

**UNIVERSIDAD PERUANA LOS ANDES**  
**FACULTAD INGENIERIA INGENIERÍA DE**  
**SISTEMAS Y COMPUTACIÓN**



**MANUAL DE AUTOMATIZACIÓN Y  
MANTENIMIENTO**

**ASIGNATURA: BASE DE DATOS II**

**DOCENTE: FERNÁNDEZ BEJARANO RAUL**

**ESTUDIANTE: Bonifacio Hilario Erick**

**CÓDIGO: S01238F**

**HUANCAYO-2025**

**Objetivo:**

Administrar tareas repetitivas y asegurar la integridad del sistema. Esta semana se abordarán las herramientas y técnicas necesarias para automatizar tareas, optimizar el rendimiento y garantizar que el sistema se mantenga operando correctamente a largo plazo, sin intervención manual constante.

## **1. Creación de Jobs y Alerts con SQL Server Agent**

### **Introducción:**

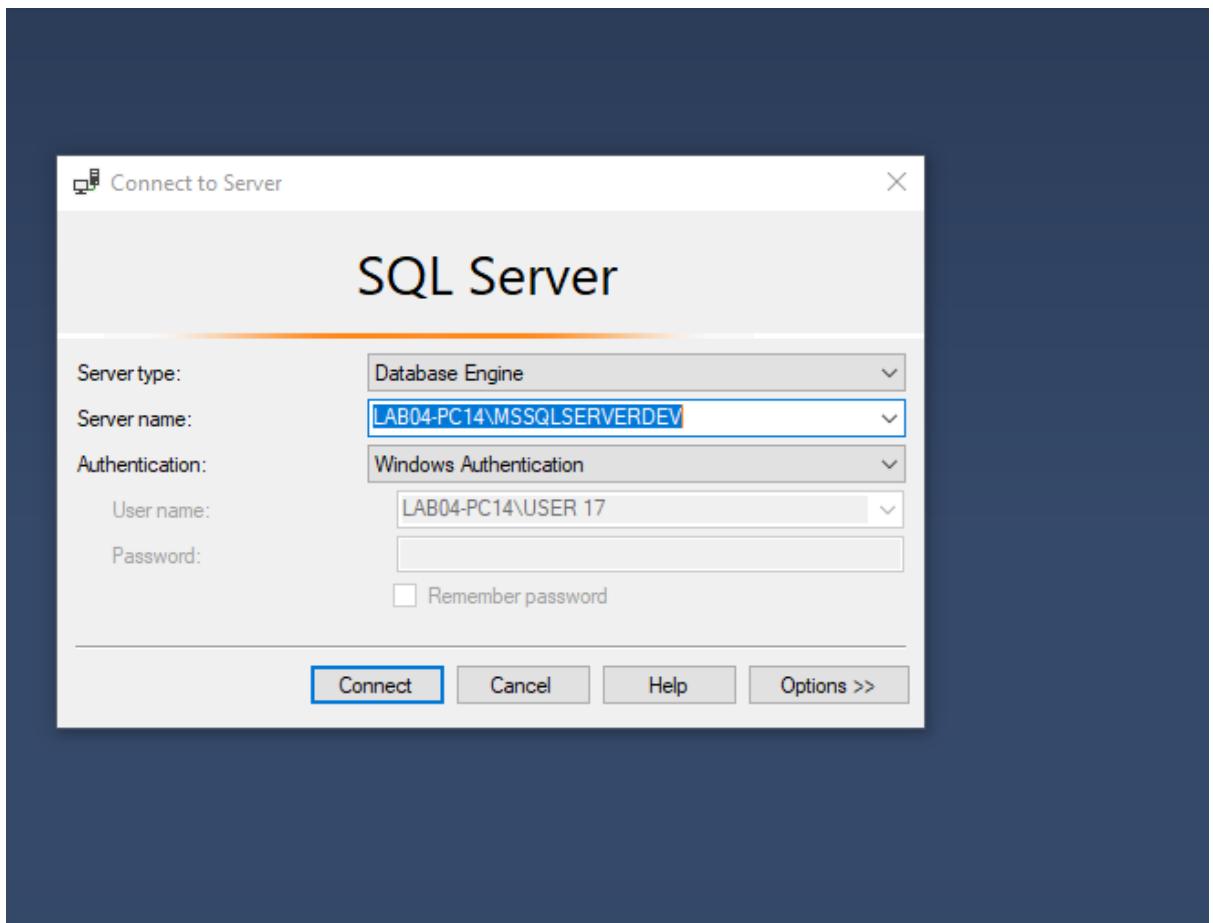
SQL Server Agent es una herramienta que permite automatizar la ejecución de tareas, como la ejecución de consultas T-SQL, copias de seguridad, y tareas programadas. La automatización de estos trabajos reduce la carga administrativa y garantiza que las tareas se ejecuten sin intervención manual.

