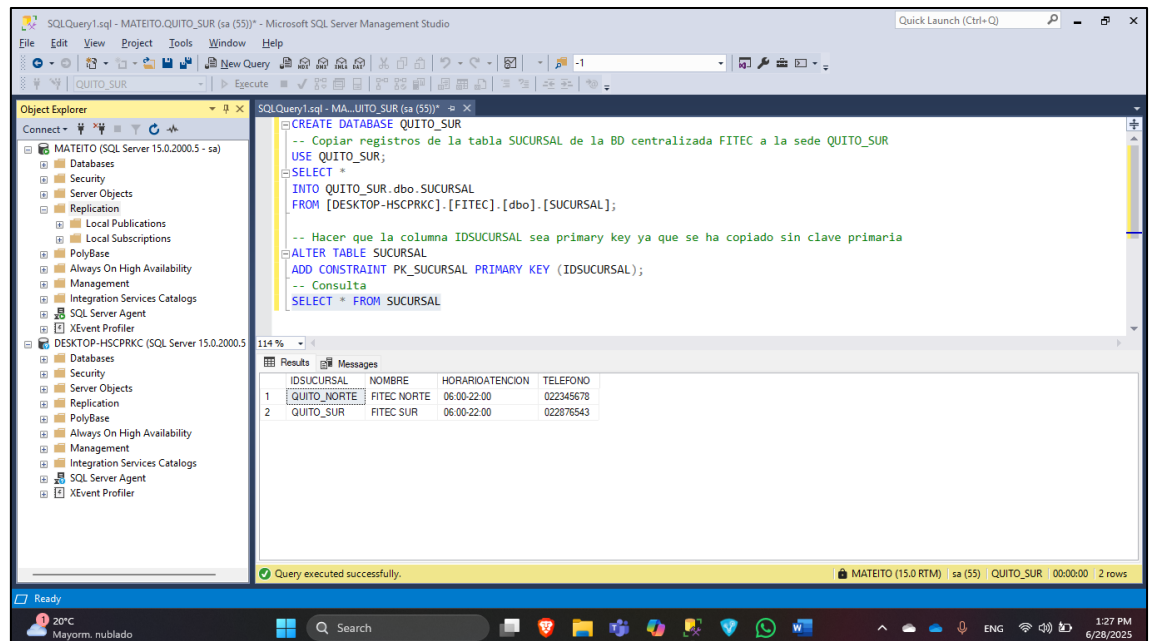


Figura 4. Copia de la tabla **SUCURSAL** de la BD **FITEC** en el nodo **QUITO_SUR**.

Luego, en el Publicador/Distribuidor se creó la carpeta “ReplData” en la ruta del disco Local (C:\).

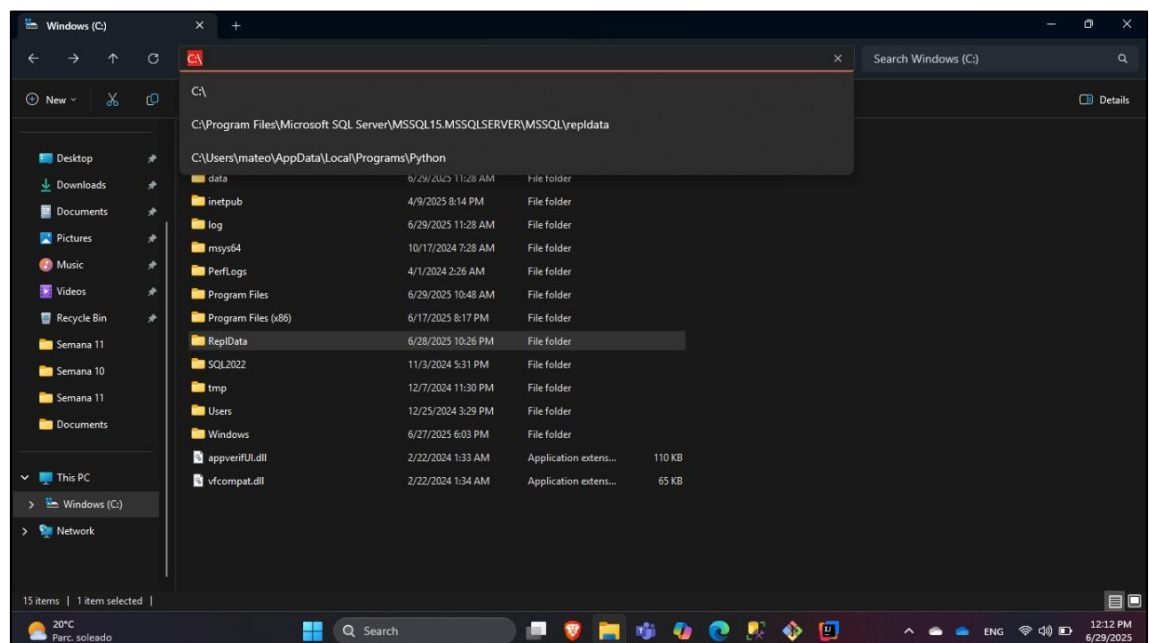


Figura 5. Creación de la carpeta “ReplData” en el directorio del disco local C:\.

Después, en el SSMS del servidor del Publicador/Distribuidor, en la carpeta de “Replication”, se procedió a configurar la distribución usando el Wizard. El primer paso en la configuración de la distribución fue especificar que el mismo



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

servidor Publicador sea su mismo Distribuidor en lugar de especificar otro servidor externo como Distribuidor.

