**Universidad Peruana Los Andes Facultad de Ingeniería**

**Escuela profesional de Ingeniería de Sistemas**

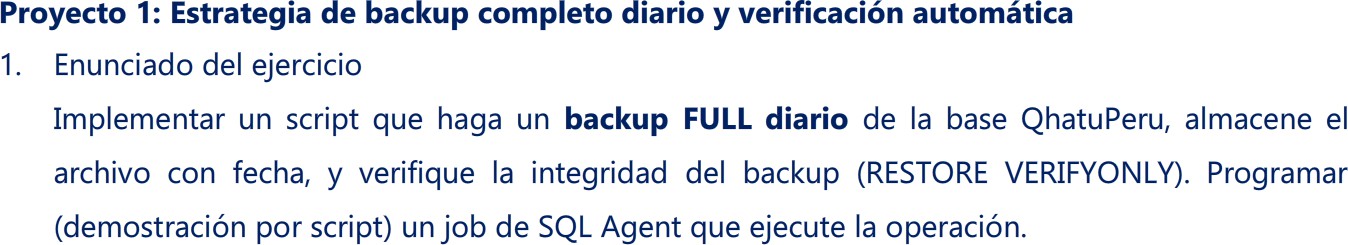
****

**Curso: Base de datos II**

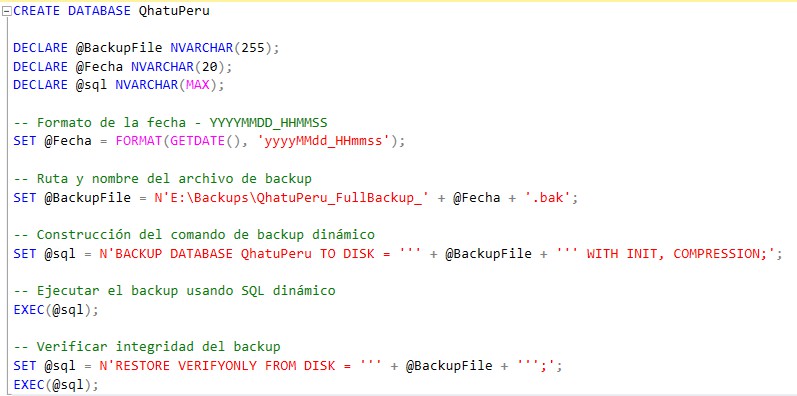
**Docente: Raul Enrique Fernandez Bejarano**

**Estudiante: CONDOR HUAMAN ERICK JHOSHUA**

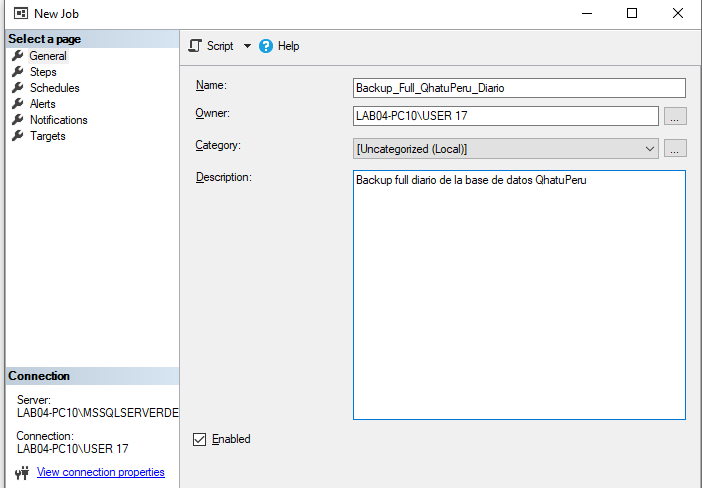
**EJERCICIOS PRACTICOS DE LA SEMANA 12**

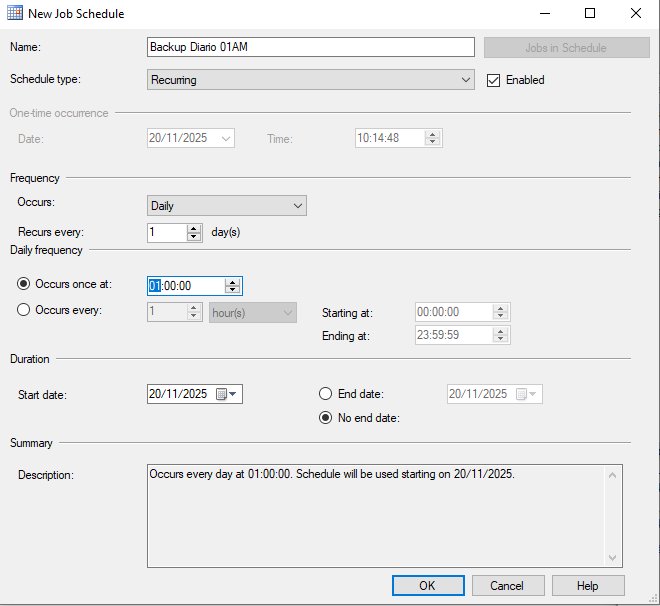
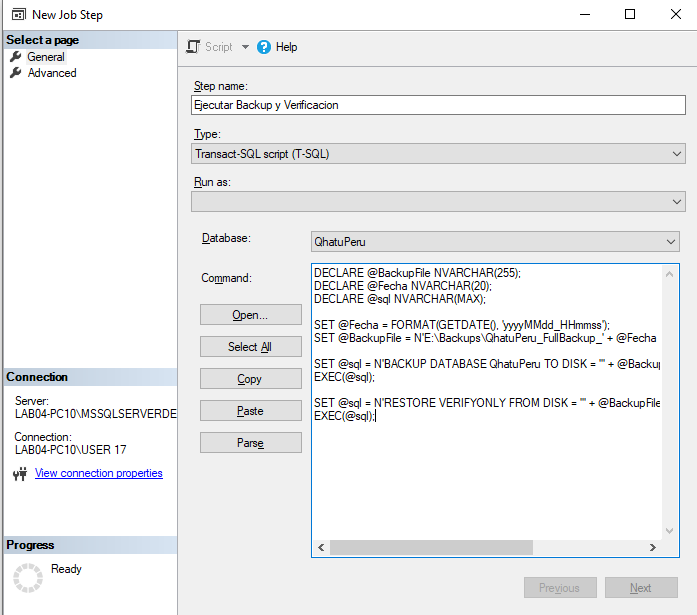
****

**SCRIPT DE LA SOLUCION**

Para el Backup diario de la base de datos QhatuPeru

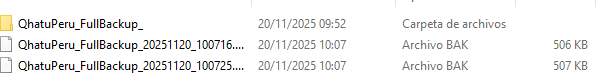
Para la implementación del backup diario y el job que ejecuta la operación (SQL Agent)



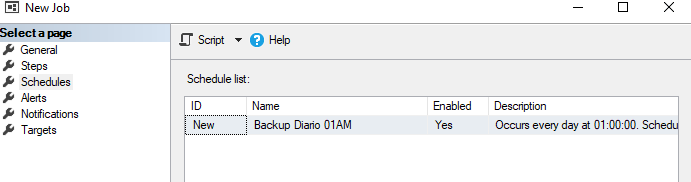


**RESULTADO**

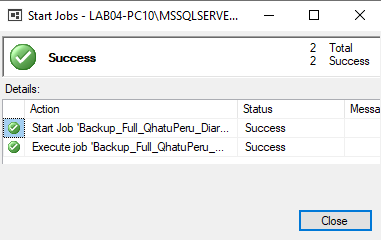
Creación del backup de la base de datos



Creación del job y configuración de fecha



Ejecución del job de SQL Agent



**JUSTIFICACION TECNICA DE LA SOLUCION APLICADA**

* **Backup FULL diario:** Se utiliza **BACKUP DATABASE** con la opción de **INIT** para sobrescribir backups antiguos y **COMPRESSION** para optimizar espacio.
* **Archivo con fecha:** El nombre del archivo incluye fecha y hora para evitar sobrescribir backups y facilitar la organización y restauración.
* **Verificación de integridad:** Se ejecuta **RESTORE VERIFYONLY** para asegurar que el backup es legible y no está corrupto.
* **Automatización con SQL Agent:** El job de SQL Agent permite automatizar el proceso, garantizando que el backup se realice diariamente sin intervención manual, mejorando la confiabilidad.

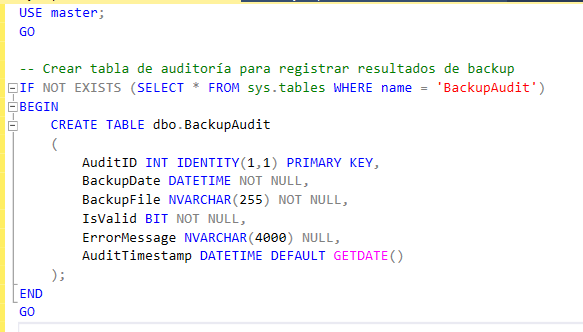
**EXPLICACION DE BUENAS PRACTICAS UTILIZADAS**

* **Automatización:** Uso del SQL Agent para evitar errores y asegurar la constancia diaria del backup.
* **Nomenclatura clara:** Incluir fecha y hora en el nombre del archivo ayuda en la gestión y recuperación rápida.
* **Compresión:** Reduce espacio en disco y mejora la eficiencia del almacenamiento.
* **Verificación:** Realizar **RESTORE VERIFYONLY** evita almacenar backups

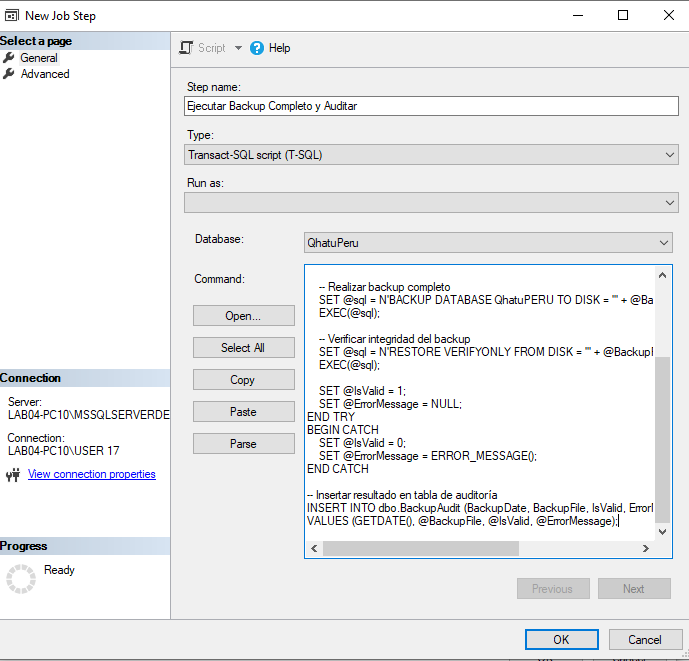
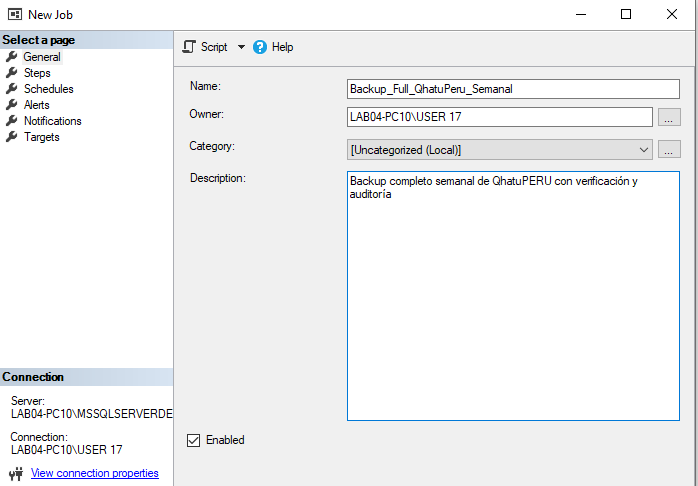


