

Universidad Peruana Los Andes

Facultad de Ingeniería

Escuela profesional de Ingeniería de Sistemas



UPLA
UNIVERSIDAD PERUANA LOS ANDES

PRACTICA SEMANA 12

Curso: Base de datos II

Docente: Raul Enrique Fernandez Bejarano

Estudiante: Sarmiento Mosquera Yeims Abraham

Ciclo: V - Código: s03807f

Huancayo - 2025

EJERCICIOS PRACTICOS DE LA SEMANA 12

Proyecto 1: Estrategia de backup completo diario y verificación automática

1. Enunciado del ejercicio

Implementar un script que haga un **backup FULL diario** de la base QhatuPeru, almacene el archivo con fecha, y verifique la integridad del backup (RESTORE VERIFYONLY). Programar (demostración por script) un job de SQL Agent que ejecute la operación.

SCRIPT DE LA SOLUCION

Para el Backup diario de la base de datos QhatuPeru

```
CREATE DATABASE QhatuPeru

DECLARE @BackupFile NVARCHAR(255);
DECLARE @Fecha NVARCHAR(20);
DECLARE @sql NVARCHAR(MAX);

-- Formato de la fecha - YYYYMMDD_HHMMSS
SET @Fecha = FORMAT(GETDATE(), 'yyyyMMdd_HHmss');

-- Ruta y nombre del archivo de backup
SET @BackupFile = N'E:\Backups\QhatuPeru_FullBackup_' + @Fecha + '.bak';

-- Construcción del comando de backup dinámico
SET @sql = N'BACKUP DATABASE QhatuPeru TO DISK = ''' + @BackupFile + ''' WITH INIT, COMPRESSION;';

-- Ejecutar el backup usando SQL dinámico
EXEC(@sql);

-- Verificar integridad del backup
SET @sql = N'RESTORE VERIFYONLY FROM DISK = ''' + @BackupFile + ''';
EXEC(@sql);
```

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

New Job

Select a page

General

Steps

Schedules

Alerts

Notifications

Targets

Script

Help

Name:

Backup_Full_QhatuPeru_Diario

Owner:

LAB04-PC10\USER 17

...

Category:

[Uncategorized (Local)]

...

Description:

Backup full diario de la base de datos QhatuPeru

☒ Enabled


Connection

Server:

LAB04-PC10\MSSQLSERVERDE

Connection:

LAB04-PC10\USER 17

 [View connection properties](#)

New Job Step

Select a page

- General
- Advanced

Script ? Help

Step name: Ejecutar Backup y Verificacion

Type: Transact-SQL script (T-SQL)

Run as:

Database: QhатуPeru

Command:

```
DECLARE @BackupFile NVARCHAR(255);
DECLARE @Fecha NVARCHAR(20);
DECLARE @sql NVARCHAR(MAX);

SET @Fecha = FORMAT(GETDATE(), 'yyyyMMdd_HH:mm:ss');
SET @BackupFile = N'E:\Backups\QhатуPeru_FullBackup_' + @Fecha;

SET @sql = N'BACKUP DATABASE QhатуPeru TO DISK = "' + @BackupFile + '" WITH NOFORMAT, NOINIT, NORECOVERY, NOVERIFY;';
EXEC (@sql);

SET @sql = N'RESTORE VERIFYONLY FROM DISK = "' + @BackupFile + '" WITH NOFORMAT, NOINIT, NORECOVERY, NOVERIFY;';
EXEC (@sql);
```

Open... Select All Copy Paste Parse

Connection

Server: LAB04-PC10\MSSQLSERVERDE

Connection: LAB04-PC10\USER 17

[View connection properties](#)

Progress

Ready

Previous Next

New Job Schedule

Name: Backup Diario 01AM

Jobs in Schedule

Schedule type: Recuring

Enabled

One-time occurrence

Date: 20/11/2025 Time: 10:14:48

Frequency

Occurs: Daily

Recurs every: 1 day(s)

Daily frequency

Occurs once at: 01:00:00

Occurs every: 1 hour(s)

Starting at: 00:00:00

Ending at: 23:59:59

Duration

Start date: 20/11/2025

End date: 20/11/2025

No end date:

Summary

Description: Occurs every day at 01:00:00. Schedule will be used starting on 20/11/2025.