### **Explicación:**

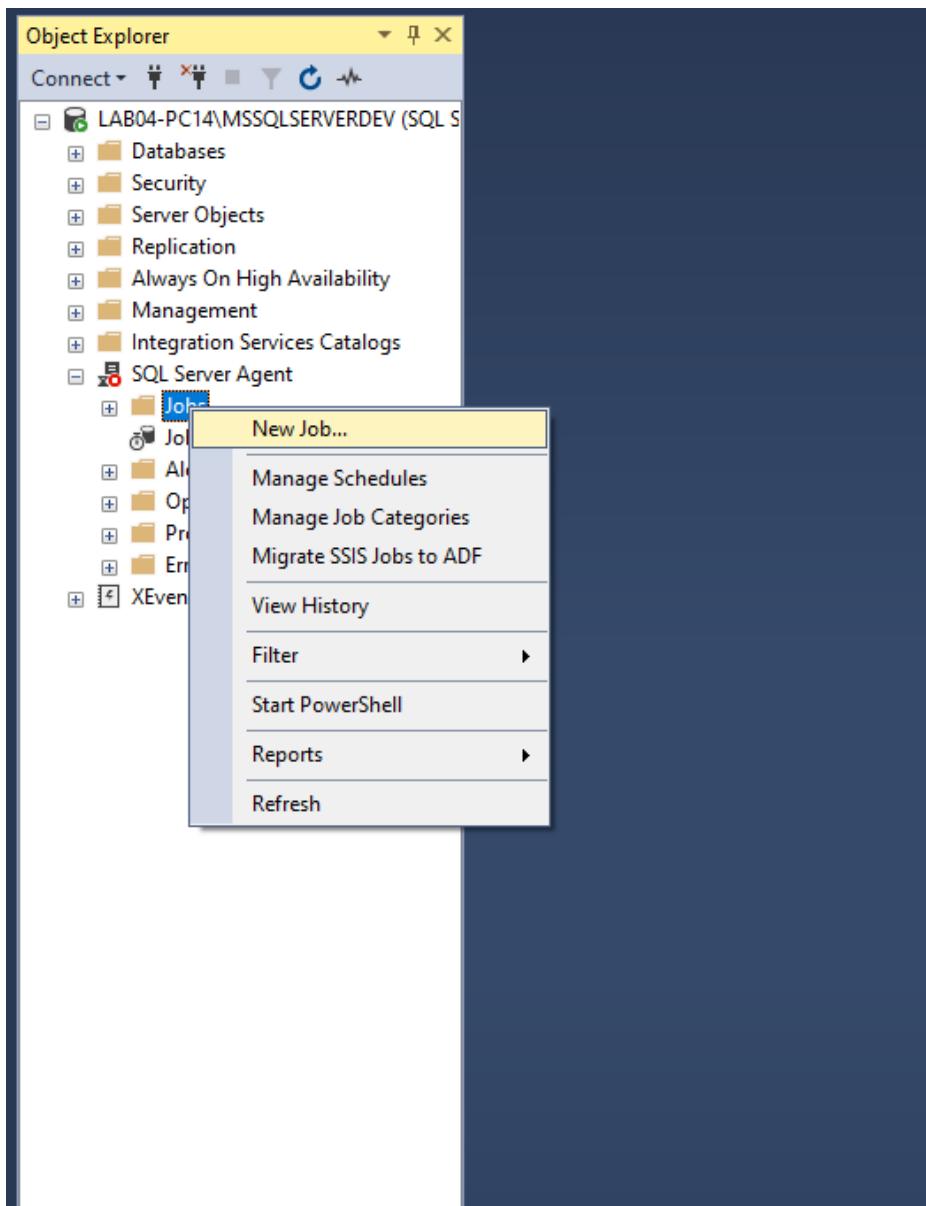
- **Jobs:** Son colecciones de uno o más pasos que se ejecutan en un orden determinado. Estos pasos pueden incluir tareas como respaldos, mantenimiento, ejecución de scripts, entre otros.
- **Alerts:** Son notificaciones configuradas para dispararse cuando ocurren ciertos eventos, como errores críticos, tiempos de espera largos o problemas de rendimiento.