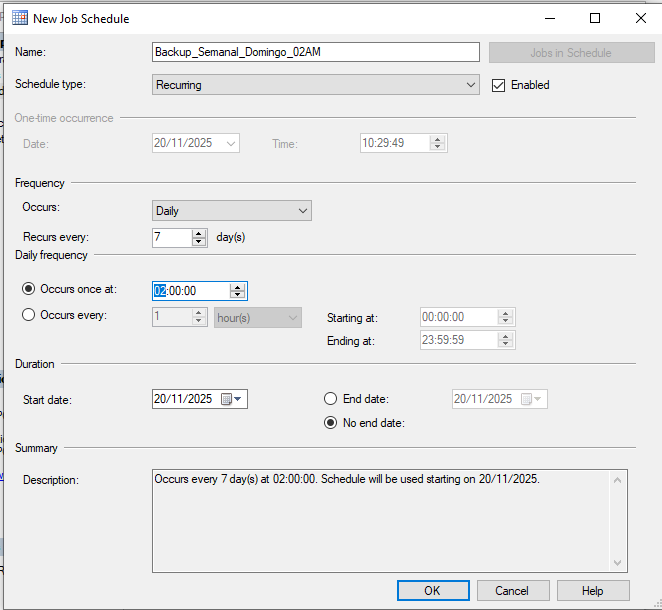
**SCRIPT DE LA SOLUCION**

Creamos la tabla para simular la auditoria

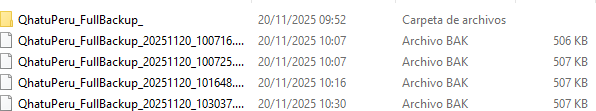
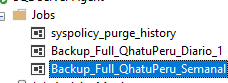


Creamos el job en el SQL Agent





**RESULTADO**



****

**JUSTIFICACION TECNICA DE LA SOLUCION APLICADA**

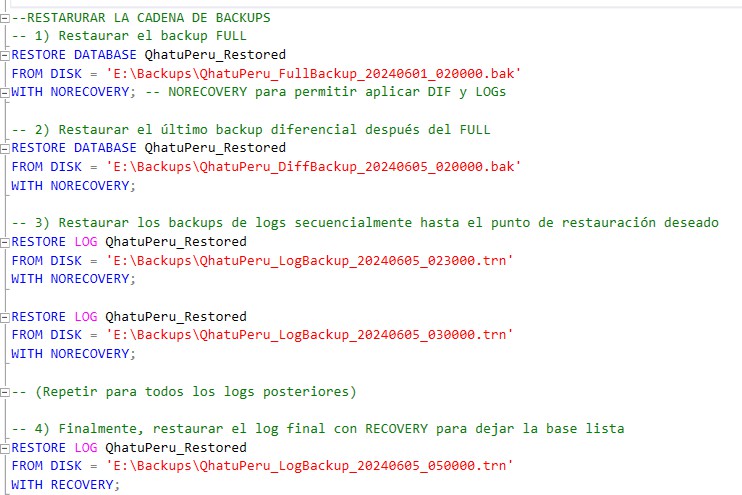
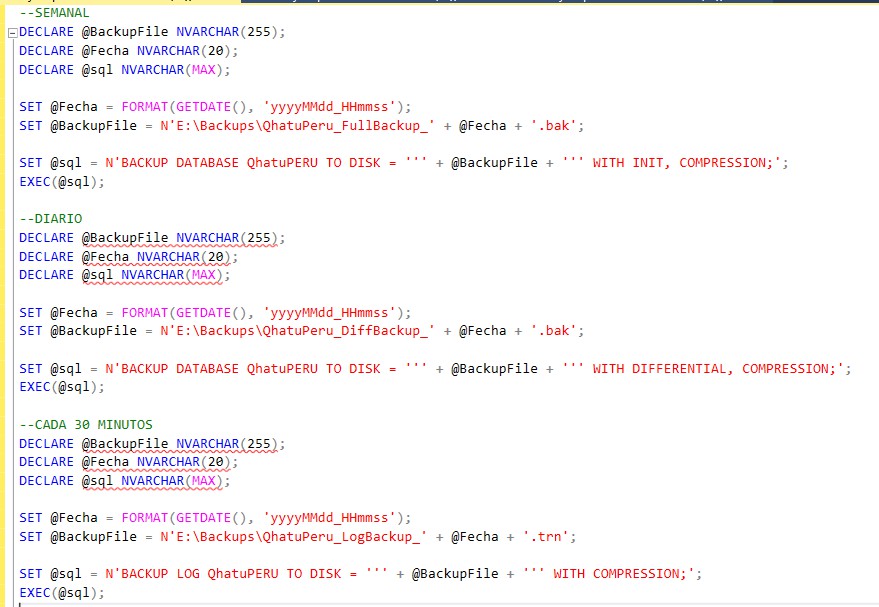
* Se crea una tabla de auditoría **BackupAudit** para registrar fecha, ruta del backup, si fue válido y un mensaje de error (si ocurrió).
* El backup completo se realiza dinámicamente con **BACKUP DATABASE**.
* Se ejecuta **RESTORE VERIFYONLY** para comprobar el archivo de backup.
* Se maneja errores con **TRY CATCH** para capturar fallas y registrar en auditoría.
* Se programa el job para que se ejecute cada domingo a las 2:00 AM usando SQL Server Agent.
* Se incluya compresión para optimizar espacio.

**EXPLICACION DE BUENAS PRACTICAS UTILIZADAS**

* **Auditoría detallada:** Registro con fecha, ruta, estado y mensaje permite seguimiento y diagnóstico fácil.
* **Manejo de errores con TRY CATCH:** Previene que el job falle silenciosamente y permite capturar el error.
* **Programación semanal:** Optimiza recursos con backup full semanal, alternativa común a backups diarios.
* **Rutas y nombres con fechas:** Facilita restauración precisa y organización.
* **Compresión activada:** Ahorra espacio de almacenamiento.
* **Automatización con SQL Agent:** Asegura ejecución automática.



**SCRIPT DE LA SOLUCION**

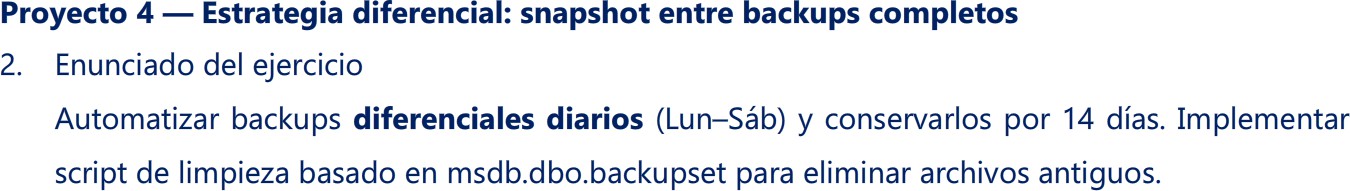


**JUSTIFICACION TECNICA DE LA SOLUCION APLICADA**

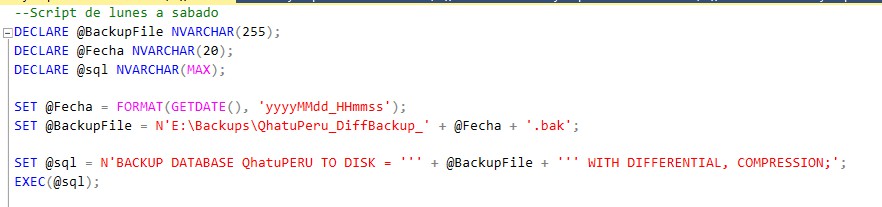
* El backup FULL semanal crea un punto base completo de la base.
* El backup diferencial diario guarda los cambios desde el último FULL, optimizando tiempo y espacio.
* Los backups LOG cada 30 minutos capturan transacciones frecuentes para minimizar posible pérdida de datos.
* La restauración utiliza la secuencia lógica: primero FULL, luego la última diferencia, y finalmente los logs en orden cronológico, aplicándolos con **NORECOVERY** para que continúe la cadena, y terminando con **RECOVERY** para poner la base en línea.
* Uso de compresión para optimizar almacenamiento.

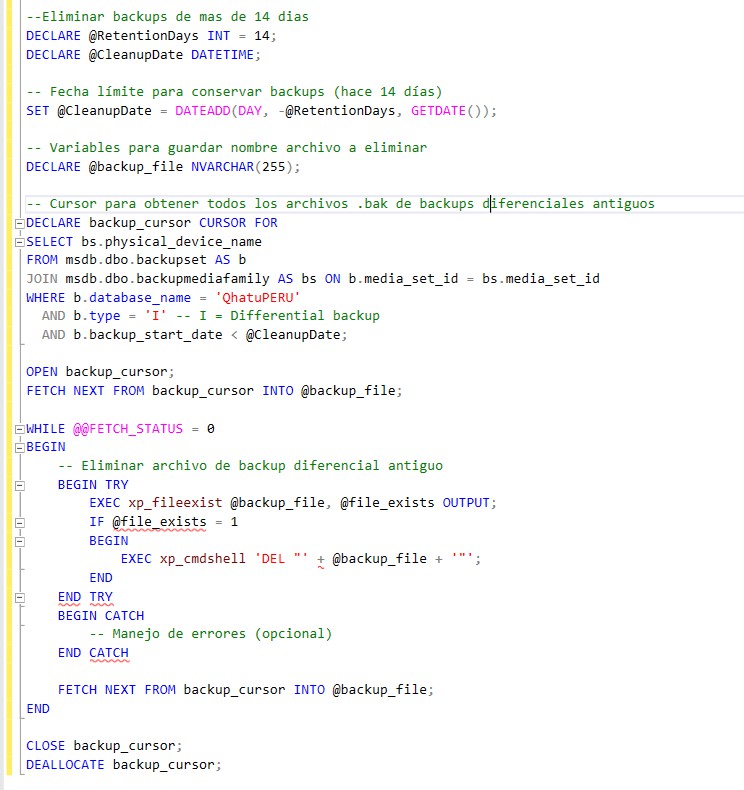
**EXPLICACION DE BUENAS PRACTICAS UTILIZADAS**

* **Separación de tipos de backup según frecuencia y tipo de datos** para optimizar desempeño y espacio.
* **Nombres de archivo con marca de fecha y hora**, facilita la identificación y restauración exacta.
* **Uso de recuperación con NORECOVERY en pasos intermedios** previene que se finalice la restauración prematuramente.
* **Backup de logs frecuente para minimizar ventana de pérdida de datos.**
* **Automatización recomendada** mediante jobs SQL Agent (aunque no está incluida aquí, es aconsejable).