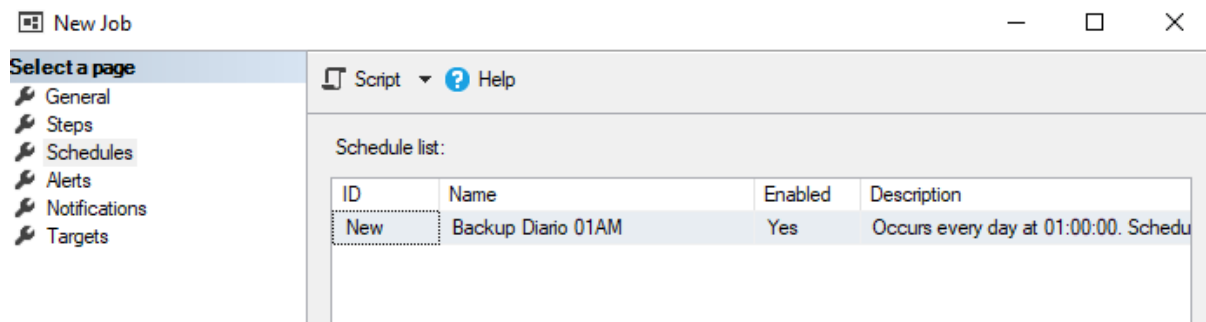
OK Cancel Help

RESULTADO

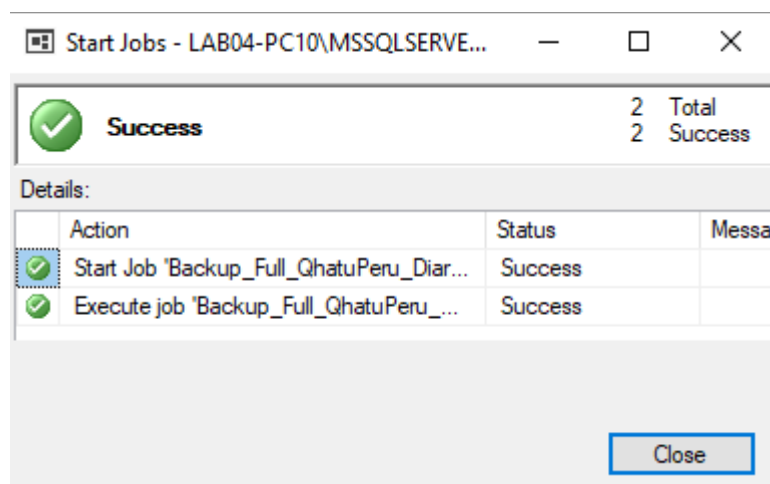
Creación del backup de la base de datos

QhatsuPeru_FullBackup_	20/11/2025 09:52	Carpeta de archivos	
QhatsuPeru_FullBackup_20251120_100716....	20/11/2025 10:07	Archivo BAK	506 KB
QhatsuPeru_FullBackup_20251120_100725....	20/11/2025 10:07	Archivo BAK	507 KB

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

Proyecto 2 — Estrategia completa: backup full semanal + verificación automática

2. Enunciado del ejercicio

Implementar un plan que realice un **backup completo** de la base QhatuPERU cada domingo a las 02:00, compruebe la validez del backup y registre el resultado en una tabla de auditoría.

SCRIPT DE LA SOLUCION

Creamos la tabla para simular la auditoria

```
USE master;
GO

-- Crear tabla de auditoría para registrar resultados de backup
IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'BackupAudit')
BEGIN
    CREATE TABLE dbo.BackupAudit
    (
        AuditID INT IDENTITY(1,1) PRIMARY KEY,
        BackupDate DATETIME NOT NULL,
        BackupFile NVARCHAR(255) NOT NULL,
        IsValid BIT NOT NULL,
        ErrorMessage NVARCHAR(4000) NULL,
        AuditTimestamp DATETIME DEFAULT GETDATE()
    );
END
GO
```

Creamos el job en el SQL Agent

New Job

Select a page

General

Steps

Schedules

Alerts

Notifications

Targets

Script

Help

Name:

Backup_Full_QhatuPeru_Semanal

Owner:

LAB04-PC10\USER 17

...