### **Pasos para crear un Job:**

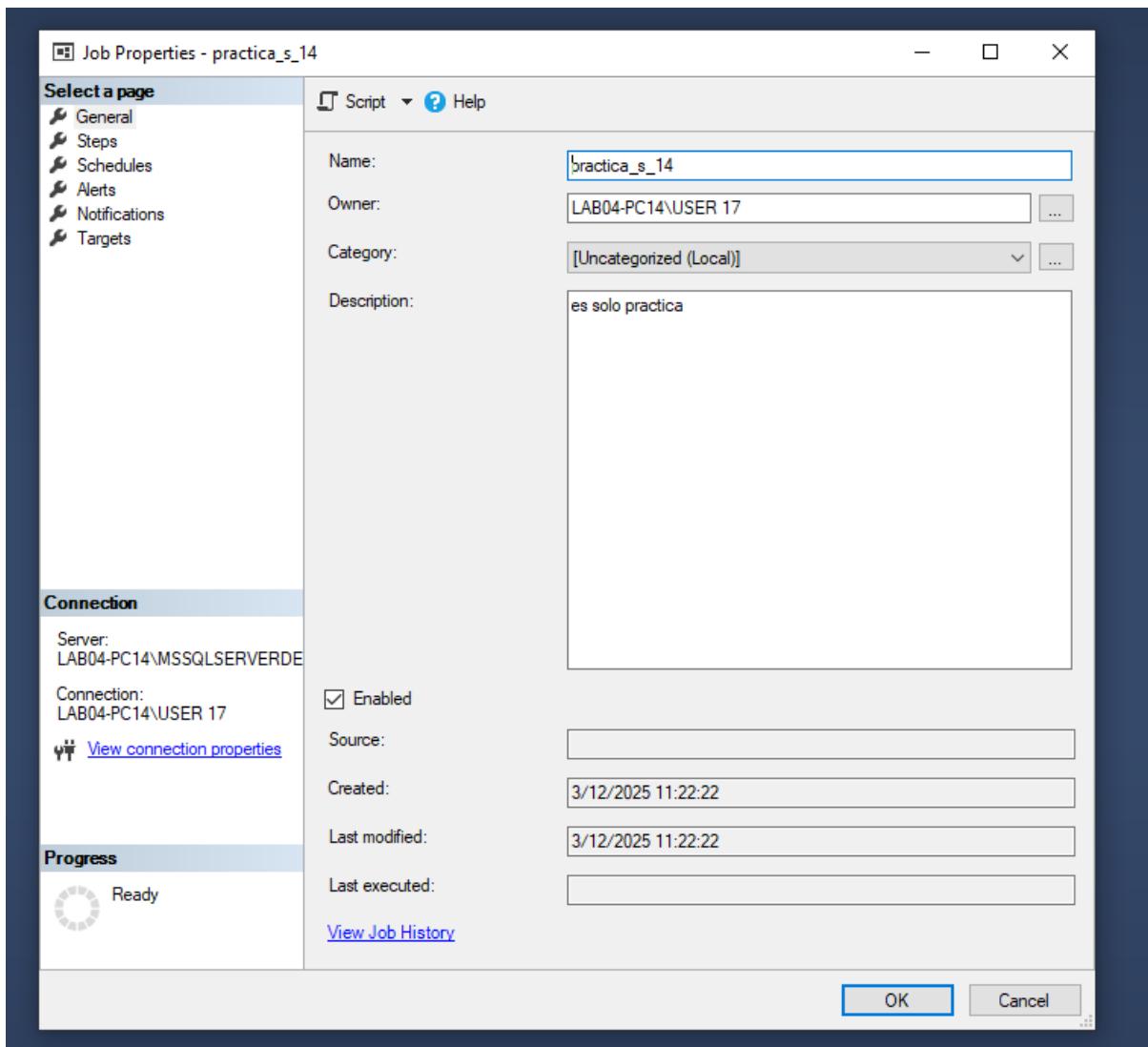
1. Abrir SQL Server Management Studio (SSMS).