**SCRIPT DE LA SOLUCION**



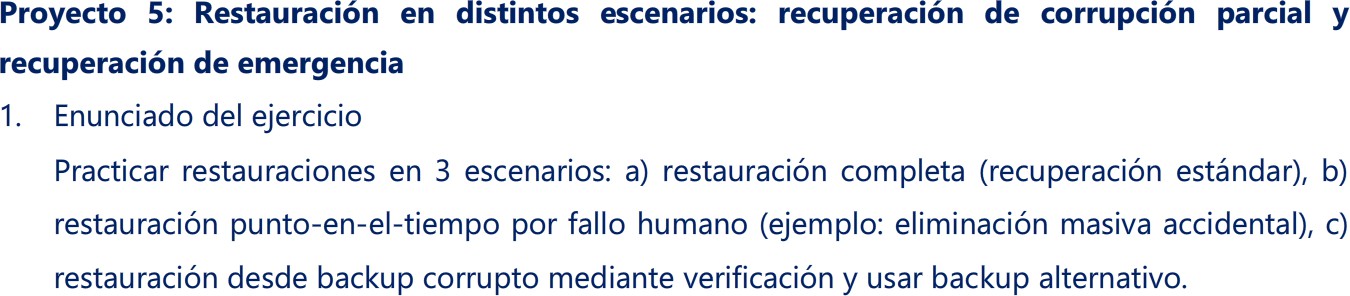


**JUSTIFICACION DE LA TECNICA APLICADA**

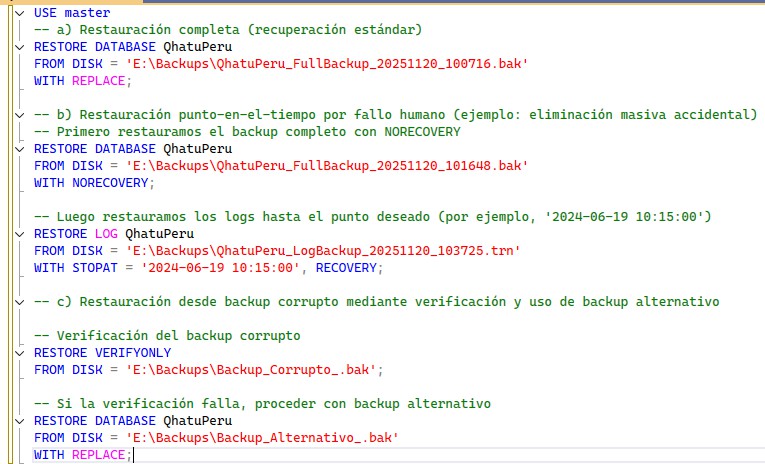
* El backup diferencial diario permite capturar cambios desde el último backup full, optimizando espacio y tiempo de respaldo.
* La retención por 14 días cumple con prácticas comunes para balancear recuperación y almacenamiento.
* La limpieza se realiza consultando la tabla **msdb.dbo.backupset** para obtener la metadata oficial de backups y evitar eliminar archivos incorrectos.
* Uso de **xp\_cmdshell** para eliminar archivos físicos con precaución.
* Uso de cursores para procesar todos los archivos a eliminar.

**EXPLICACION DE LAS BUENAS PRACTICAS APLICADAS**

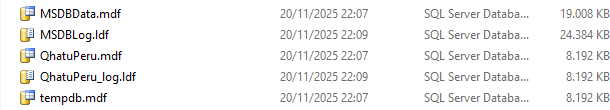
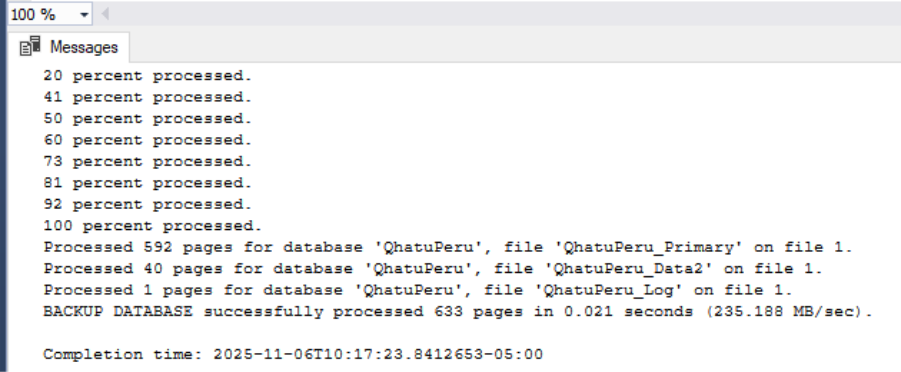
* Control de retención automático para evitar crecimiento descontrolado de backups.
* Diferenciación correcta de tipo de backup (**type = 'I'**) para seleccionar backups diferenciales.
* Separación entre backup y limpieza para mejor mantenimiento y control.
* Se recomienda asegurar permisos para **xp\_cmdshell** y ejecutar scripts con cuentas adecuadas para evitar riesgos.



**SCRIPT DE LA SOLUCION**



**RESULTADO**

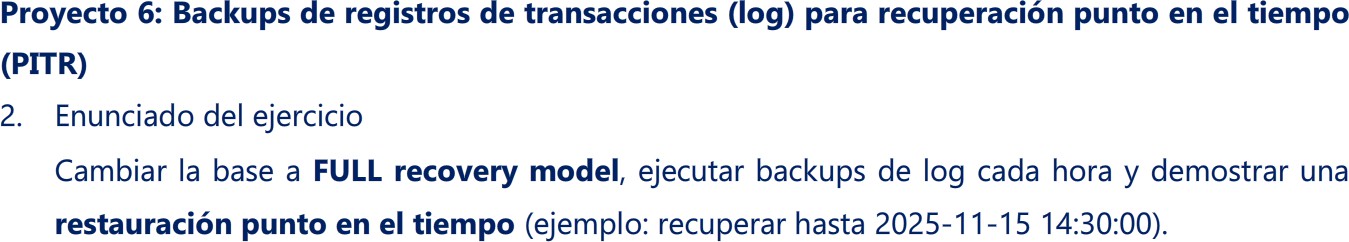


**JUSTIFICACION DE LAS TECNICAS APLICADAS**

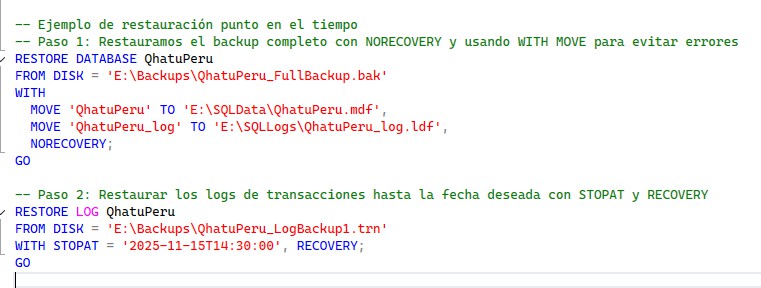
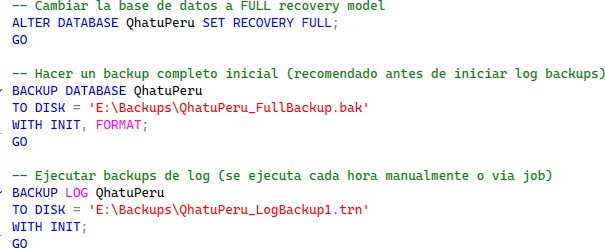
* Restauración completa permite recuperar la base de datos a su estado más reciente almacenado en el backup completo. Es la forma estándar y más sencilla de restaurar una base de datos.
* Restauración punto-en-el-tiempo es crucial ante fallos donde se necesita restaurar hasta justo antes del error, evitando pérdidas innecesarias de datos posteriores. Se usa el backup completo con NORECOVERY para mantener la base en estado no accesible y permitir restaurar los logs con la cláusula STOPAT para definir el momento exacto.
* Restauración desde backup corrupto (c) requiere validar primero la integridad del backup con RESTORE VERIFYONLY para detectar corrupción. Si el backup está corrupto, se debe proceder a usar un backup alternativo para asegurar la integridad y disponibilidad de los datos.

**EXPLICACION DE LAS BUENAS PRACTICAS**

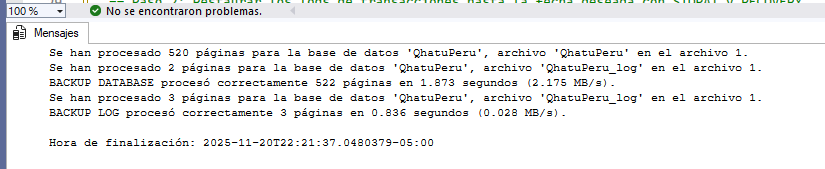
* Seguridad y confiabilidad: Uso de backups regulares y combinación de backups completos y logs para tener puntos de restauración flexibles.
* Pruebas de restauración: Ensayar diferentes escenarios de restauración para garantizar una recuperación rápida y efectiva ante cualquier incidente.
* Verificación del backup: Verificar integridad de los archivos de backup antes de su uso para evitar fallas durante la restauración.
* Uso de opciones específicas de RESTORE: Como WITH NORECOVERY y WITH STOPAT para controlar el proceso y minimizar pérdida de datos.
* Documentación clara: Definir claramente el procedimiento para cada tipo de restauración con scripts reproducibles.



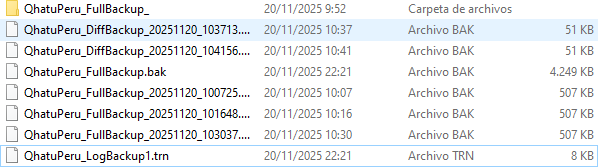
**SCRIPT DE LA SOLUCION**

****

**RESULTADO**

****

Archivos modificados con los nombres respectivos

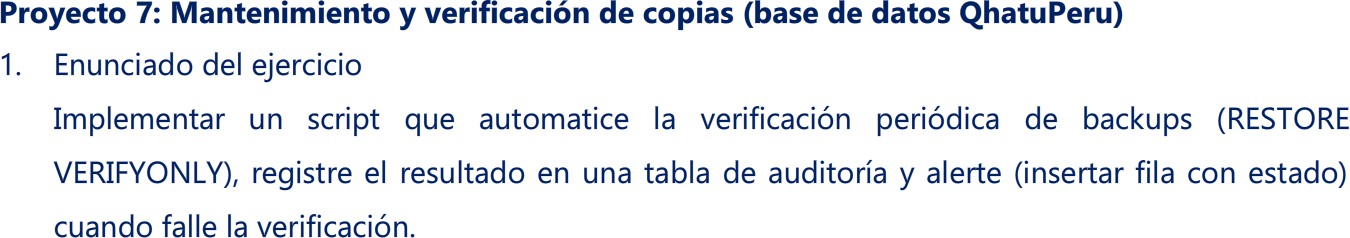


**JUSTIFICACION DE LAS TECNICAS APLICADAS**

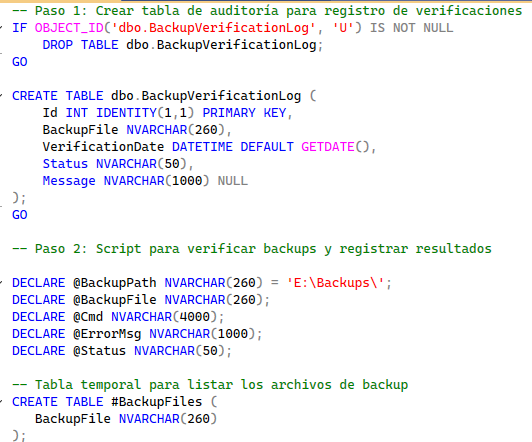
* Cambiar el modelo de recuperación a **FULL** es indispensable para poder realizar backups de log y así llevar a cabo restauraciones punto en el tiempo.
* Realizar backups de log con la frecuencia deseada (cada hora en este caso) asegura que los cambios realizados entre backups completos se pueden recuperar sin pérdida.
* La restauración punto en el tiempo usa el backup completo y luego aplica el backup de log hasta el instante deseado con la opción **STOPAT** para recuperar la base justo antes de un error o incidente.
* La cláusula **WITH MOVE** en la restauración es necesaria para especificar rutas válidas y evitar errores por ubicaciones de archivos no existentes o inaccesibles.