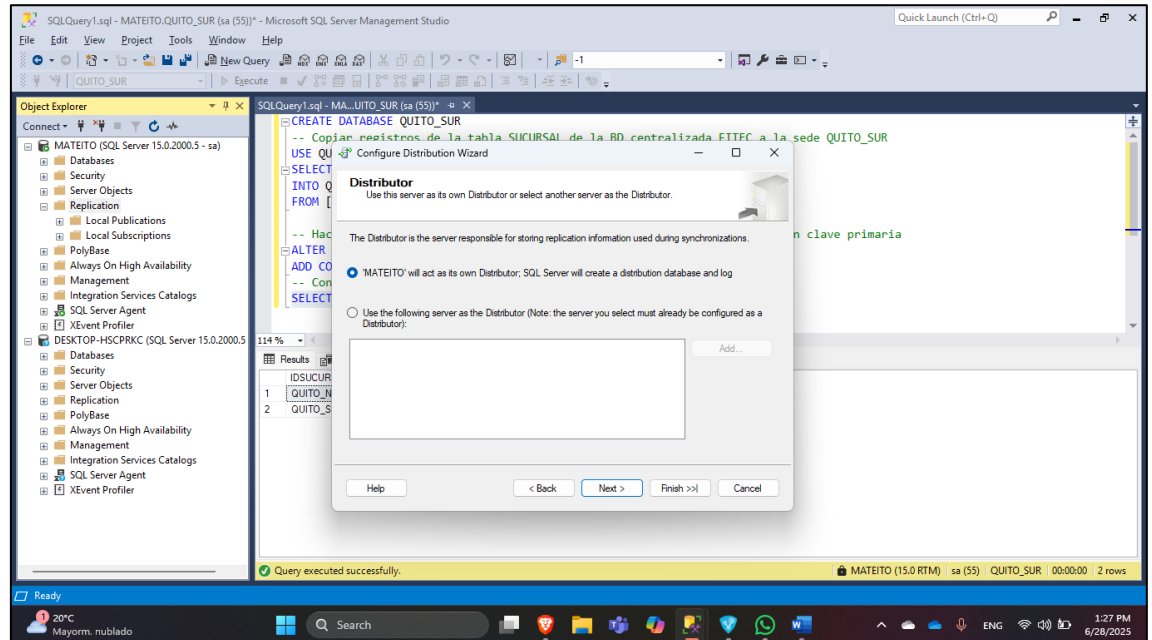


Figura 6. Especificación del servidor Publicador como su mismo Distribuidor.

El siguiente paso fue establecer la ruta de la carpeta de Snapshots (para guardar las instantáneas de nuestra tabla en ese momento del tiempo), la cual fue la ruta de RepIData establecida anteriormente.

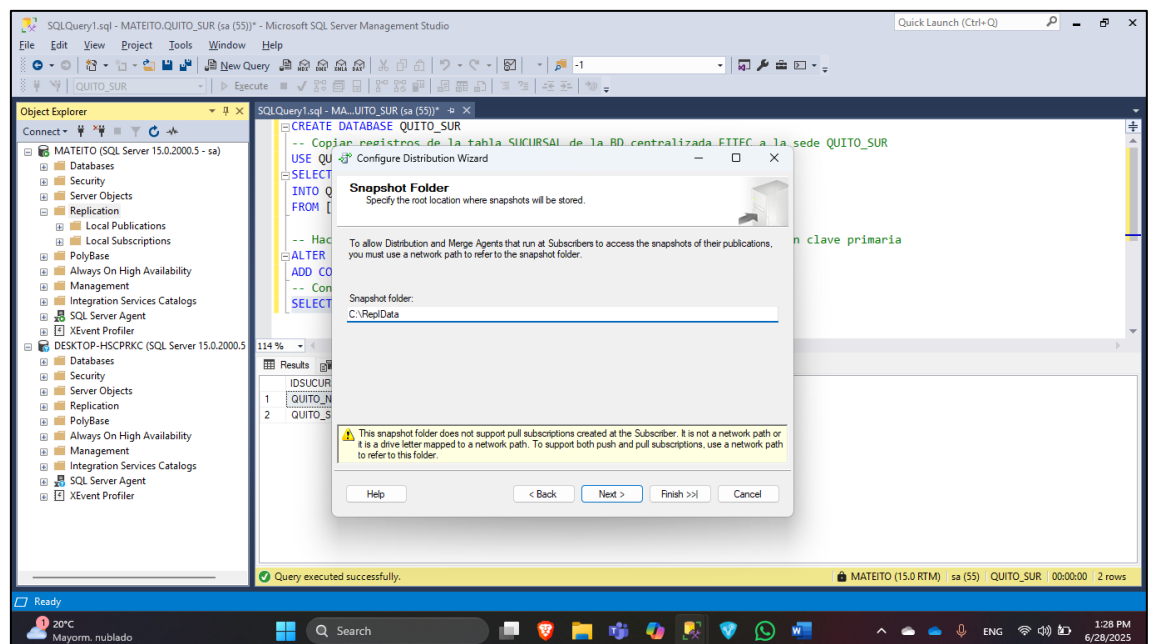


Figura 7. Colocación de la ruta para guardar la carpeta de las Snapshots.



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

Después, se dejó por defecto la configuración del nombre de la BD de Distribución (que lleva los cambios del Publicador al Suscriptor), la ubicación de los archivos de la BD (C:\data) y el registro de las transacciones (C:\log).