2. Conectarse al servidor y navegar hasta **SQL Server Agent**.
3. Clic derecho en **Jobs > New Job**.

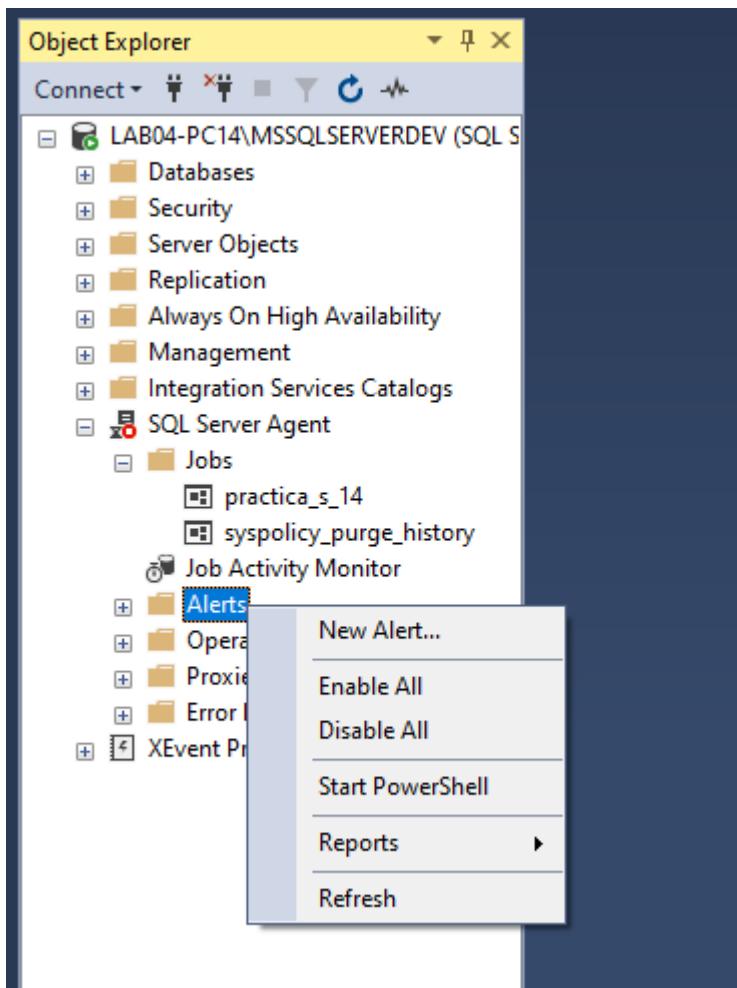


4. Configurar los detalles del Job: nombre, propietario, descripción.
5. Agregar pasos que definan qué acción realizar.
6. Establecer el horario para que el Job se ejecute automáticamente.