Category:

[Uncategorized (Local)]

...

Description:

Backup completo semanal de QhatuPERU con verificación y auditoría

Enabled

Connection

Server:
LAB04-PC10\MSSQLSERVERDE

Connection:
LAB04-PC10\USER 17

View connection properties

New Job Step

Select a page

General

Advanced

Script

Help

Step name:

Ejecutar Backup Completo y Auditar

Type:

Transact-SQL script (T-SQL)

...

Run as:

...

Database:

QhatuPeru

...

Command:

Open...

Select All

Copy

Paste

Parse

-- Realizar backup completo
SET @sql = N'BACKUP DATABASE QhatuPERU TO DISK = "" + @BackupFile;
EXEC(@sql);

-- Verificar integridad del backup
SET @sql = N'RESTORE VERIFYONLY FROM DISK = "" + @BackupFile;
EXEC(@sql);

SET @IsValid = 1;
SET @ErrorMessage = NULL;
END TRY
BEGIN CATCH
SET @IsValid = 0;
SET @ErrorMessage = ERROR_MESSAGE();
END CATCH

-- Insertar resultado en tabla de auditoría
INSERT INTO dbo.BackupAudit (BackupDate, BackupFile, IsValid, ErrorMessage) VALUES (GETDATE(), @BackupFile, @IsValid, @ErrorMessage);

Previous

Next

OK

Cancel

New Job Schedule

Name: Jobs in Schedule

Schedule type: Recurring ☒ Enabled

One-time occurrence

Date: Time:

Frequency

Occurs: Daily

Recurs every: day(s)

Daily frequency

☒ Occurs once at:

☐ Occurs every: hour(s)

Starting at: Ending at:

Duration

Start date: ☐ End date:

☒ No end date:


Summary

Description:

OK Cancel Help

RESULTADO

Jobs			
syspolicy_purge_history			
Backup_Full_QhatuPeru_Diario_1			
Backup_Full_QhatuPeru_Semanal			
QhatuPeru_FullBackup_	20/11/2025 09:52	Carpeta de archivos	
QhatuPeru_FullBackup_20251120_100716....	20/11/2025 10:07	Archivo BAK	506 KB
QhatuPeru_FullBackup_20251120_100725....	20/11/2025 10:07	Archivo BAK	507 KB
QhatuPeru_FullBackup_20251120_101648....	20/11/2025 10:16	Archivo BAK	507 KB
QhatuPeru_FullBackup_20251120_103037....	20/11/2025 10:30	Archivo BAK	507 KB

Action	Status	Message
 Start Job 'Backup_Full_QhatuPeru_Semanal'	Success	

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 incluye 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.

Proyecto 3: Estrategia combinada: FULL semanal + DIFF diario + LOG cada 30 minutos

1. Enunciado del ejercicio

Diseñar scripts que implementen: backup FULL semanal, backup diferencial diario (entre FULLs) y backup de transacciones (LOG) cada 30 minutos. Demostrar cómo restaurar una cadena (FULL + última DIFF + logs).