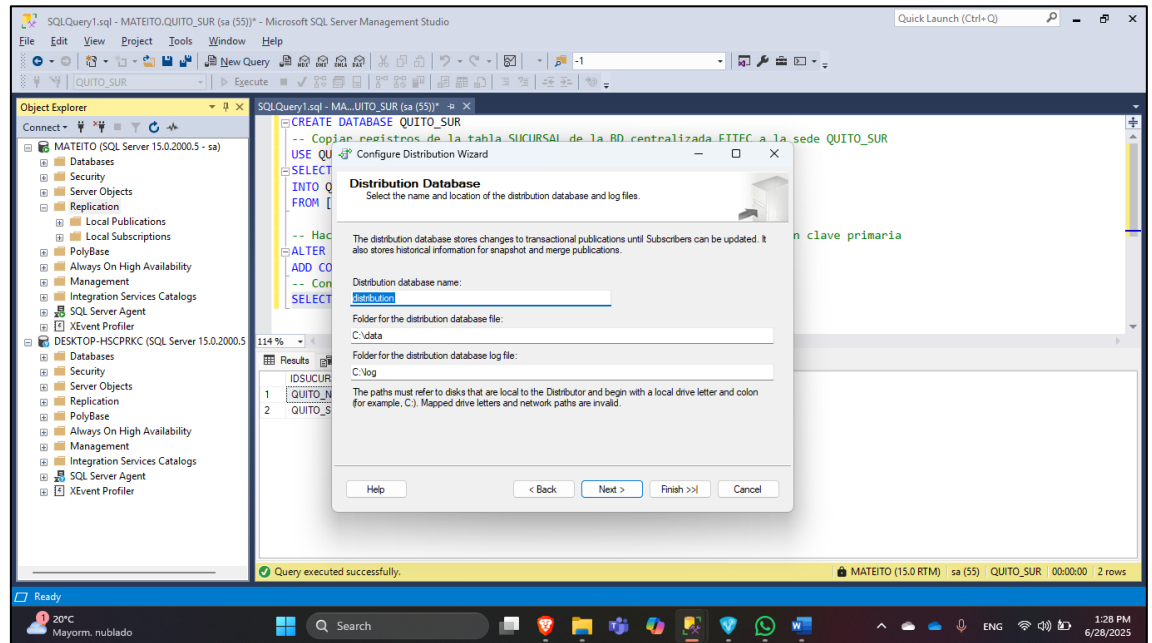


Figura 8. Configuración por defecto del nombre de la base de datos de distribución y la ubicación de los archivos.

Luego de configurar el Distribuidor, se procedió a habilitar el servidor que actuará como Publicador para poder enviar datos a través del Distribuidor. Se escogió el mismo servidor MATEITO para que sea desde donde se repliquen los datos a otros servidores (Suscriptores).



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

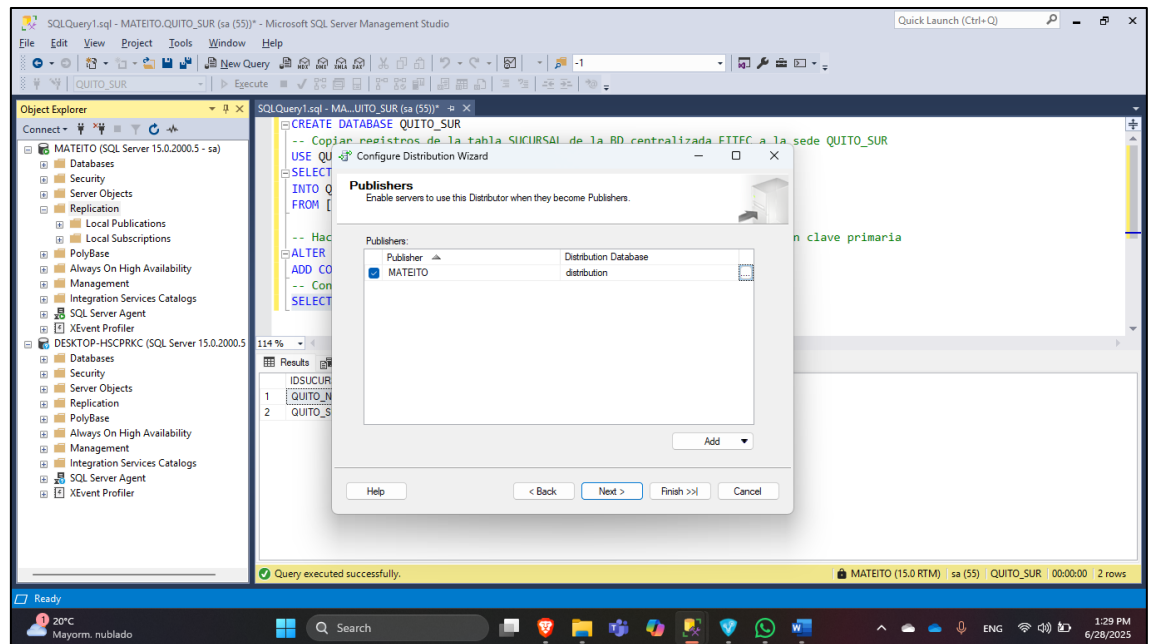


Figura 9. Habilitación del servidor Publicador.

Luego, se escogió la opción de “Configure distribution” para aplicar de forma instantánea la configuración establecida en lugar de generar un script y realizar la ejecución manual.