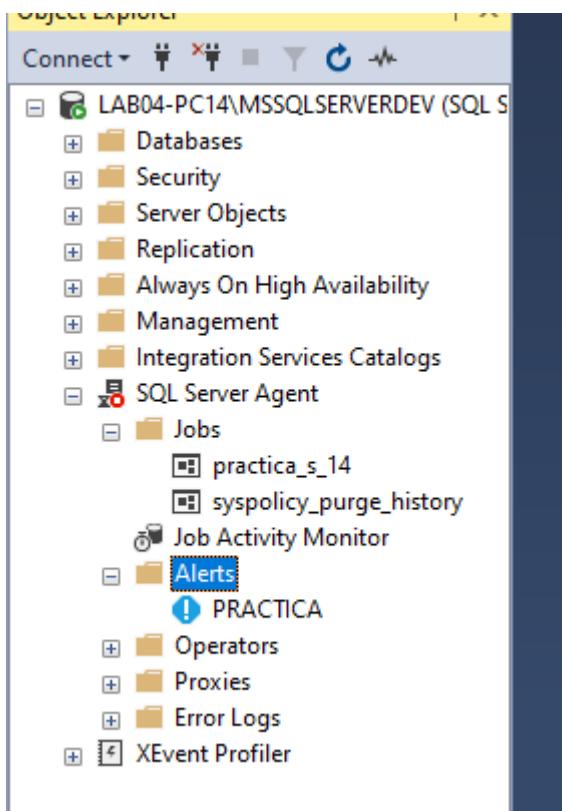


### Pasos para crear un Alert:

1. En SQL Server Agent, clic derecho en **Alerts** > **New Alert**.



2. Configurar el evento o condición que disparará la alerta (por ejemplo, error 823).
3. Establecer las acciones a realizar cuando la alerta se active, como enviar un correo electrónico.



## 2. Uso de Planes de Mantenimiento (Database Maintenance Plans)

### Introducción:

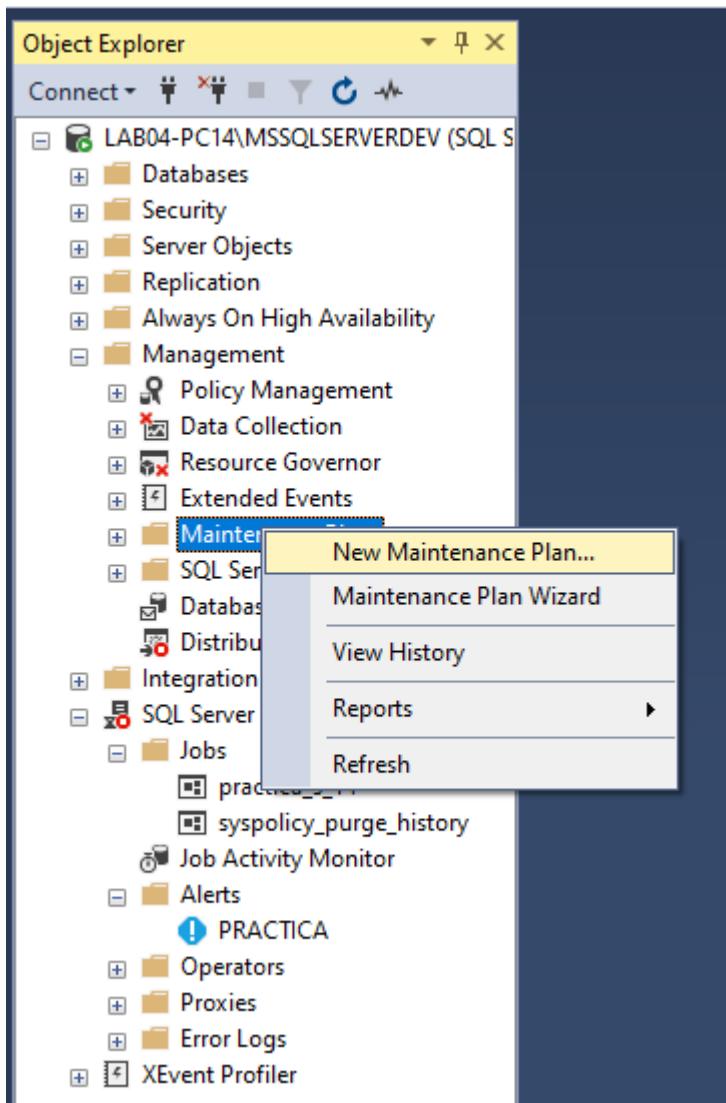
Los planes de mantenimiento permiten definir una serie de tareas que se ejecutan de forma automática en la base de datos, garantizando su salud y rendimiento. Esto incluye tareas como la reindexación, limpieza de registros, y respaldos.