SCRIPT DE LA SOLUCION

```

--SEMANAL
DECLARE @BackupFile NVARCHAR(255);
DECLARE @Fecha NVARCHAR(20);
DECLARE @sql NVARCHAR(MAX);

SET @Fecha = FORMAT(GETDATE(), 'yyyyMMdd_HH:mm:ss');
SET @BackupFile = N'E:\Backups\QhatuPeru_FullBackup_' + @Fecha + '.bak';

SET @sql = N'BACKUP DATABASE QhatuPERU TO DISK = ''' + @BackupFile + ''' WITH INIT, COMPRESSION;';
EXEC(@sql);

--DIARIO
DECLARE @BackupFile NVARCHAR(255);
DECLARE @Fecha NVARCHAR(20);
DECLARE @sql NVARCHAR(MAX);

SET @Fecha = FORMAT(GETDATE(), 'yyyyMMdd_HH:mm:ss');
SET @BackupFile = N'E:\Backups\QhatuPeru_DiffBackup_' + @Fecha + '.bak';

SET @sql = N'BACKUP DATABASE QhatuPERU TO DISK = ''' + @BackupFile + ''' WITH DIFFERENTIAL, COMPRESSION;';
EXEC(@sql);

--CADA 30 MINUTOS
DECLARE @BackupFile NVARCHAR(255);
DECLARE @Fecha NVARCHAR(20);
DECLARE @sql NVARCHAR(MAX);

SET @Fecha = FORMAT(GETDATE(), 'yyyyMMdd_HH:mm:ss');
SET @BackupFile = N'E:\Backups\QhatuPeru_LogBackup_' + @Fecha + '.trn';

SET @sql = N'BACKUP LOG QhatuPERU TO DISK = ''' + @BackupFile + ''' WITH COMPRESSION;';
EXEC(@sql);

--RESTARURAR LA CADENA DE BACKUPS
-- 1) Restaurar el backup FULL
RESTORE DATABASE QhatuPeru_Restored
FROM DISK = 'E:\Backups\QhatuPeru_FullBackup_20240601_020000.bak'
WITH NORECOVERY; -- NORECOVERY para permitir aplicar DIF y LOGs

-- 2) Restaurar el último backup diferencial después del FULL
RESTORE DATABASE QhatuPeru_Restored
FROM DISK = 'E:\Backups\QhatuPeru_DiffBackup_20240605_020000.bak'
WITH NORECOVERY;

-- 3) Restaurar los backups de logs secuencialmente hasta el punto de restauración deseado
RESTORE LOG QhatuPeru_Restored
FROM DISK = 'E:\Backups\QhatuPeru_LogBackup_20240605_023000.trn'
WITH NORECOVERY;

RESTORE LOG QhatuPeru_Restored
FROM DISK = 'E:\Backups\QhatuPeru_LogBackup_20240605_030000.trn'
WITH NORECOVERY;

-- (Repetir para todos los logs posteriores)

-- 4) Finalmente, restaurar el log final con RECOVERY para dejar la base lista
RESTORE LOG QhatuPeru_Restored
FROM DISK = 'E:\Backups\QhatuPeru_LogBackup_20240605_050000.trn'
WITH RECOVERY;

```

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).

Proyecto 4 — Estrategia diferencial: snapshot entre backups completos

2. Enunciado del ejercicio

Automatizar backups **diferenciales diarios** (Lun-Sáb) y conservarlos por 14 días. Implementar script de limpieza basado en msdb.dbo.backupset para eliminar archivos antiguos.

SCRIPT DE LA SOLUCION

```
--Script de lunes a sabado
DECLARE @BackupFile NVARCHAR(255);
DECLARE @Fecha NVARCHAR(20);
DECLARE @sql NVARCHAR(MAX);

SET @Fecha = FORMAT(GETDATE(), 'yyyyMMdd_HHmmss');
SET @BackupFile = N'E:\Backups\QhatuPeru_DiffBackup_' + @Fecha + '.bak';

SET @sql = N'BACKUP DATABASE QhatuPERU TO DISK = ''' + @BackupFile + ''' WITH DIFFERENTIAL, COMPRESSION;';
EXEC(@sql);
```

```

--Eliminar backups de mas de 14 dias
DECLARE @RetentionDays INT = 14;
DECLARE @CleanupDate DATETIME;

-- Fecha límite para conservar backups (hace 14 días)
SET @CleanupDate = DATEADD(DAY, -@RetentionDays, GETDATE());

-- Variables para guardar nombre archivo a eliminar
DECLARE @backup_file NVARCHAR(255);

-- Cursor para obtener todos los archivos .bak de backups diferenciales antiguos
DECLARE backup_cursor CURSOR FOR
SELECT bs.physical_device_name
FROM msdb.dbo.backupset AS b
JOIN msdb.dbo.backupmediafamily AS bs ON b.media_set_id = bs.media_set_id
WHERE b.database_name = 'QhatuPERU'
AND b.type = 'I' -- I = Differential backup
AND b.backup_start_date < @CleanupDate;

OPEN backup_cursor;
FETCH NEXT FROM backup_cursor INTO @backup_file;

WHILE @@FETCH_STATUS = 0
BEGIN
    -- Eliminar archivo de backup diferencial antiguo
    BEGIN TRY
        EXEC xp_fileexist @backup_file, @file_exists OUTPUT;
        IF @file_exists = 1
        BEGIN
            EXEC xp_cmdshell 'DEL "' + @backup_file + '"';
        END
    END TRY
    BEGIN CATCH
        -- Manejo de errores (opcional)
    END CATCH

    FETCH NEXT FROM backup_cursor INTO @backup_file;
END

CLOSE backup_cursor;
DEALLOCATE backup_cursor;

```

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.