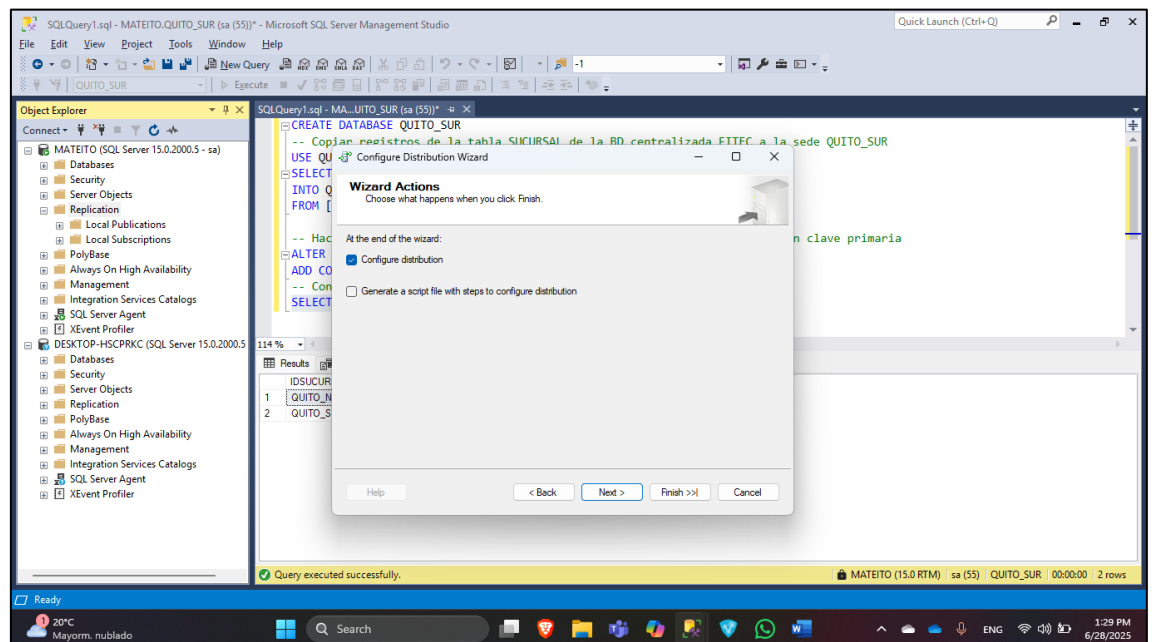


Figura 10. Selección para aplicar de forma automática la configuración de la distribución.

Finalmente, en la pantalla que mostró el resumen de la configuración, se pulsó en el botón “Finish” para confirmar e iniciar la configuración de la distribución.



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

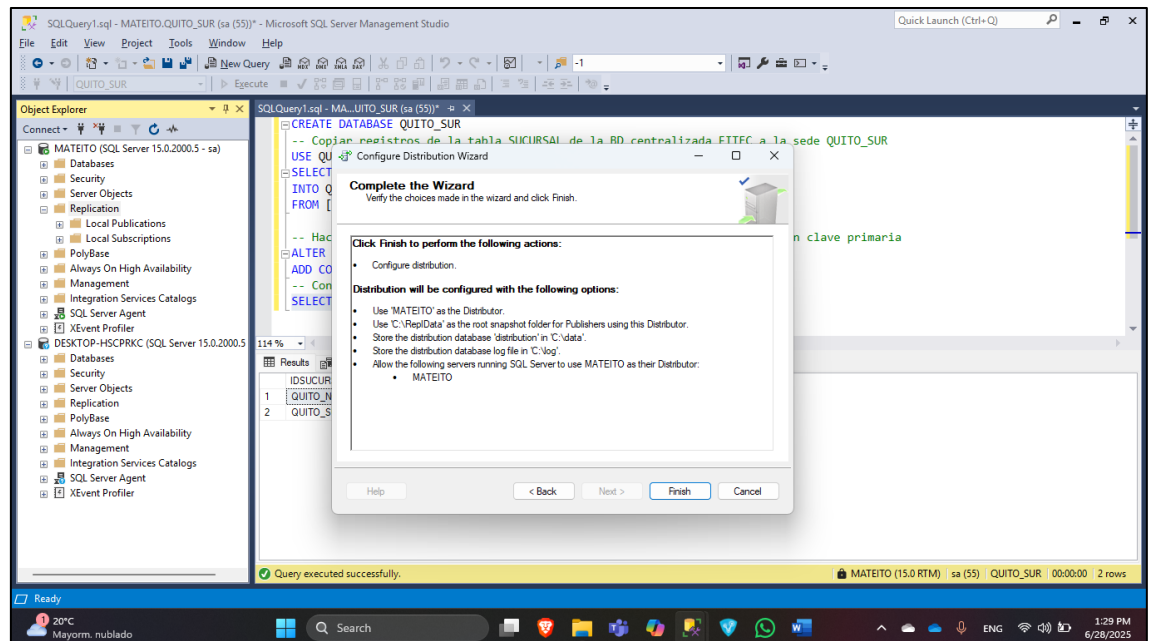
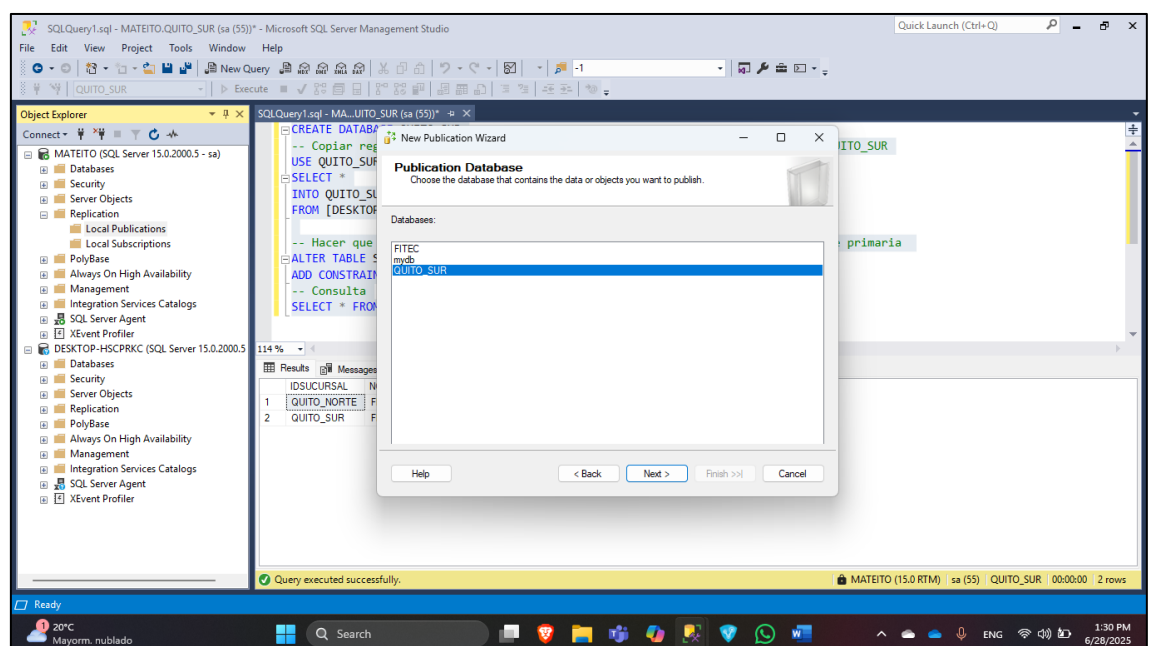


Figura 11. Pantalla con un resumen de la configuración de distribución a ser aplicada.