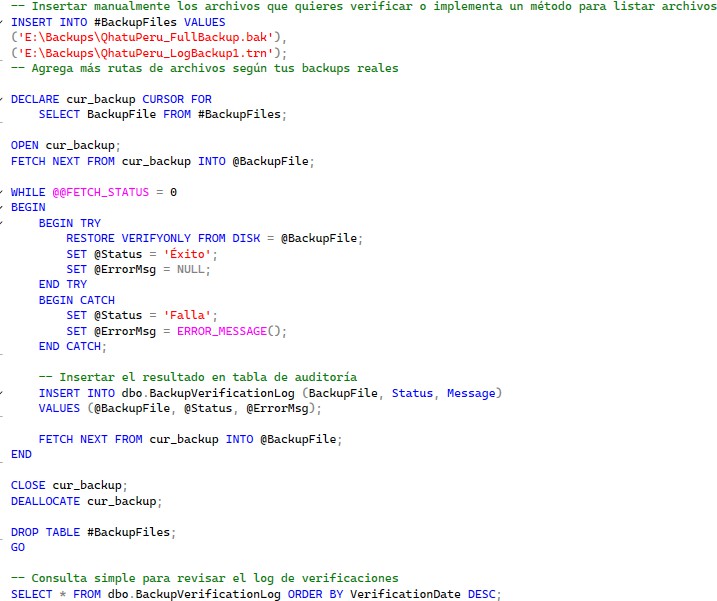
**EXPLICACION DE LAS BUENAS PRACTICAS**

* **Frecuencia de backup:** Hacer backups de log de forma regular (cada hora según el ejercicio) para minimizar la ventana de pérdida de datos.
* **Validación de backups:** Verificar los backups para asegurar su integridad antes de usarlos en restauraciones.
* **Separación de archivos:** Guardar los archivos de datos y logs en discos diferentes para mejorar rendimiento y administración.
* **Documentación:** Mantener un registro de horarios y rutas de backups para facilitar restauraciones rápidas.

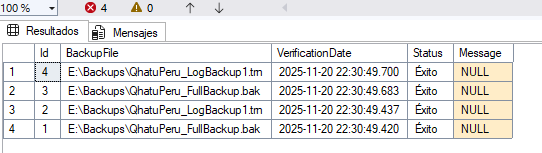


**SCRIPT DE LA SOLUCION**



****

**RESULTADO**

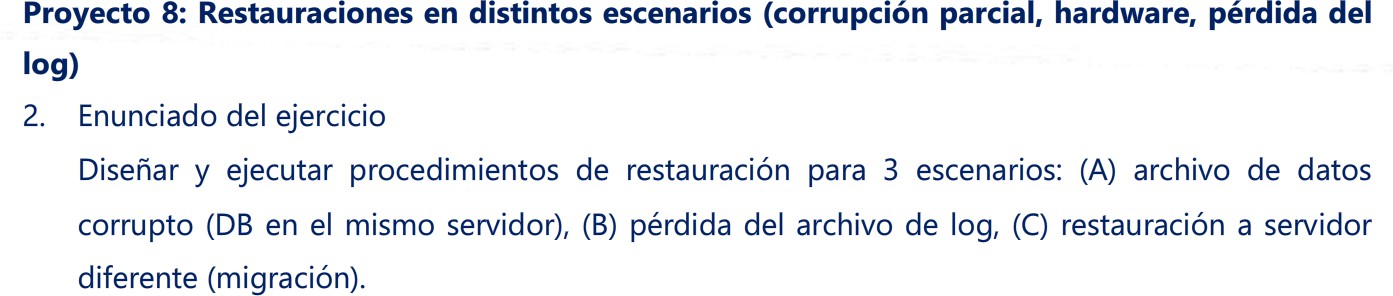


**JUSTIFICACION DE LA SOLUCION APLICADA**

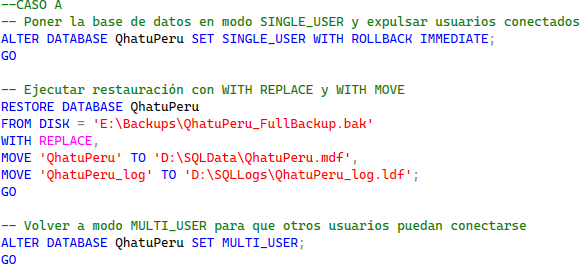
* **RESTORE VERIFYONLY** valida que el archivo de backup esté íntegro y pueda ser restaurado sin errores, sin realizar la restauración completa.
* Crear una **tabla de auditoría** permite registrar historial de verificaciones, con el estado y mensajes de error en caso de fallas.
* Se usa un **cursor** para iterar sobre cada archivo de backup que se desea verificar, lo que es útil si tienes varios backups regulares (completos, logs, diferenciales).
* En el bloque de manejo de errores **TRY...CATCH** se captura cualquier fallo durante la verificación para registrar el mensaje y el indicador de error.
* La tabla puede posteriormente ser consultada para monitorear el estado de los backups y tomar acciones preventivas.

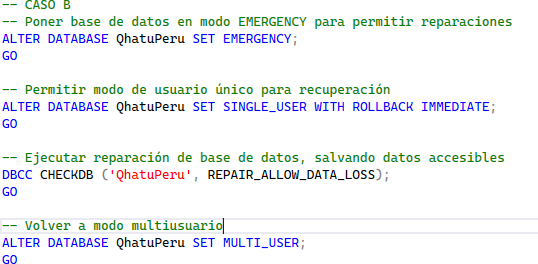
**EXPLICACION DE LAS BUENAS PRACTICAS**

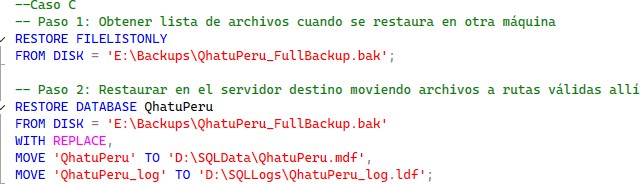
* **Automatización:** Facilita la gestión y control automático de la integridad de backups, evitando restauraciones fallidas en momentos críticos.
* **Auditoría:** Permite un seguimiento histórico y trazabilidad de las verificaciones.
* **Manejo de errores explícito:** Para registrar detalles detallados de la falla y facilitar troubleshooting.
* **Escalabilidad:** El script es extensible para verificar múltiples archivos de backup agregando rutas.
* **Separación de responsabilidades:** La verificación se hace sin afectar la base de datos, y el registro de resultados se guarda para análisis posterior.



**SCRIPT DE LA SOLUCION**

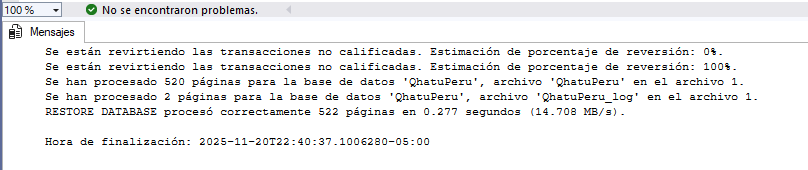


****

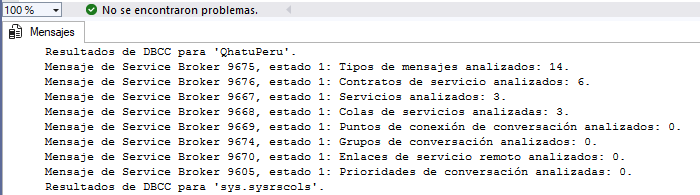
****

**RESULTADO**

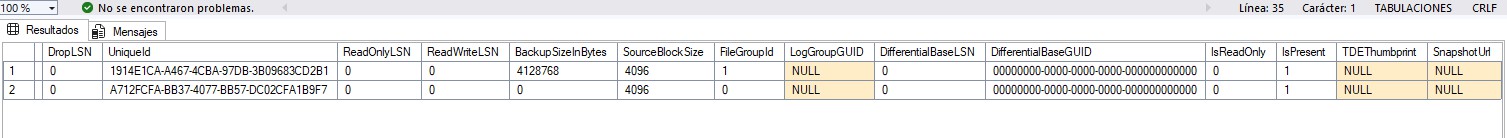
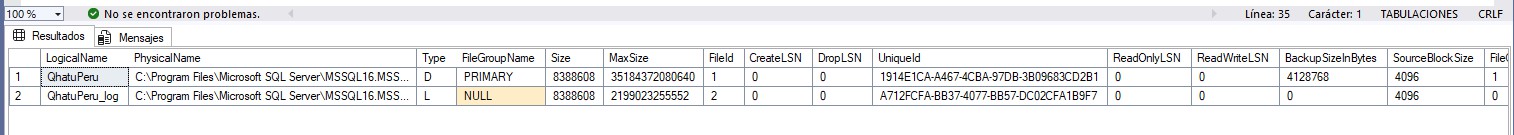
Caso A



Caso B



Caso C



**JUSTIFICACION DE LA SOLUCION APLICADA**

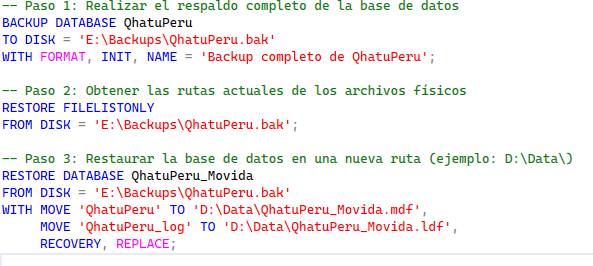
* **(A)** El comando **WITH REPLACE** permite sobrescribir los archivos corruptos físicos con los del backup, recuperando la base completa sin crear nueva base.
* **(B)** En caso de pérdida de archivo de log cuando no hay backups, la base se pone en modo EMERGENCY y se ejecuta **DBCC CHECKDB** con **REPAIR\_ALLOW\_DATA\_LOSS** para intentar recuperar lo máximo posible, asumiendo posibles pérdidas.
* **(C)** La restauración en un servidor diferente requiere conocer el nombre lógico de los archivos para usar **WITH MOVE** y así ubicar los archivos físicos en las rutas válidas del servidor destino.

**EXPLICACION DE LAS BUENAS PRACTICAS**

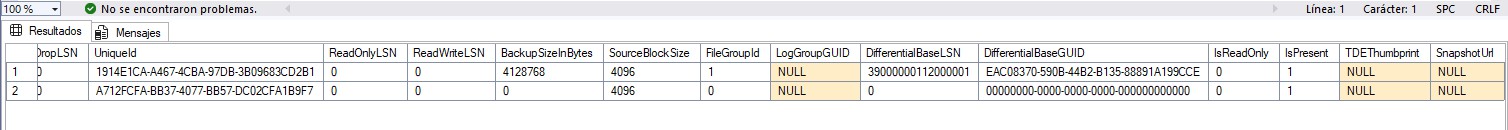
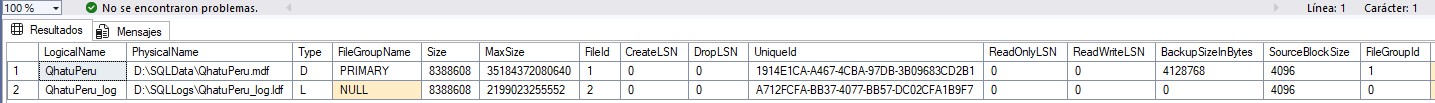
* **Backups regulares y verificación:** Para evitar pérdidas mayores y facilitar restauraciones completas.
* **Cuidado con modo emergencia:** Usar únicamente como último recurso porque implica posibles pérdidas.
* **Planificación de migraciones:** Preparar adecuadamente la ubicación de archivos al migrar a un nuevo servidor para evitar errores.
* **Documentar rutas y configuraciones:** Para facilitar restauraciones rápidas y correctas.
* **Pruebas periódicas:** De restauraciones en distintos escenarios para validar procedimientos y tiempos de recuperación.