Proyecto 5: Restauración en distintos escenarios: recuperación de corrupción parcial y recuperación de emergencia

1. Enunciado del ejercicio

Practicar restauraciones en 3 escenarios: a) restauración completa (recuperación estándar), b) restauración punto-en-el-tiempo por fallo humano (ejemplo: eliminación masiva accidental), c) restauración desde backup corrupto mediante verificación y usar backup alternativo.

SCRIPT DE LA SOLUCION

```
USE master
-- a) Restauración completa (recuperación estándar)
RESTORE DATABASE QhatuPeru
FROM DISK = 'E:\Backups\QhatuPeru_FullBackup_20251120_100716.bak'
WITH REPLACE;

-- b) Restauración punto-en-el-tiempo por fallo humano (ejemplo: eliminación masiva accidental)
-- Primero restauramos el backup completo con NORECOVERY
RESTORE DATABASE QhatuPeru
FROM DISK = 'E:\Backups\QhatuPeru_FullBackup_20251120_101648.bak'
WITH NORECOVERY;

-- Luego restauramos los logs hasta el punto deseado (por ejemplo, '2024-06-19 10:15:00')
RESTORE LOG QhatuPeru
FROM DISK = 'E:\Backups\QhatuPeru_LogBackup_20251120_103725.trn'
WITH STOPAT = '2024-06-19 10:15:00', RECOVERY;

-- c) Restauración desde backup corrupto mediante verificación y uso de backup alternativo
-- Verificación del backup corrupto
RESTORE VERIFYONLY
FROM DISK = 'E:\Backups\Backup_Corrupto_.bak';

-- Si la verificación falla, proceder con backup alternativo
RESTORE DATABASE QhatuPeru
FROM DISK = 'E:\Backups\Backup_Alternativo_.bak'
WITH REPLACE;
```

RESULTADO

```
100 %
Messages
20 percent processed.
41 percent processed.
50 percent processed.
60 percent processed.
73 percent processed.
81 percent processed.
92 percent processed.
100 percent processed.
Processed 592 pages for database 'QhatuPeru', file 'QhatuPeru_Primary' on file 1.
Processed 40 pages for database 'QhatuPeru', file 'QhatuPeru_Data2' on file 1.
Processed 1 pages for database 'QhatuPeru', file 'QhatuPeru_Log' on file 1.
BACKUP DATABASE successfully processed 633 pages in 0.021 seconds (235.188 MB/sec).

Completion time: 2025-11-06T10:17:23.8412653-05:00

MSDBData.mdf          20/11/2025 22:07      SQL Server Databa...  19.008 KB
MSDBLog.ldf           20/11/2025 22:09      SQL Server Databa...  24.384 KB
QhatuPeru.mdf         20/11/2025 22:07      SQL Server Databa...   8.192 KB
QhatuPeru_log.ldf     20/11/2025 22:09      SQL Server Databa...   8.192 KB
tempdb.mdf            20/11/2025 22:07      SQL Server Databa...   8.192 KB
```

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.

Proyecto 6: Backups de registros de transacciones (log) para recuperación punto en el tiempo (PITR)

2. Enunciado del ejercicio

Cambiar la base a **FULL recovery model**, ejecutar backups de log cada hora y demostrar una **restauración punto en el tiempo** (ejemplo: recuperar hasta 2025-11-15 14:30:00).