Creación Publicación *P_SUCURSAL_QUITO_SUR* en servidor MATEITO

El siguiente paso fue crear una publicación en la subcarpeta “Local Publications” de la carpeta Replication. Primero, se especificó la base de datos **QUITO_SUR** que tenía la tabla **SUCURSAL** que se buscó publicar (replicar).





ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

Figura 12. Especificación de la BD que contenga los datos que se quieran publicar.

Luego, según el esquema de replicación para la tabla **SUCURSAL**, al ser de tipo unidireccional, se escogió la opción “Transactional publication”, para que solo la sede **QUITO_SUR** pudiera realizar cambios que se reflejen en todas las réplicas.

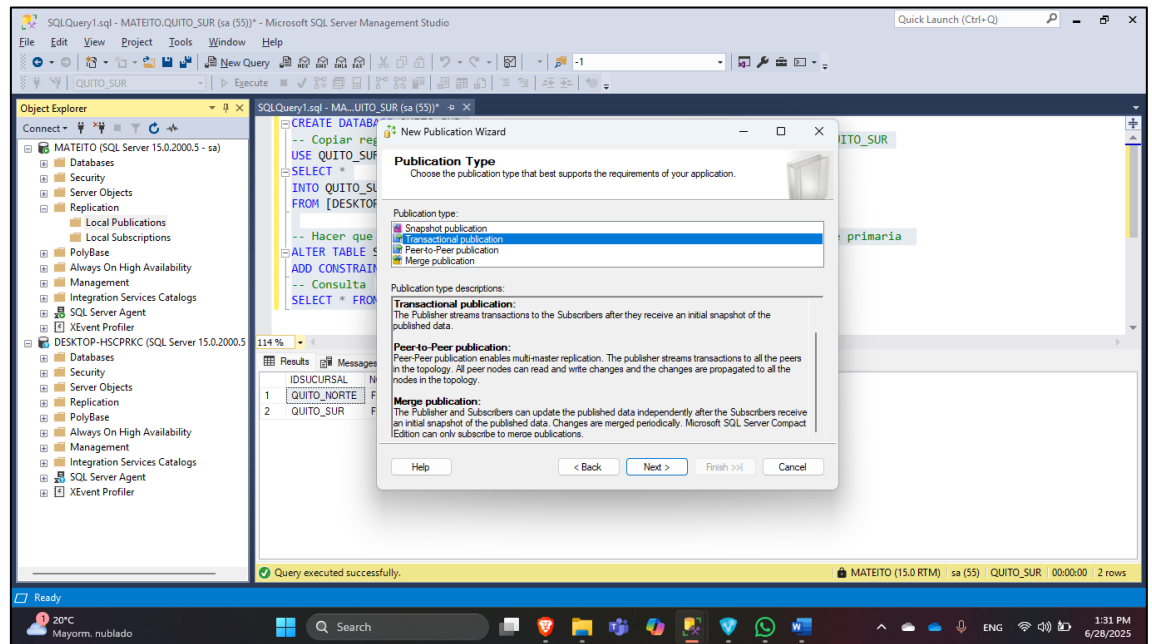


Figura 13. Selección de la replicación de tipo transaccional o unidireccional.

Después, de las tablas existentes en la base de datos **QUITO_SUR**, se escogió la tabla **SUCURSAL**, para así convertirla en un artículo de la publicación.



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

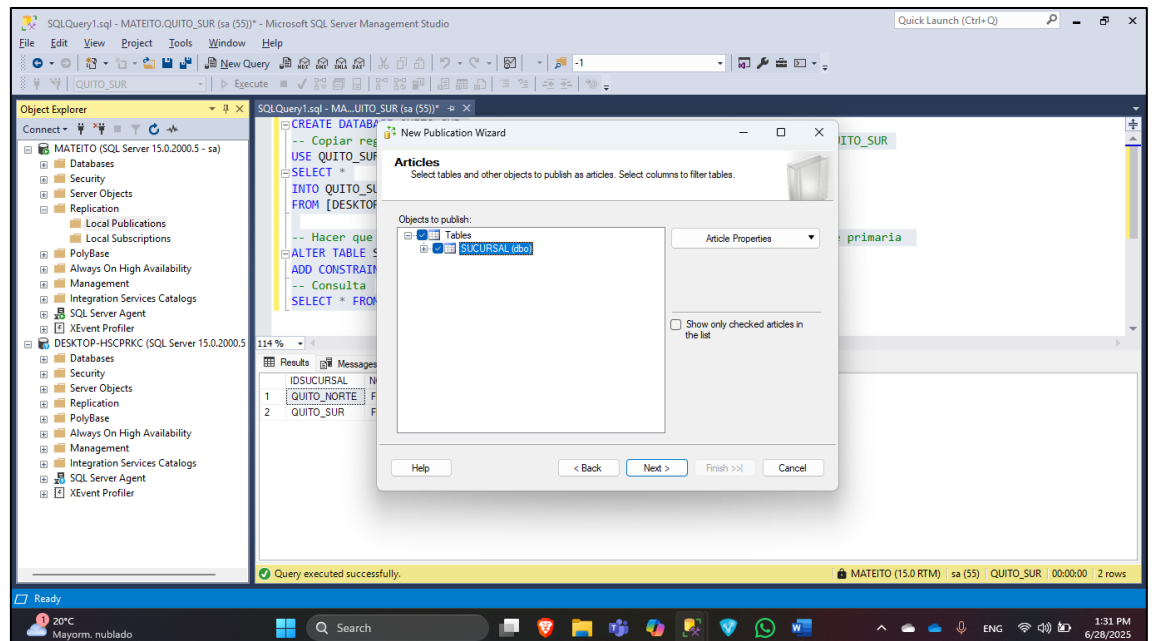


Figura 14. Establecimiento de la tabla **SURCURSALES** como artículo de la publicación.