### Explicación:

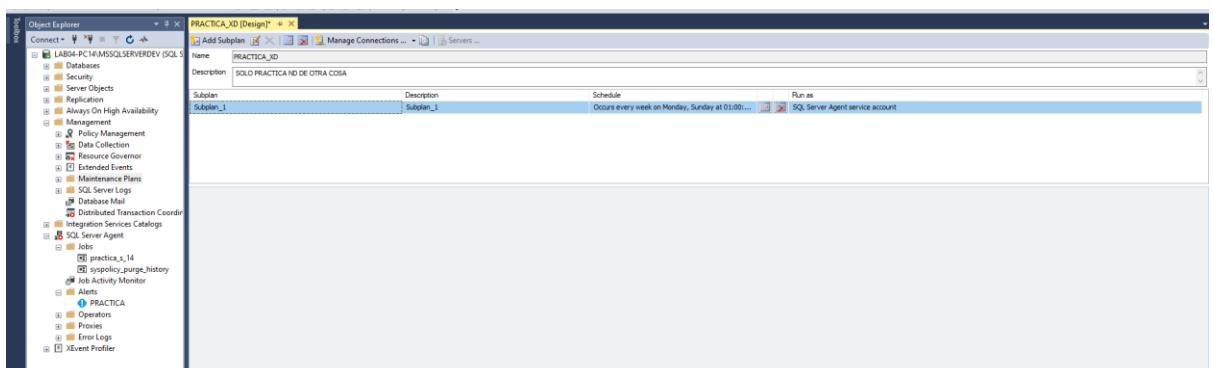
- **Database Maintenance Plans** son plantillas o secuencias de tareas predefinidas, que pueden incluir actividades como:
  - Copia de seguridad de bases de datos.
  - Verificación de la integridad de bases de datos.
  - Limpieza de registros.
  - Reorganización de índices.
- Pueden ser programados para ejecutarse automáticamente a intervalos específicos.

### Pasos para crear un Plan de Mantenimiento:

1. Abrir SSMS.
2. Navegar a **Management > Maintenance Plans**.



3. Clic derecho y seleccionar **New Maintenance Plan**.
4. Usar el diseñador gráfico para agregar tareas, como respaldos o reindexación.
5. Establecer el horario y configurar notificaciones.



### 3. Automatización con T-SQL y PowerShell

### **Introducción:**

La automatización de tareas con **T-SQL** y **PowerShell** permite realizar tareas complejas de manera programática. Estas herramientas son útiles para scripts personalizados que no pueden ser cubiertos por Jobs o Planes de Mantenimiento predeterminados.

### **Explicación:**

- **T-SQL:** Se puede utilizar para escribir scripts complejos que se ejecuten de manera programada, como limpiar datos obsoletos, ejecutar consultas específicas o realizar mantenimiento de índices.

### **Ejemplo de T-SQL para automatización:**

```
BACKUP DATABASE MiBaseDeDatos TO DISK =  
'C:\Backup\MiBaseDeDatos.bak';
```

SQLQuery1.sql - LA...PC14\USER 17 (52)\* ↗ X

```

CREATE DATABASE MiBaseDeDatos
BACKUP DATABASE MiBaseDeDatos TO DISK = 'C:\Backup\MiBaseDeDatos.bak';

```

100 %

Messages

```

Processed 368 pages for database 'MiBaseDeDatos', file 'MiBaseDeDatos' on file 1.
Processed 1 pages for database 'MiBaseDeDatos', file 'MiBaseDeDatos_log' on file 1.
BACKUP DATABASE successfully processed 369 pages in 0.018 seconds (159.803 MB/sec).

Completion time: 2025-12-03T11:35:45.5216625-05:00

```

Backup

Archivo Inicio Compartir Vista

Este equipo > Windows (C:) > Backup

	Nombre	Fecha de modificación	Tipo	Tamaño
Acceso rápido				
Escritorio				
Descargas				
Documentos				
Imágenes				
Música				
Sw2023ksz				
Videos				
Windows (C:)				
OneDrive				
Este equipo				
Descargas				
Documentos				
Escritorio				
Imágenes				

- **PowerShell:** Permite automatizar tareas de administración de SQL Server a través de scripts. Se puede usar para realizar operaciones como copias de seguridad, ejecución de consultas, manejo de archivos de registro, entre otros.