**SCRIPT DE LA SOLUCION**

****

**RESULTADO**

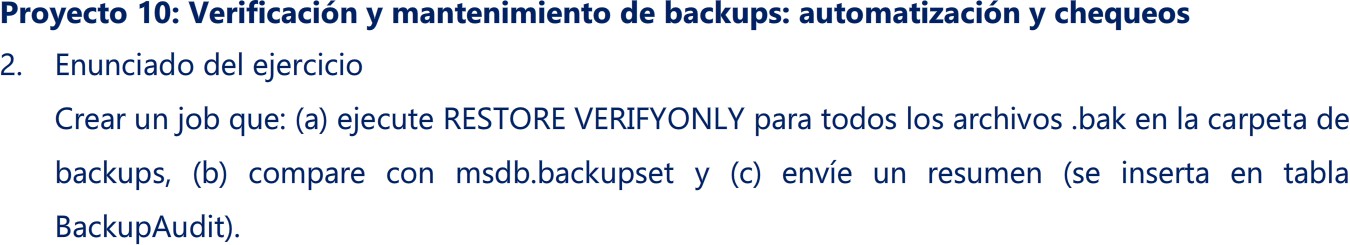


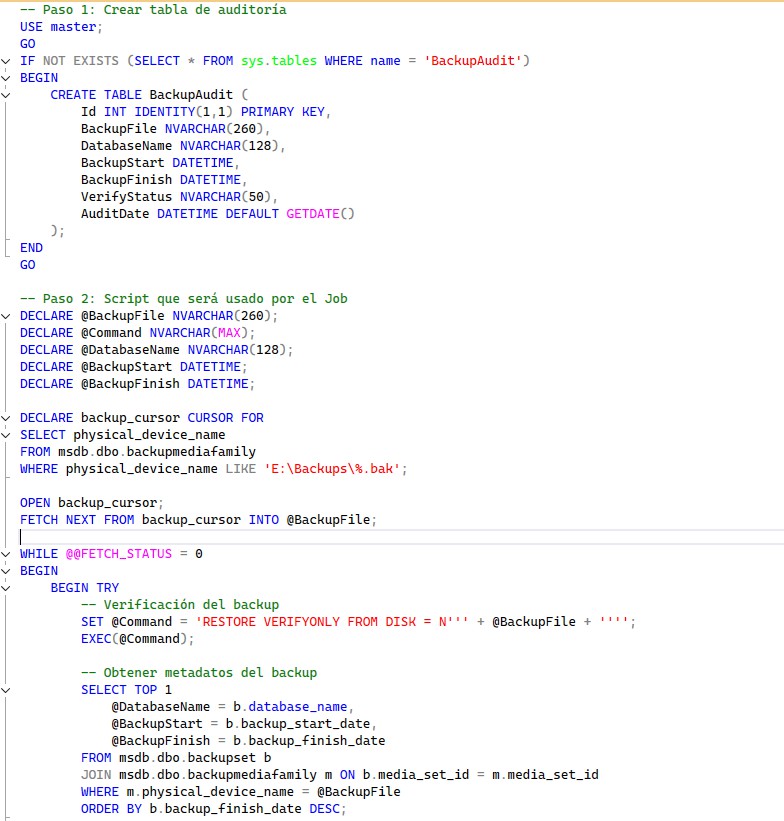
**JUSTIFICACION DE LA SOLUCION APLICADA**

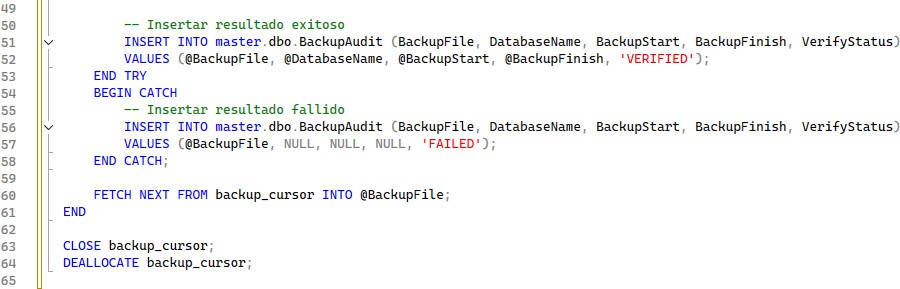
* Se usa BACKUP DATABASE para generar una copia segura y reutilizable de la base original.
* RESTORE FILELISTONLY permite identificar los nombres lógicos necesarios para el WITH MOVE.
* RESTORE DATABASE ... WITH MOVE redirige los archivos físicos a una nueva ubicación sin necesidad de modificar la base original.
* Se utiliza REPLACE para sobrescribir si ya existe una base con el mismo nombre (solo si es necesario).
* Se crea una nueva instancia (QhatuPeru\_Movida) para preservar la original y validar el proceso sin riesgo.

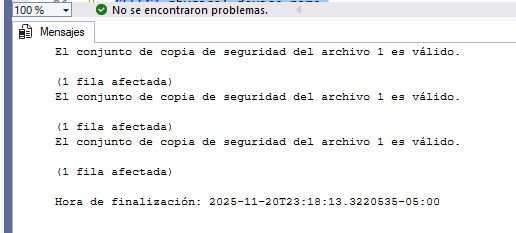
**EXPLICACION DE LAS BUENAS PRACTICAS**

* **Separación de entornos**: Restaurar como QhatuPeru\_Movida evita afectar la base original.
* **Uso de** RESTORE FILELISTONLY: Es esencial para obtener los nombres lógicos correctos y evitar errores en el MOVE.
* **Ruta de respaldo clara y documentada**: E:\Backups\ es una ubicación dedicada, lo que facilita la gestión y auditoría.
* **Evitar sobrescritura accidental**: Se recomienda no usar REPLACE a menos que se tenga control total del entorno.
* **Validación post-restauración**: Consultar sys.master\_files asegura que los archivos fueron movidos correctamente.



**SCRIPT DE LA SOLUCION**



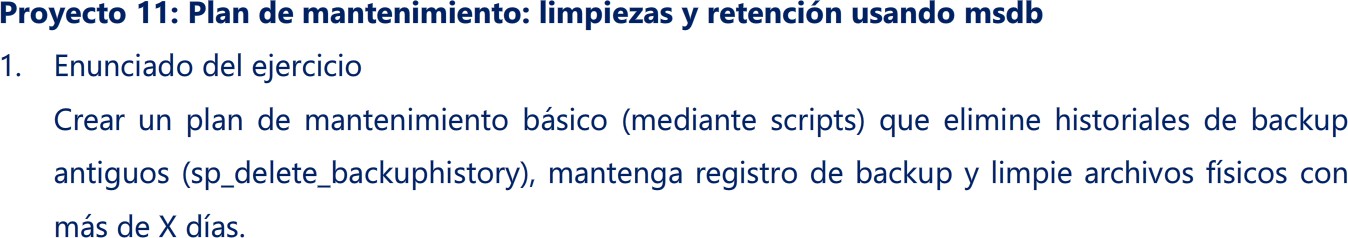
**RESULTADO**

**JUSTIFICACION DE LA SOLUCION APLICADA**

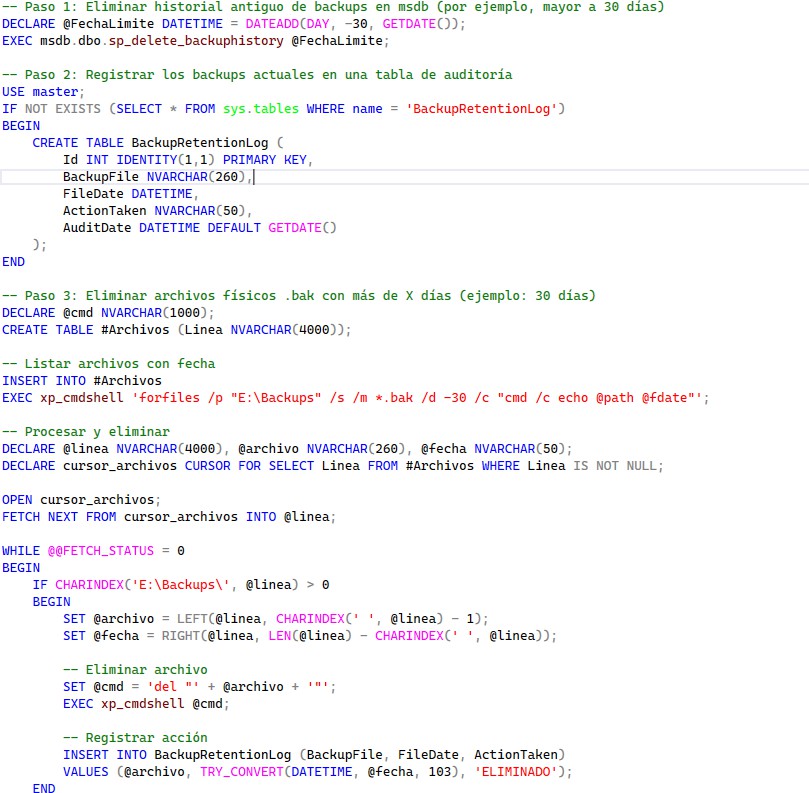
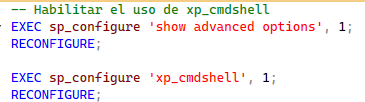
* Se usa RESTORE VERIFYONLY para validar la integridad del archivo sin restaurar la base.
* Se consulta msdb.dbo.backupset para obtener metadatos confiables del backup.
* Se registra todo en BackupAudit para trazabilidad y auditoría.
* Se usa TRY...CATCH para manejar errores y registrar fallos sin detener el proceso.
* Se implementa un cursor para recorrer múltiples archivos de forma automatizada.

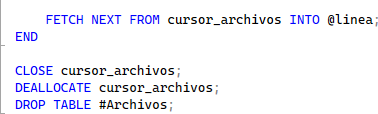
**EXPLICACION DE LAS BUENAS PRACTICAS**

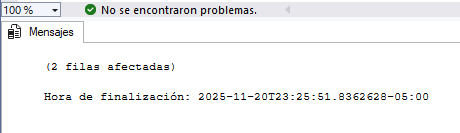
* **Automatización con SQL Server Agent**: Permite ejecutar el proceso periódicamente sin intervención manual.
* **Verificación sin restauración**: RESTORE VERIFYONLY es eficiente y segura para validar backups.
* **Auditoría centralizada**: La tabla BackupAudit permite monitorear el estado de los respaldos en un solo lugar.
* **Manejo de errores robusto**: El uso de TRY...CATCH evita que un error detenga el proceso completo.
* **Consulta de metadatos confiables**: msdb.dbo.backupset garantiza que los datos del backup sean precisos y auditables.

**SCRIPT**

**DE LA SOLUCION**

****



**RESULTADO**

**JUSTIFICACION DE LA SOLUCION APLICADA**

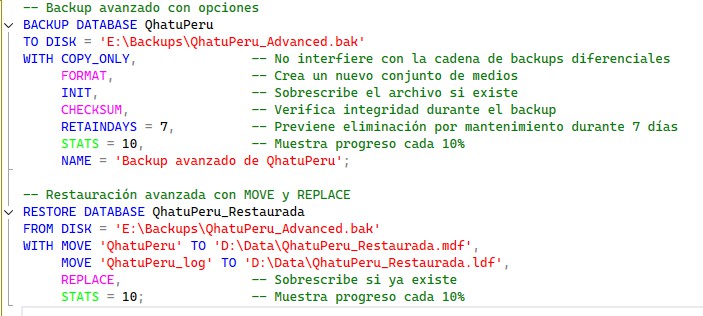
* sp\_delete\_backuphistory limpia el historial en msdb, evitando crecimiento innecesario.
* Se usa forfiles para identificar archivos antiguos sin depender de nombres específicos.
* xp\_cmdshell permite ejecutar comandos del sistema para eliminar archivos directamente.
* Se registra cada eliminación en BackupRetentionLog para trazabilidad y auditoría.