La siguiente pantalla presentó la opción de filtros para excluir ciertos registros de los artículos escogidos. Sin embargo, no se configuró nada en esta sección y se pasó a la siguiente pantalla.

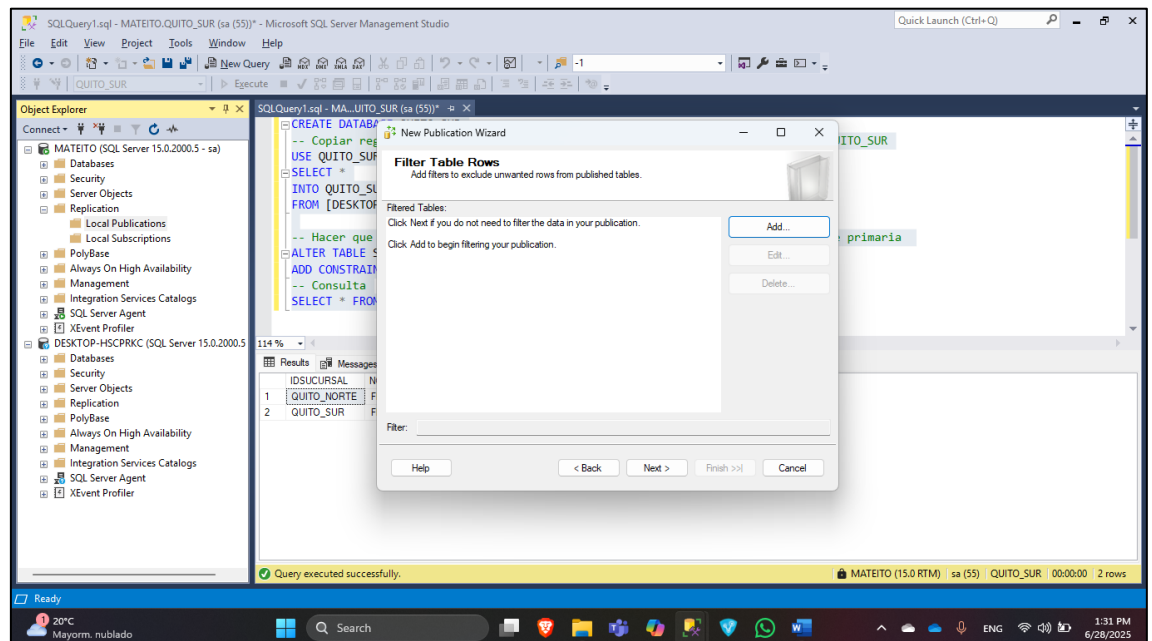


Figura 15. Pantalla para filtrado de registros configurada sin modificaciones.

Luego, para la creación del Snapshot inicial de la publicación, se escogió la primera opción para crear un Snapshot inmediatamente con los datos actuales.



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

De esta manera, ya se aseguró tener listo un Snapshot cuando llegue la configuración de la suscripción.

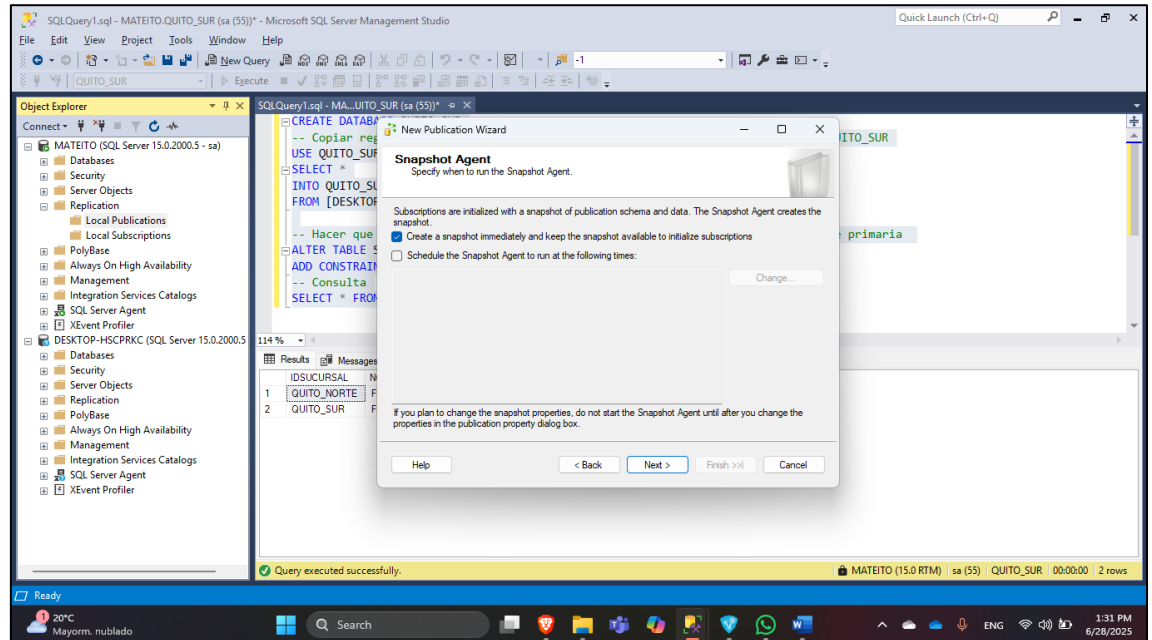


Figura 16. Configuración de la creación inmediata de un Snapshot habilitado para las Suscripciones.