SCRIPT DE LA SOLUCION

```
-- Cambiar la base de datos a FULL recovery model
ALTER DATABASE QhatuPeru SET RECOVERY FULL;
GO

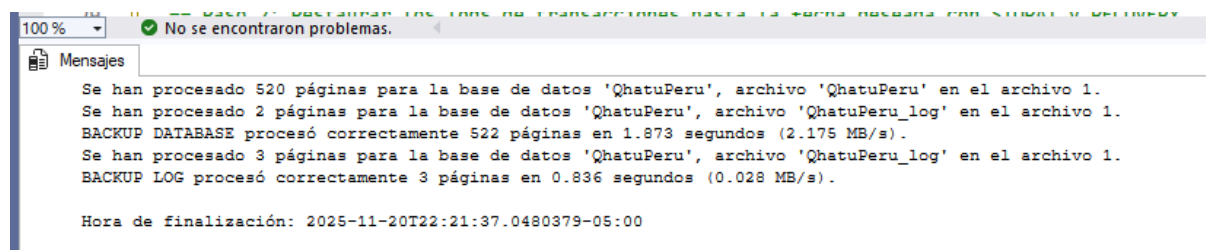
-- Hacer un backup completo inicial (recomendado antes de iniciar log backups)
BACKUP DATABASE QhatuPeru
TO DISK = 'E:\Backups\QhatuPeru_FullBackup.bak'
WITH INIT, FORMAT;
GO

-- Ejecutar backups de log (se ejecuta cada hora manualmente o via job)
BACKUP LOG QhatuPeru
TO DISK = 'E:\Backups\QhatuPeru_LogBackup1.trn'
WITH INIT;
GO









-- Ejemplo de restauración punto en el tiempo
-- Paso 1: Restauramos el backup completo con NORECOVERY y usando WITH MOVE para evitar errores
RESTORE DATABASE QhatuPeru
FROM DISK = 'E:\Backups\QhatuPeru_FullBackup.bak'
WITH
    MOVE 'QhatuPeru' TO 'E:\SQLData\QhatuPeru.mdf',
    MOVE 'QhatuPeru_log' TO 'E:\SQLLogs\QhatuPeru_log.ldf',
    NORECOVERY;
GO

-- Paso 2: Restaurar los logs de transacciones hasta la fecha deseada con STOPAT y RECOVERY
RESTORE LOG QhatuPeru
FROM DISK = 'E:\Backups\QhatuPeru_LogBackup1.trn'
WITH STOPAT = '2025-11-15T14:30:00', RECOVERY;
GO
```

RESULTADO



Archivos modificados con los nombres respectivos

	QhatuPeru_FullBackup_	20/11/2025 9:52	Carpeta de archivos	
	QhatuPeru_DiffBackup_20251120_103713....	20/11/2025 10:37	Archivo BAK	51 KB
	QhatuPeru_DiffBackup_20251120_104156....	20/11/2025 10:41	Archivo BAK	51 KB
	QhatuPeru_FullBackup.bak	20/11/2025 22:21	Archivo BAK	4,249 KB
	QhatuPeru_FullBackup_20251120_100725....	20/11/2025 10:07	Archivo BAK	507 KB
	QhatuPeru_FullBackup_20251120_101648....	20/11/2025 10:16	Archivo BAK	507 KB
	QhatuPeru_FullBackup_20251120_103037....	20/11/2025 10:30	Archivo BAK	507 KB
	QhatuPeru_LogBackup1.trn	20/11/2025 22:21	Archivo TRN	8 KB

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.

Proyecto 7: Mantenimiento y verificación de copias (base de datos QhatuPeru)

1. Enunciado del ejercicio

Implementar un script que automatice la verificación periódica de backups (RESTORE VERIFYONLY), registre el resultado en una tabla de auditoría y alerte (insertar fila con estado) cuando falle la verificación.