## **4. Monitoreo Proactivo (Correo, Logs, Alertas de Rendimiento)**

### **Introducción:**

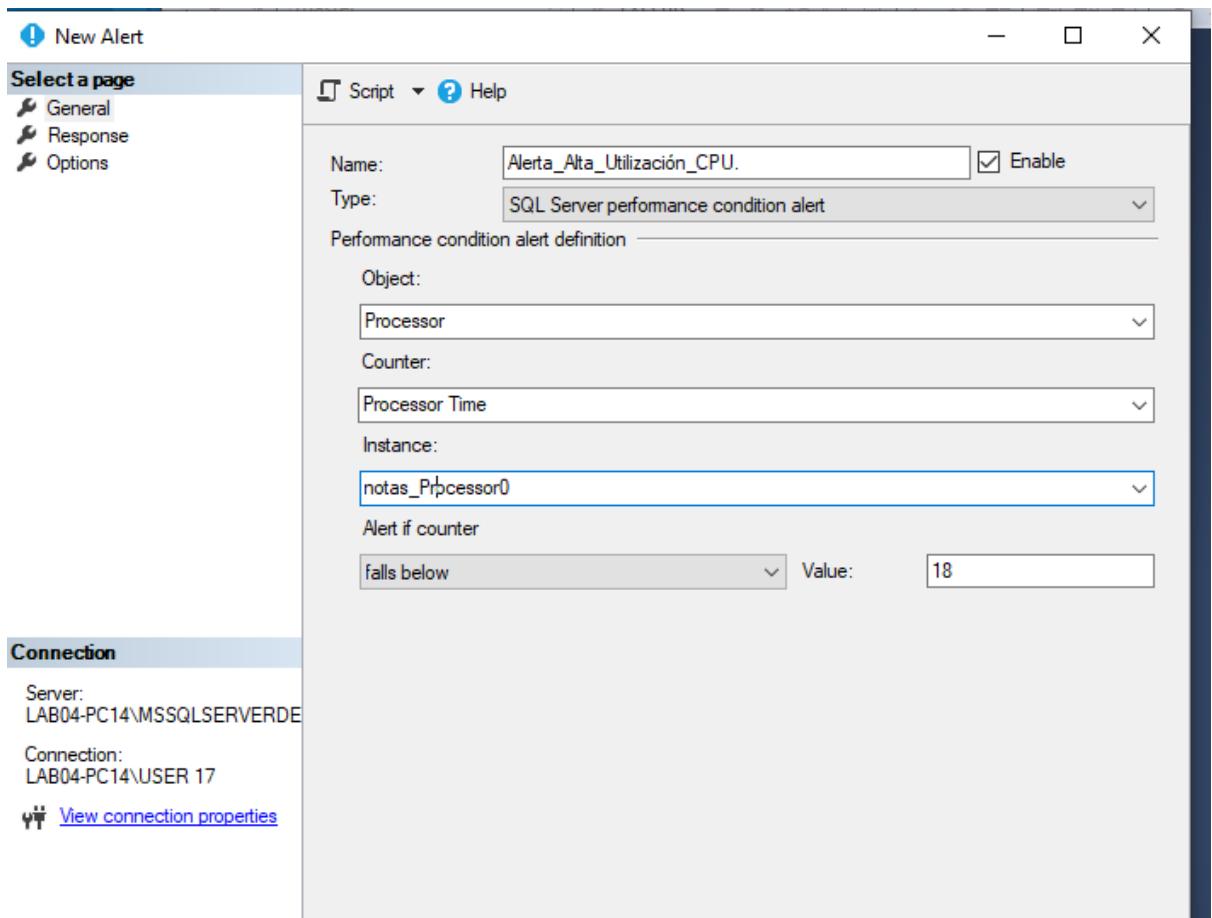
El monitoreo proactivo es fundamental para detectar problemas antes de que se conviertan en fallos graves. A través de correos electrónicos, logs y alertas de rendimiento, es posible mantener el sistema funcionando de manera óptima y evitar tiempos de inactividad no planificados.