Después, en “Security Settings” se escogió la segunda opción para que el Snapshot Agent se ejecute con la misma cuenta que el servicio SQL Server Agent del servidor. De esta manera, el Snapshot Agent tuvo los privilegios necesarios para acceder al Publicador, sacar Snapshots de los datos y moverse libremente entre los servidores para entregar la información.

Además, para la conexión con este Publicador, se especificó que se lo haría si se usaban las credenciales “sa” y “P@ssw0rd” del login que se había establecido en el SQL Server.



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

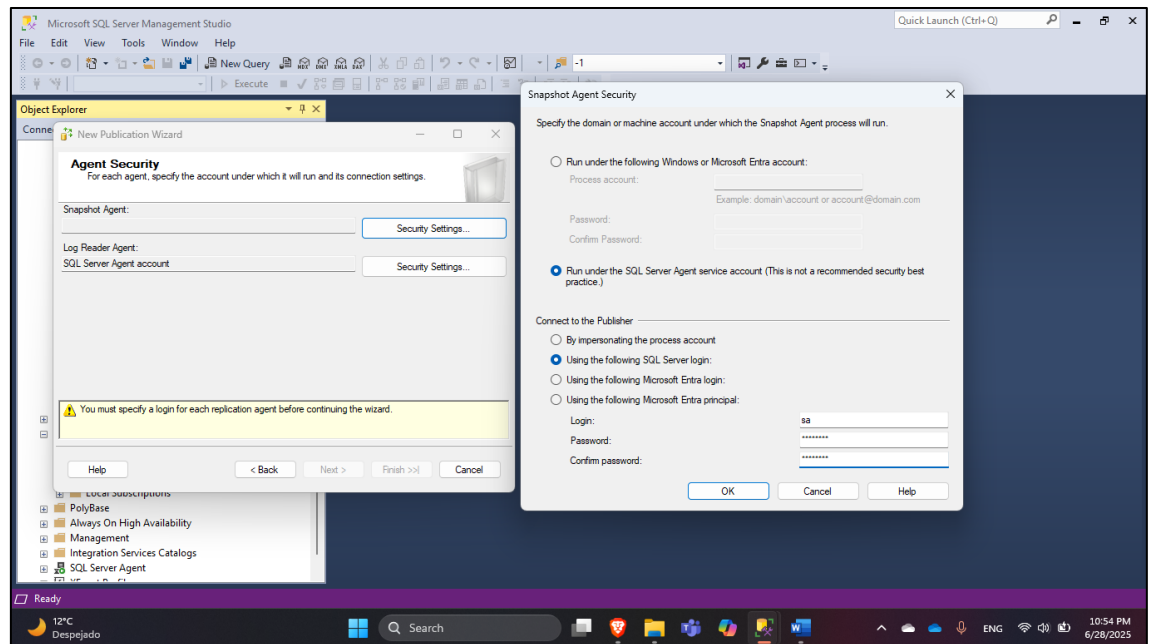


Figura 17. Configuración de la seguridad del Snapshot Agent.

Luego, se escogió la primera opción para que la publicación se cree automáticamente en base a las configuraciones establecidas.

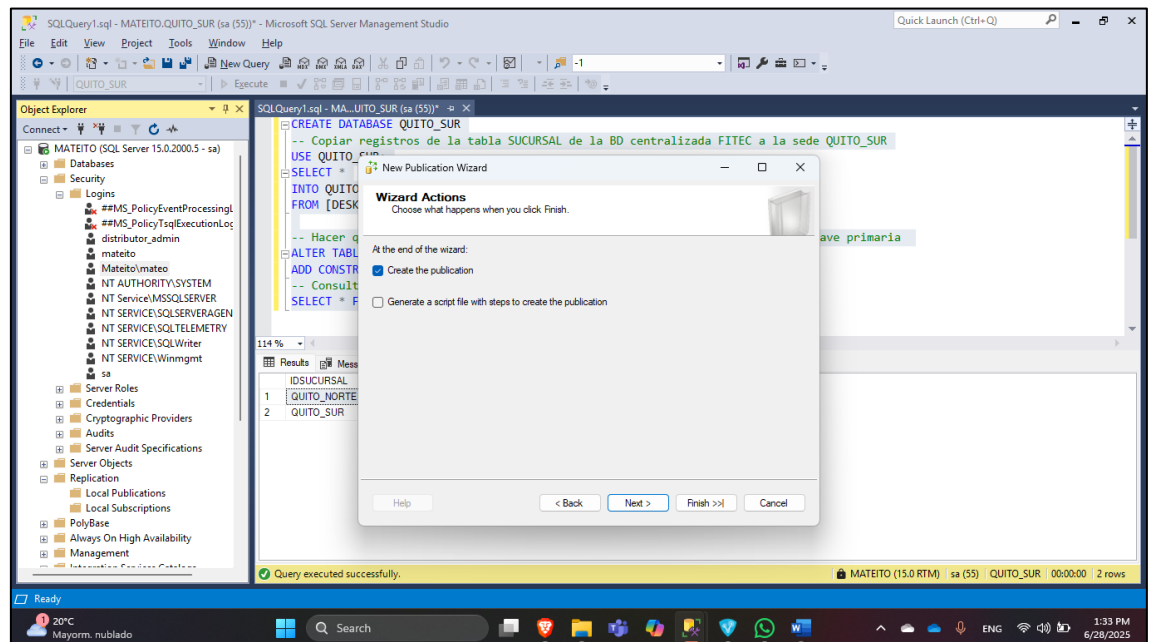


Figura 18. Selección de la manera de crear y configurar la publicación.