SCRIPT DE LA SOLUCION


```

-- Paso 1: Crear tabla de auditoría para registro de verificaciones
/ IF OBJECT_ID('dbo.BackupVerificationLog', 'U') IS NOT NULL
    DROP TABLE dbo.BackupVerificationLog;
GO

/ CREATE TABLE dbo.BackupVerificationLog (
    Id INT IDENTITY(1,1) PRIMARY KEY,
    BackupFile NVARCHAR(260),
    VerificationDate DATETIME DEFAULT GETDATE(),
    Status NVARCHAR(50),
    Message NVARCHAR(1000) NULL
);
GO

-- Paso 2: Script para verificar backups y registrar resultados

/ DECLARE @BackupPath NVARCHAR(260) = 'E:\Backups\';
DECLARE @BackupFile NVARCHAR(260);
DECLARE @Cmd NVARCHAR(4000);
DECLARE @ErrorMsg NVARCHAR(1000);
DECLARE @Status NVARCHAR(50);

-- Tabla temporal para listar los archivos de backup
/ CREATE TABLE #BackupFiles (
    BackupFile NVARCHAR(260)
);

-- Insertar manualmente los archivos que quieres verificar o implementa un método para listar archivos
/ INSERT INTO #BackupFiles VALUES
('E:\Backups\QhatuPeru_FullBackup.bak'),
('E:\Backups\QhatuPeru_LogBackup1.trn');
-- Agrega más rutas de archivos según tus backups reales

/ DECLARE cur_backup CURSOR FOR
    SELECT BackupFile FROM #BackupFiles;

OPEN cur_backup;
FETCH NEXT FROM cur_backup INTO @BackupFile;

/ WHILE @@FETCH_STATUS = 0
/ BEGIN
    BEGIN TRY
        RESTORE VERIFYONLY FROM DISK = @BackupFile;
        SET @Status = 'Éxito';
        SET @ErrorMsg = NULL;
    END TRY
    BEGIN CATCH
        SET @Status = 'Falla';
        SET @ErrorMsg = ERROR_MESSAGE();
    END CATCH;

    -- Insertar el resultado en tabla de auditoría
/ INSERT INTO dbo.BackupVerificationLog (BackupFile, Status, Message)
VALUES (@BackupFile, @Status, @ErrorMsg);

    FETCH NEXT FROM cur_backup INTO @BackupFile;
END

CLOSE cur_backup;
DEALLOCATE cur_backup;

/ DROP TABLE #BackupFiles;
GO

-- Consulta simple para revisar el log de verificaciones
SELECT * FROM dbo.BackupVerificationLog ORDER BY VerificationDate DESC;

```

RESULTADO

100 %					
<div> <div>✖ 4</div> <div>⚠ 0</div> <div>↑</div> <div>↓</div> </div>					
<div> <div>Resultados</div> <div>Mensajes</div> </div>					
	Id	BackupFile	VerificationDate	Status	Message
1	4	E:\Backups\QhatuPeru_LogBackup1.tm	2025-11-20 22:30:49.700	Éxito	NULL
2	3	E:\Backups\QhatuPeru_FullBackup.bak	2025-11-20 22:30:49.683	Éxito	NULL
3	2	E:\Backups\QhatuPeru_LogBackup1.tm	2025-11-20 22:30:49.437	Éxito	NULL
4	1	E:\Backups\QhatuPeru_FullBackup.bak	2025-11-20 22:30:49.420	Éxito	NULL

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.

Proyecto 8: Restauraciones en distintos escenarios (corrupción parcial, hardware, pérdida del log)

2. Enunciado del ejercicio

Diseñar y ejecutar procedimientos de restauración para 3 escenarios: (A) archivo de datos corrupto (DB en el mismo servidor), (B) pérdida del archivo de log, (C) restauración a servidor diferente (migración).