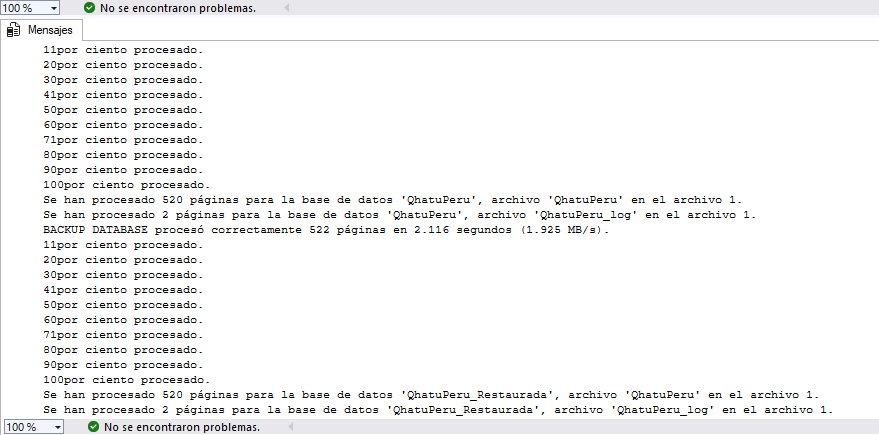
**EXPLICACION DE LAS BUENAS PRACTICAS**

* **Retención controlada**: Mantener solo los respaldos recientes reduce el uso de espacio y mejora la gestión.
* **Auditoría completa**: Registrar cada acción permite revisar qué se eliminó y cuándo.
* **Separación de lógica**: Separamos limpieza lógica (msdb) de limpieza física (.bak), evitando errores.
* **Automatización segura**: El uso de forfiles y xp\_cmdshell permite automatizar sin depender de nombres fijos.
* **Evitar sobresaturación de disco**: Eliminar archivos antiguos previene fallos por falta de espacio.



**SCRIPT DE LA SOLUCION**



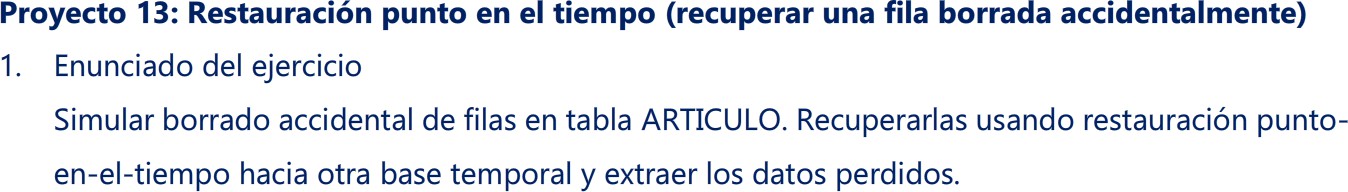
**RESULTADO**

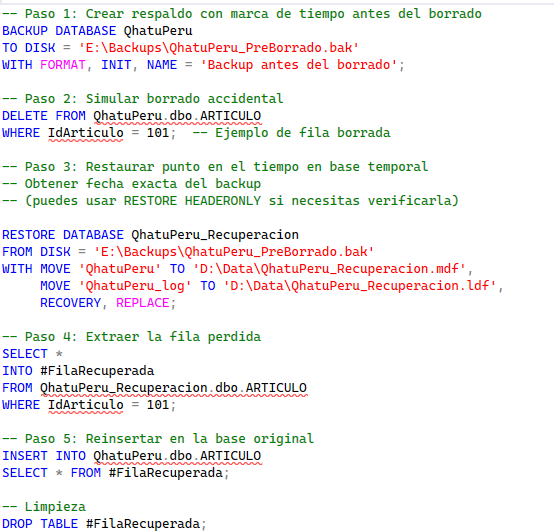
**JUSTIFICACION DE LA SOLUCION APLICADA**

* **COPY\_ONLY** evita interferencias con backups diferenciales programados.
* **FORMAT + INIT** aseguran que el archivo sea limpio y controlado.
* **CHECKSUM** valida la integridad del backup, detectando errores de disco o corrupción.
* **RETAINDAYS** protege el archivo de eliminación prematura por planes de mantenimiento.
* **STATS** mejora la trazabilidad visual del proceso.
* **MOVE + REPLACE** permiten restaurar en una nueva ubicación sin conflictos.

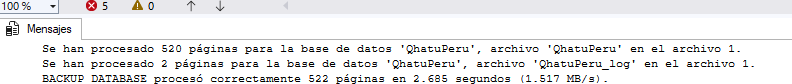
**EXPLICACION DE LAS BUENAS PRACTICAS**

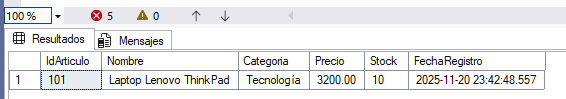
* **Integridad garantizada**: CHECKSUM y RESTORE VERIFYONLY son esenciales para entornos críticos.
* **No interferencia con backups programados**: COPY\_ONLY es ideal para respaldos manuales o de auditoría.
* **Retención controlada**: RETAINDAYS evita que el archivo sea eliminado por error.
* **Progreso visible**: STATS permite monitorear el avance, útil en bases grandes.
* **Restauración segura**: MOVE y REPLACE permiten simular restauraciones sin afectar la base original.



**SCRIPT DE LA SOLUCION**

**RESULTADO**



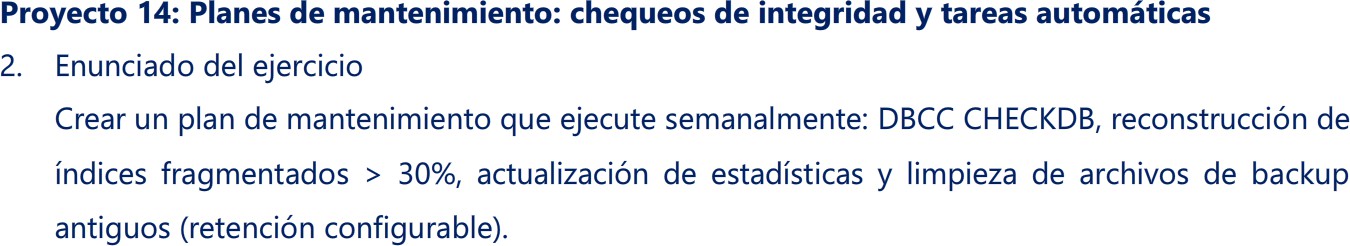
****

**JUSTIFICACION DE LA SOLUCION APLICADA**

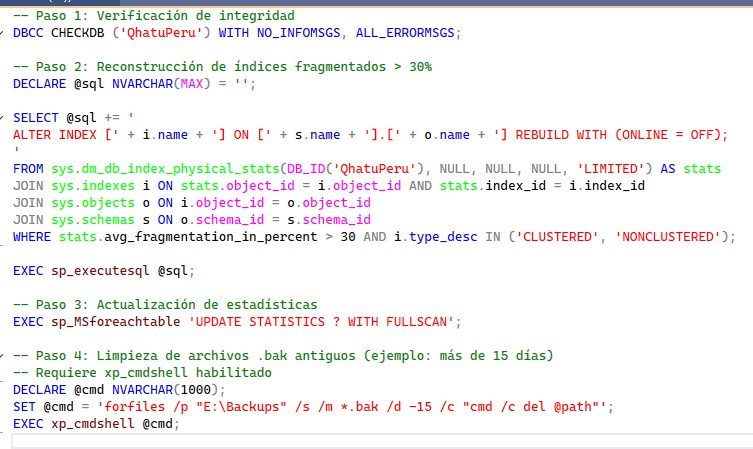
* Se realiza un respaldo preventivo antes de cualquier operación destructiva.
* Se simula el error para demostrar la técnica de recuperación.
* Se usa restauración en base temporal para evitar sobrescribir la base original.
* Se extrae la fila específica sin necesidad de restaurar toda la base.
* Se mantiene trazabilidad y control en cada paso.

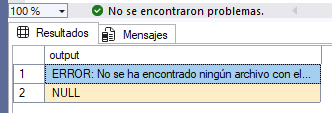
**EXPLICACION DE LAS BUENAS PRACTICAS**

* **Backup antes de operaciones críticas**: Siempre respaldar antes de modificar datos sensibles.
* **Restauración en base temporal**: Evita riesgos y permite recuperación selectiva.
* **Uso de tablas temporales**: Facilita la extracción y reinserción sin afectar estructuras.
* **Separación de entornos**: Mantener la base original intacta durante el proceso.
* **Validación post-reinserción**: Confirmar que los datos recuperados son correctos.



**SCRIPT DE LA SOLUCION**



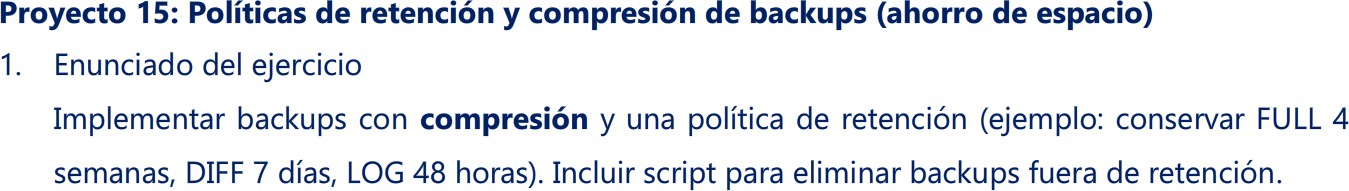
**RESULTADO**

**JUSTIFICACION DE LA SOLUCION APLICADA**

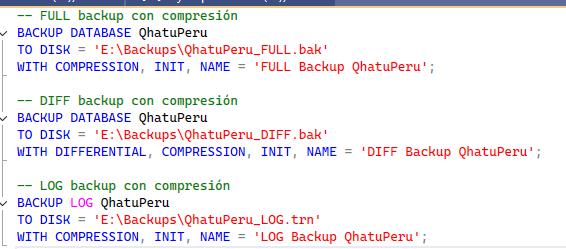
* Se ejecuta DBCC CHECKDB sobre QhatuPeru y se reportan errores si existen.
* Se reconstruyen automáticamente los índices con fragmentación superior al 30%.
* Se actualizan todas las estadísticas con escaneo completo (FULLSCAN).
* Se eliminan archivos .bak con más de 15 días de antigüedad en E:\Backups.

**EXPLICACION DE LAS BUENAS PRACTICAS**

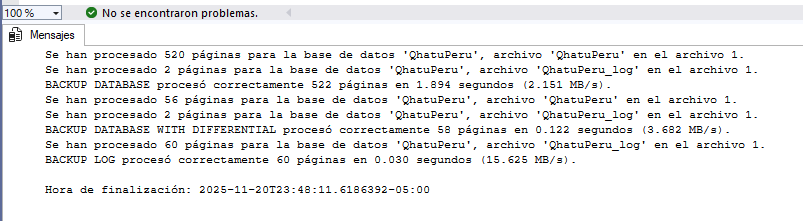
* DBCC CHECKDB detecta corrupción lógica y física en la base.
* La reconstrucción de índices mejora el rendimiento de consultas y operaciones.
* La actualización de estadísticas garantiza planes de ejecución óptimos.
* La limpieza de backups antiguos evita saturación de disco y mantiene orden.