Después se especificó un nombre para la publicación junto con un resumen de su configuración.



ESCUELA POLITÉCNICA NACIONAL
FACULTAD DE INGENIERÍA DE SISTEMAS
INGENIERÍA DE SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN

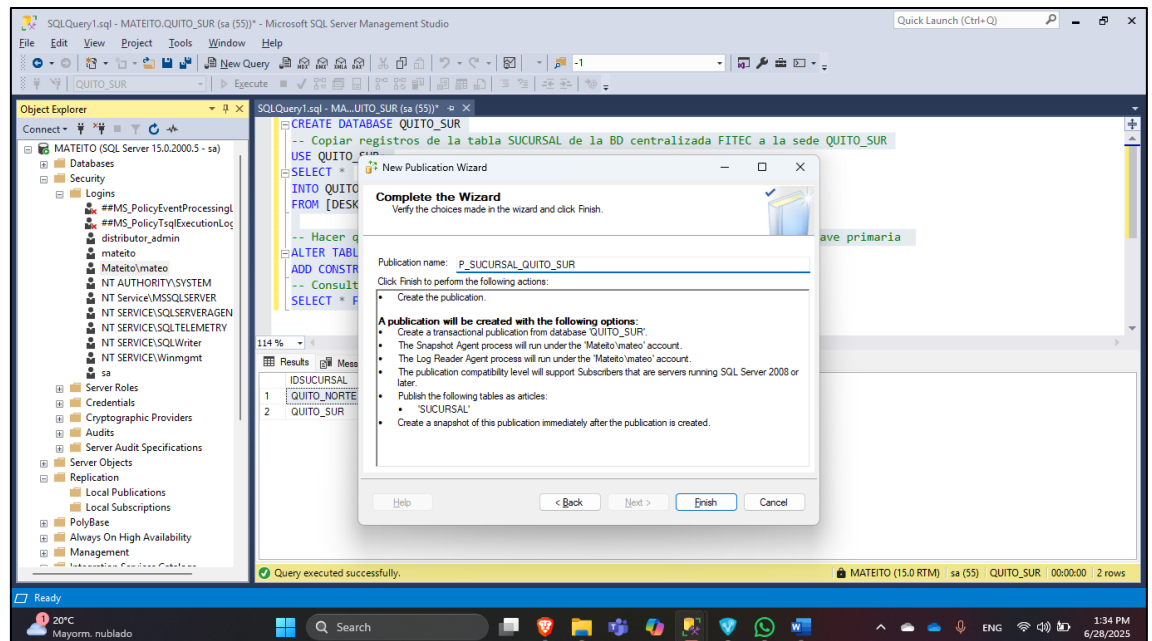


Figura 19. Establecimiento del nombre de la publicación junto con el resumen de su configuración.

La siguiente pantalla mostró el resultado de la creación de la publicación. Aparentemente todo pareció ejecutarse correctamente, sin embargo, **esta no es la evidencia real de la correcta creación de la publicación.**

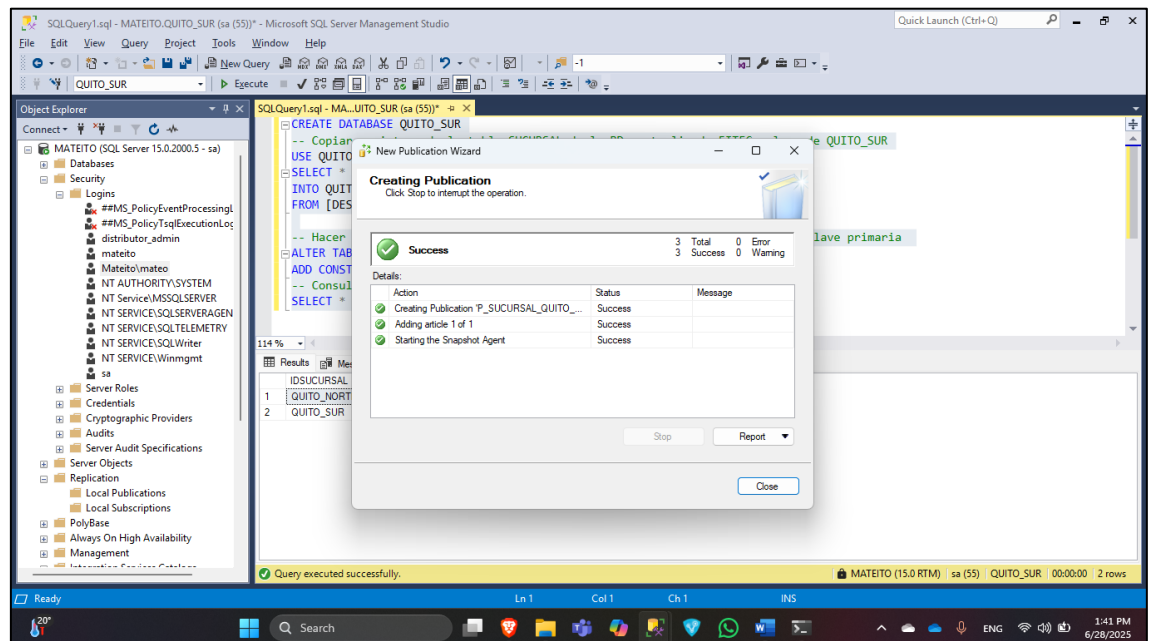


Figura 20. Pantalla de la supuesta “exitosa” creación de la publicación.

Luego de obtener un error en el monitor de replicación, se procedió a colocar en el path las siguientes variables de entorno del sistema