### **Explicación:**

- **Correo electrónico:** Configurar alertas por correo cuando ciertos eventos ocurren, como errores en la base de datos o cuando los trabajos no se ejecutan correctamente.
- **Logs:** Utilizar los logs de SQL Server para obtener información detallada sobre el rendimiento, errores, y eventos que ocurren en el servidor.
- **Alertas de rendimiento:** SQL Server permite crear alertas sobre el uso de recursos, como CPU, memoria y espacio en disco. Estas alertas pueden generar una notificación cuando se alcanzan umbrales específicos.

### **Pasos para configurar alertas de rendimiento:**

1. Crear una alerta de rendimiento en SQL Server Agent.}



2. Definir los umbrales para los recursos que se desean monitorear.
3. Configurar las notificaciones por correo para recibir alertas cuando los umbrales sean alcanzados.

## Prácticas:

### 1. Configuración de tareas de limpieza de logs antiguos

1. Crear un Job que se ejecute regularmente para limpiar los logs de transacciones antiguos que ya no son necesarios.

The screenshot shows the SSMS interface with the following details:

- Object Explorer:** Shows the database structure under "LAB04-PC14\MSSQLSERVERDEV (SQL Server Database)".
- SQLQuery2.sql - LA...PC14\USER 17 (51)\*:** The current query window.
- Query Text:**

```
-- Eliminar registros de transacciones que tengan más de 30 días
DELETE FROM Customers
WHERE DATEDIFF(DAY, RegistrationDate, GETDATE()) > 30;
GO
```
- Messages:** The results pane shows:
  - (0 rows affected)
  - Completion time: 2025-12-03T11:51:33.8051837-05:00

2. Configurar el Job para eliminar registros de transacciones que tengan más de X días.

```
EXEC sp_send_dbmail

    @profile_name = 'TuPerfilCorreo',
    @recipients = 'correo@dominio.com',
    @subject = 'Reporte Semanal de Clientes',
    @body = 'Adjunto el reporte de clientes registrados.',
    @query = 'SELECT CustomerID, FirstName, LastName, Email,
    RegistrationDate
        FROM Customers
        ORDER BY RegistrationDate DESC;',
    @attach_query_result_as_file = 1;

GO
```

```
-- Ejecutar el reporte de clientes y enviarlo por correo electrónico
EXEC sp_send_dbmail
    @profile_name = 'TuPerfilCorreo',
    @recipients = 'correo@dominio.com',
    @subject = 'Reporte Semanal de Clientes',
    @body = 'Adjunto el reporte de clientes registrados.',
    @query = 'SELECT CustomerID, FirstName, LastName, Email, RegistrationDate
        FROM Customers
        ORDER BY RegistrationDate DESC;',
    @attach_query_result_as_file = 1;
GO
```

## 2. Crear un Job para ejecutar reportes semanales

1. Escribir un script en T-SQL que genere los reportes necesarios.

```
EXEC msdb.dbo.sp_send_dbmail

    @profile_name = 'MiPerfilCorreo', -- ← Cambia por tu perfil de Database Mail

    @recipients = 'destinatario@correo.com', -- ← Cambia por tu correo

    @subject = 'Reporte Semanal de Clientes',

    @body = 'Adjunto encontrará el reporte semanal de clientes registrados.',

    @query = '

        SELECT

            CustomerID,
            FirstName,
            LastName,
            Email,
            RegistrationDate
        FROM TestDB.dbo.Customers
        ORDER BY RegistrationDate DESC;
    ',


    @attach_query_result_as_file = 1,
    @query_result_separator = ',',
    @query_result_no_padding = 1,
    @query_attachment_filename = 'Reporte_Semanal.csv';
```

2. Crear un Job que ejecute este script semanalmente y lo envíe por correo electrónico a los usuarios correspondientes.