**SCRIPT DE LA SOLUCION**

****

**RESULTADO**



**JUSTIFICACION DE LA SOLUCION APLICADA**

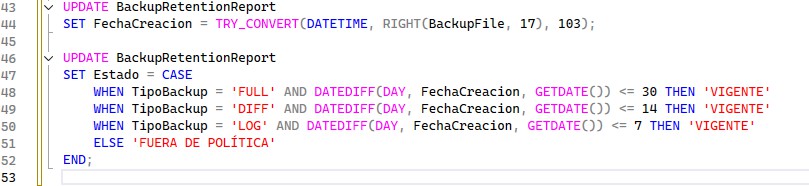
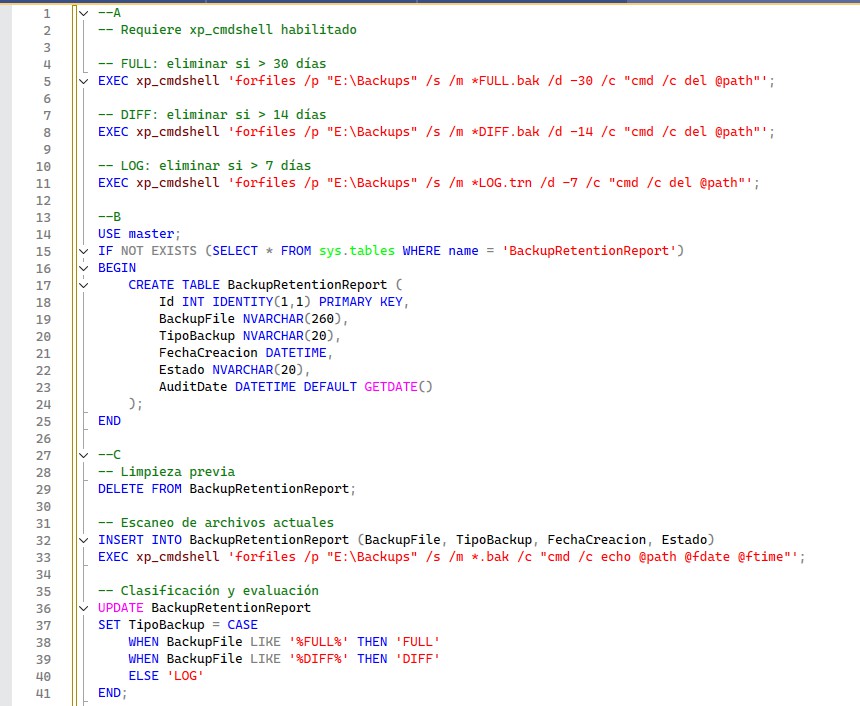
* La **compresión** reduce significativamente el tamaño de los archivos sin afectar la restauración.
* La **retención diferenciada** se ajusta al valor de cada tipo de backup:
* FULL: base sólida para restauraciones completas.
* DIFF: útil para restauraciones intermedias.
* LOG: crítico para restauraciones punto-en-el-tiempo, pero de vida corta.
* El uso de forfiles permite automatizar la limpieza sin depender de nombres exactos.

**EXPLICACION DE LAS BUENAS PRACTICAS**

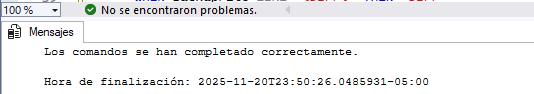
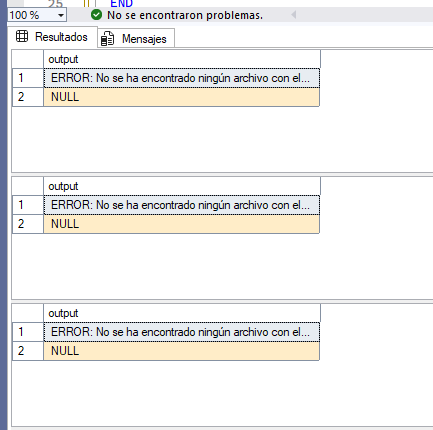
* **Compresión activa**: Reduce el uso de disco y mejora la velocidad de escritura.
* **Retención inteligente**: Cada tipo de backup tiene un ciclo de vida distinto, optimizado aquí.
* **Automatización con** forfiles: Evita acumulación manual de archivos y errores humanos.
* **Separación por tipo**: Facilita auditoría, restauración y limpieza selectiva.
* **Uso de nombres descriptivos**: Los sufijos FULL, DIFF, LOG permiten aplicar reglas específicas.



**SCRIPT DE LA SOLUCION**



**RESULTADO**

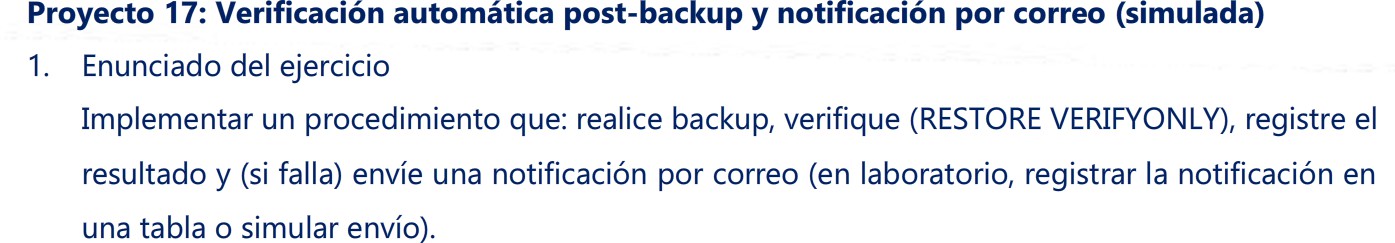


**JUSTIFICACION DE LA SOLUCION APLICADA**

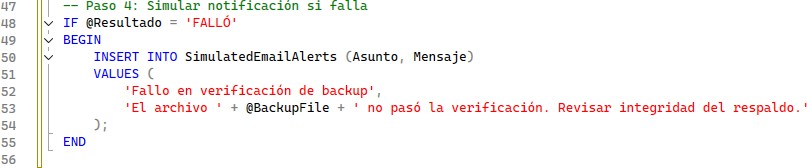
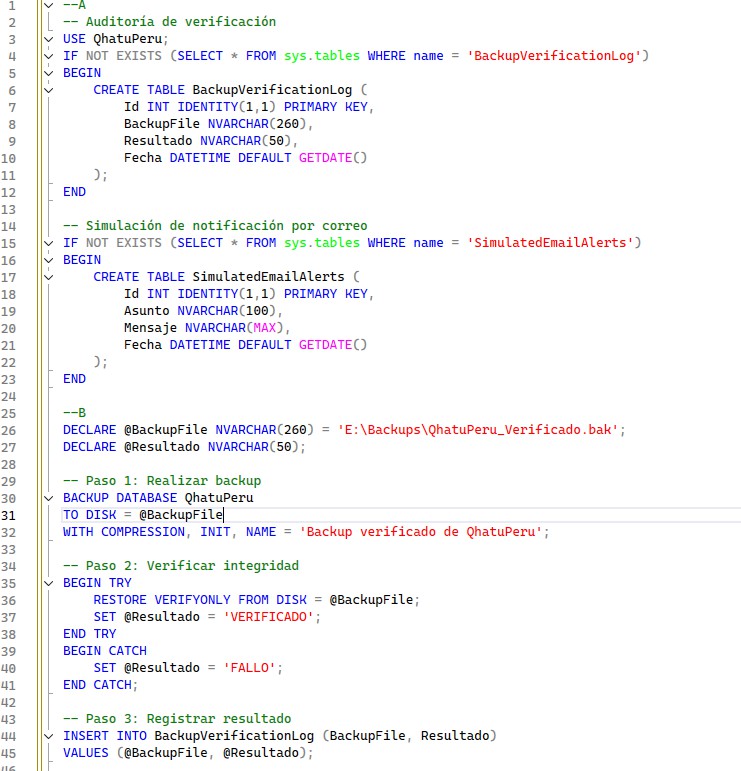
* Se implementa una política clara y diferenciada por tipo de backup.
* Se automatiza la limpieza para evitar acumulación innecesaria.
* Se genera un reporte auditable que permite verificar cumplimiento.
* Se usa forfiles y xp\_cmdshell para escaneo y eliminación sin depender de nombres fijos.

**EXPLICACION DE LAS BUENAS PRACTICAS**

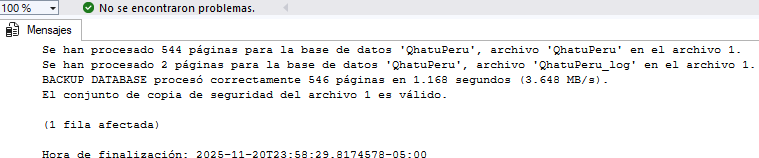
* **Retención diferenciada**: Cada tipo de backup tiene un ciclo de vida optimizado.
* **Automatización segura**: Elimina archivos sin intervención manual.
* **Auditoría centralizada**: La tabla BackupRetentionReport permite trazabilidad y revisión.
* **Clasificación inteligente**: Evalúa cada archivo según su tipo y antigüedad.
* **Separación de lógica**: Limpieza y reporte están desacoplados, facilitando mantenimiento.

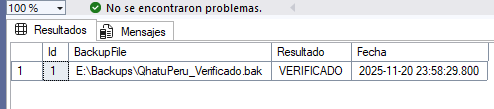


**SCRIPT DE LA SOLUCION**



**RESULTADO**



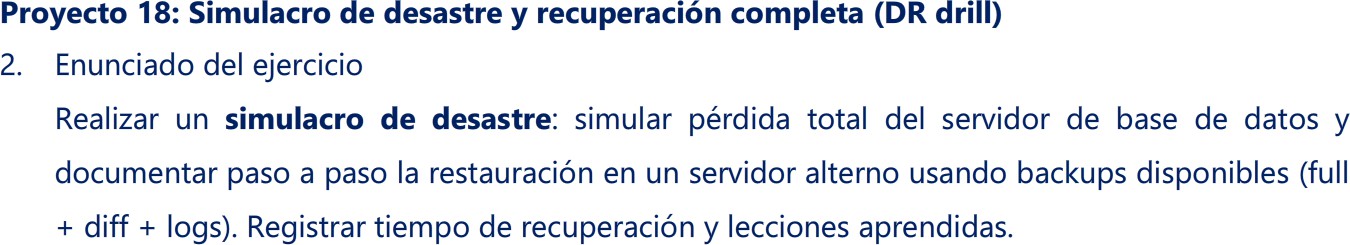
****

**JUSTIFICACION DE LA SOLUCION APLICADA**

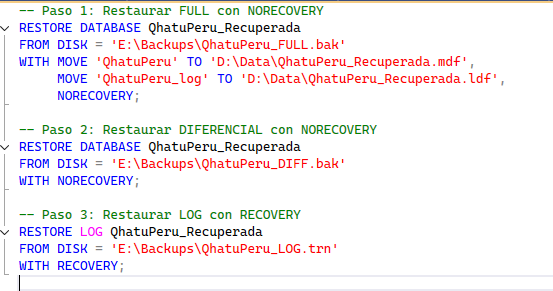
* Se automatiza el ciclo completo: respaldo, verificación, registro y alerta.
* Se usa TRY...CATCH para capturar errores sin detener el proceso.
* Se simula el envío de correo en laboratorio mediante una tabla de alertas.
* Se mantiene trazabilidad completa en dos tablas separadas.

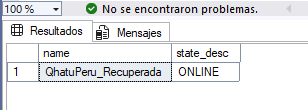
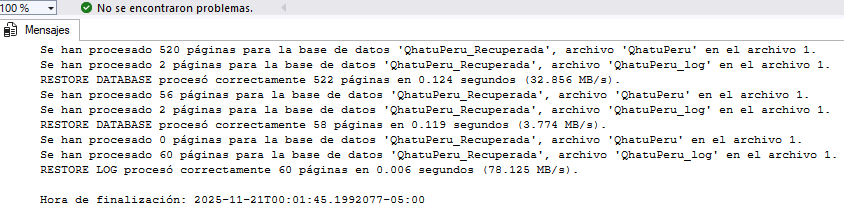
**EXPLICACION DE LAS BUENAS PRACTICAS**

* **Verificación inmediata**: RESTORE VERIFYONLY asegura que el backup es restaurable.
* **Registro estructurado**: Se documenta cada intento de verificación para auditoría.
* **Simulación controlada**: En entornos sin correo configurado, se simula la alerta sin perder trazabilidad.
* **Separación de responsabilidades**: Auditoría y alertas están desacopladas, facilitando mantenimiento.
* **Automatización escalable**: Este procedimiento puede integrarse fácilmente en un Job de SQL Server Agent.



**SCRIPT DE LA SOLUCION**



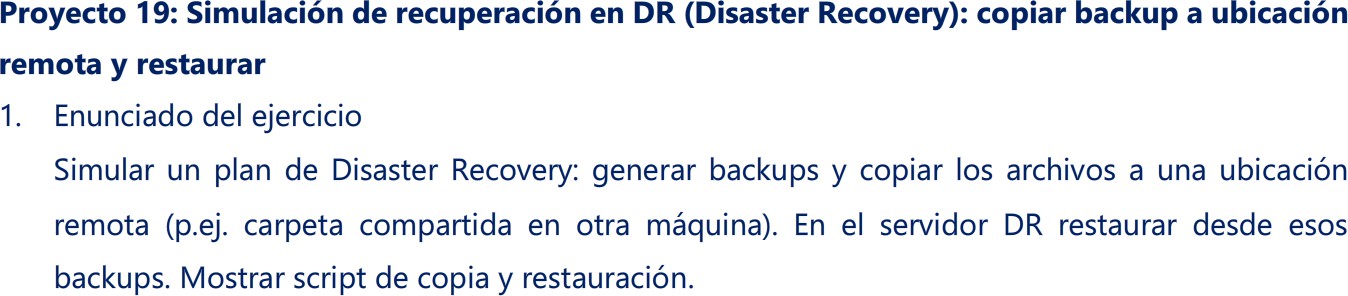
**RESULTADO**

**JUSTIFICACION DE LA SOLUCION APLICADA**

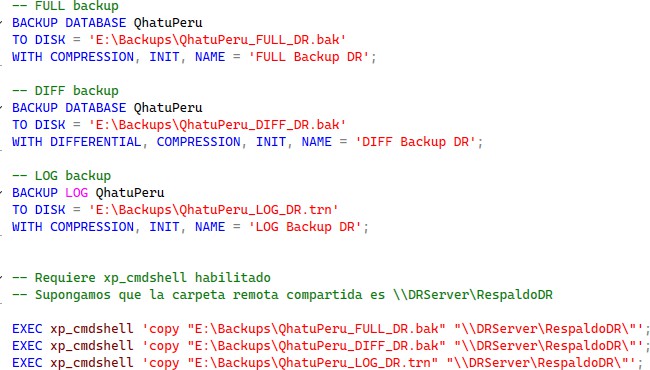
* **Restauración secuencial con NORECOVERY**: Se utilizó NORECOVERY en los pasos de restauración FULL y DIFERENCIAL para mantener la base en estado de recuperación pendiente, permitiendo aplicar los logs sin conflictos. Esto es esencial para preservar la consistencia transaccional.
* **Aplicación final con RECOVERY**: El uso de RECOVERY en el último paso (LOG) permite cerrar la secuencia de restauración y dejar la base operativa. Esta técnica garantiza que todos los cambios estén aplicados y que la base esté lista para uso.
* **Separación física en servidor alterno**: Restaurar en una ruta distinta (D:\Data\) evita sobrescribir archivos existentes y permite validar la recuperación sin afectar el entorno original.
* **Simulación controlada del desastre**: El ejercicio no requiere apagar el servidor real, sino simular la pérdida lógica y validar la capacidad de recuperación, lo cual es ideal para entornos académicos y pruebas de resiliencia.

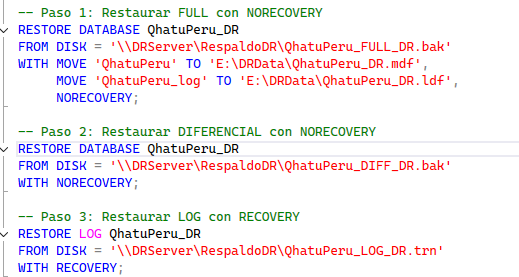
**EXPLICACION DE LAS BUENAS PRACTICAS**

* **Importancia de backups múltiples**: Tener FULL + DIFF + LOG permite restauración granular y rápida.
* **Separación de entornos**: Restaurar en servidor alterno evita riesgos y permite validación previa.
* **Uso correcto de NORECOVERY/RECOVERY**: Fundamental para aplicar secuencias sin cometer errores.
* **Documentación del proceso**: Registrar tiempos y pasos permite mejorar futuros DR drills.

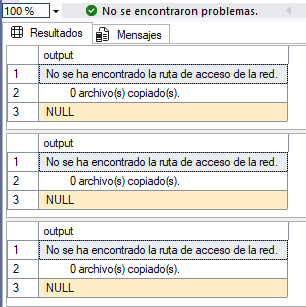
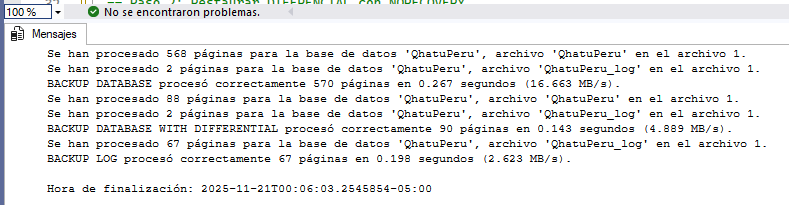


**SCRIPT DE LA SOLUCION**

****



**RESULTADO**



**JUSTIFICACION DE LA SOLUCION APLICADA**

* Se realiza una copia completa de los respaldos necesarios para una restauración punto-en-el-tiempo.
* Se usa xp\_cmdshell para automatizar la copia a una ubicación remota, simulando un entorno DR real.
* Se aplica la secuencia correcta de restauración: FULL → DIFF → LOG, usando

NORECOVERY y RECOVERY para mantener la integridad.

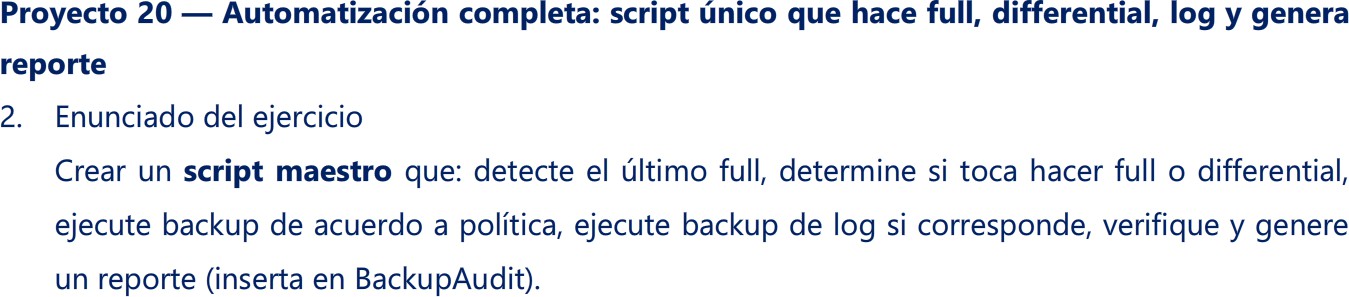
* Se restaura en una ruta distinta para evitar conflictos y validar la recuperación sin afectar el entorno original.

**EXPLICACION DE LAS BUENAS PRACTICAS**

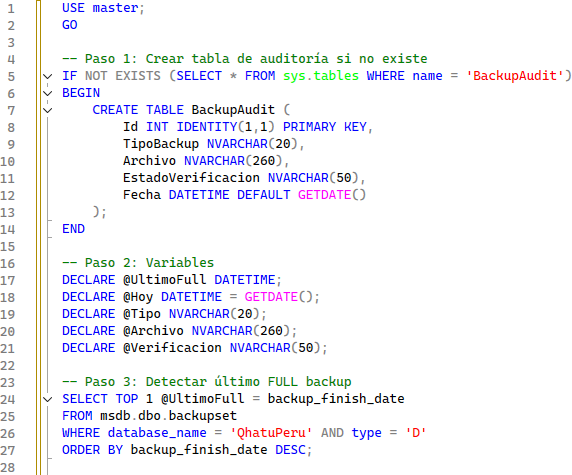
* **Separación de entornos**: Restaurar en servidor DR evita riesgos y permite validación segura.
* **Automatización del traslado**: Usar xp\_cmdshell con copy permite simular transferencia entre servidores.
* **Secuencia de restauración correcta**: Aplicar backups en orden con

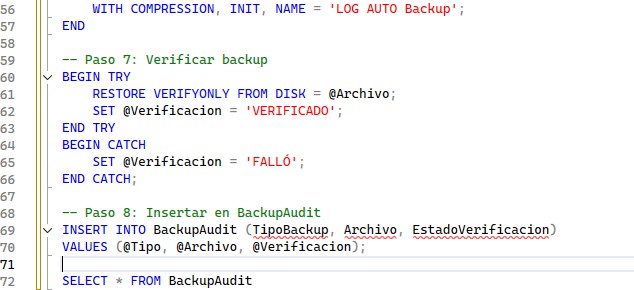
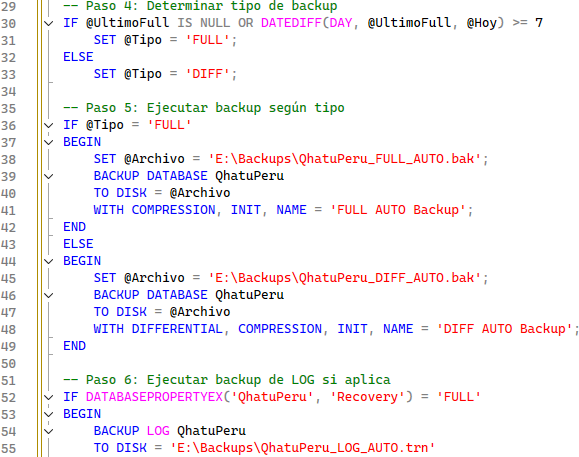
NORECOVERY y finalizar con RECOVERY garantiza consistencia.

* **Nombres descriptivos de archivos**: Facilita clasificación, trazabilidad y aplicación de políticas de retención.
* **Simulación realista**: Este ejercicio refleja un escenario DR completo, útil para auditoría y capacitación.

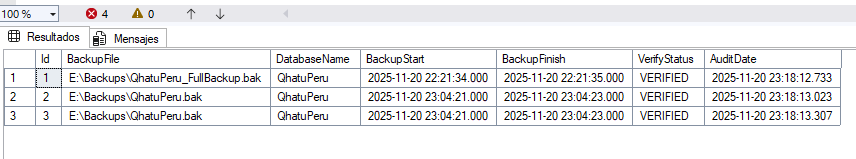
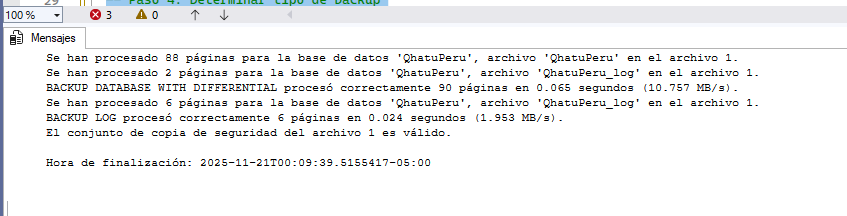


**SCRIPT DE LA SOLUCION**

****



**RESULTADO**



**JUSTIFICACION DE LA SOLUCION APLICADA**

* Se automatiza la lógica de respaldo según antigüedad del último FULL.
* Se evita hacer FULL innecesarios, optimizando espacio y tiempo.
* Se incluye respaldo de LOG solo si el modelo de recuperación lo permite.
* Se verifica la integridad del archivo antes de considerarlo válido.
* Se registra todo en una tabla para trazabilidad y auditoría.

**EXPLICACION DE LAS BUENAS PRACTICAS**

* **Respaldo inteligente**: Detectar el último FULL evita redundancia y mejora eficiencia.
* **Verificación inmediata**: RESTORE VERIFYONLY asegura que el archivo es restaurable.
* **Auditoría estructurada**: La tabla BackupAudit permite revisar historial y estado de cada respaldo.
* **Condicionalidad por modelo de recuperación**: Solo se hace LOG si el modelo lo permite.
* **Automatización escalable**: Este script puede integrarse como Job diario o semanal sin intervención manual.