SCRIPT DE LA SOLUCION

```

--CASO A
-- Poner la base de datos en modo SINGLE_USER y expulsar usuarios conectados
ALTER DATABASE QhatuPeru SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
GO

-- Ejecutar restauración con WITH REPLACE y WITH MOVE
RESTORE DATABASE QhatuPeru
FROM DISK = 'E:\Backups\QhatuPeru_FullBackup.bak'
WITH REPLACE,
MOVE 'QhatuPeru' TO 'D:\SQLData\QhatuPeru.mdf',
MOVE 'QhatuPeru_log' TO 'D:\SQLLogs\QhatuPeru_log.ldf';
GO

-- Volver a modo MULTI_USER para que otros usuarios puedan conectarse
ALTER DATABASE QhatuPeru SET MULTI_USER;
GO

-- CASO B
-- Poner base de datos en modo EMERGENCY para permitir reparaciones
ALTER DATABASE QhatuPeru SET EMERGENCY;
GO

-- Permitir modo de usuario único para recuperación
ALTER DATABASE QhatuPeru SET SINGLE_USER WITH ROLLBACK IMMEDIATE;
GO

-- Ejecutar reparación de base de datos, salvando datos accesibles
DBCC CHECKDB ('QhatuPeru', REPAIR_ALLOW_DATA_LOSS);
GO

-- Volver a modo multiusuario
ALTER DATABASE QhatuPeru SET MULTI_USER;
GO

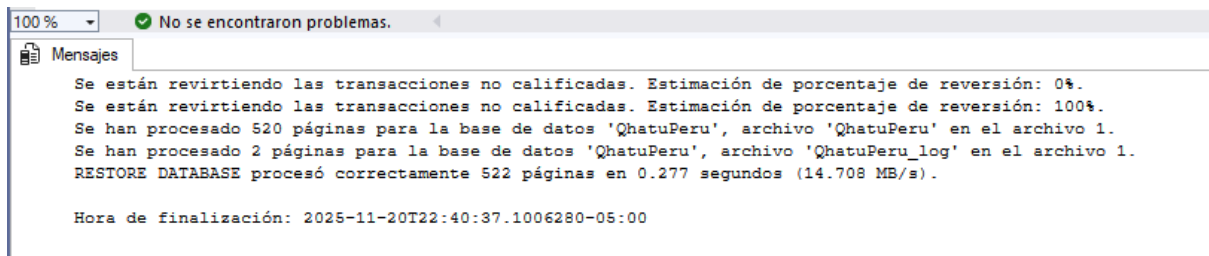
--Caso C
-- Paso 1: Obtener lista de archivos cuando se restaura en otra máquina
RESTORE FILELISTONLY
FROM DISK = 'E:\Backups\QhatuPeru_FullBackup.bak';

-- Paso 2: Restaurar en el servidor destino moviendo archivos a rutas válidas allí
RESTORE DATABASE QhatuPeru
FROM DISK = 'E:\Backups\QhatuPeru_FullBackup.bak'
WITH REPLACE,
MOVE 'QhatuPeru' TO 'D:\SQLData\QhatuPeru.mdf',
MOVE 'QhatuPeru_log' TO 'D:\SQLLogs\QhatuPeru_log.ldf';

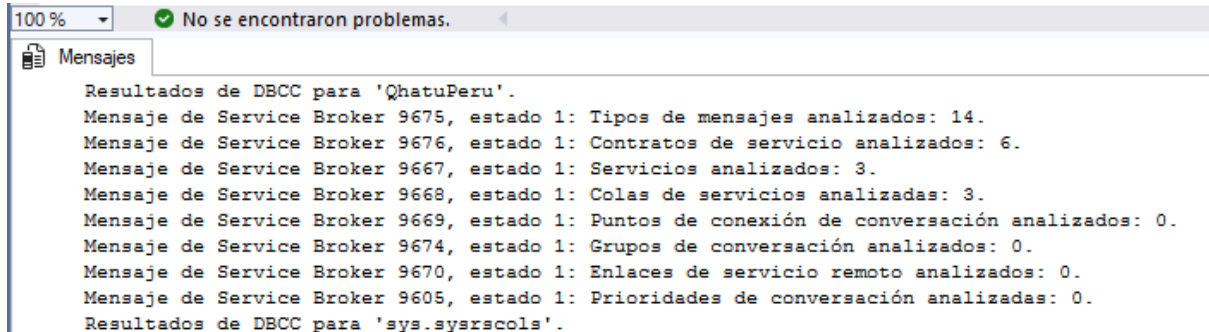
```

RESULTADO

Caso A



Caso B



Caso C

100 % No se encontraron problemas. Línea: 35 Carácter: 1 TABULACIONES CRLF														
Mensajes														
LogicalName	PhysicalName	Type	FileGroupName	Size	MaxSize	Field	CreateLSN	DropLSN	Uniqueld	ReadOnlyLSN	ReadWriteLSN	BackupSizeInBytes	SourceBlockSize	File
1	QhatuPeru	D	PRIMARY	8388608	35184372080640	1	0	0	1914E1CA-A467-4CBA-97DB-3B09683CD2B1	0	0	4128768	4096	1
2	QhatuPeru_log	L	NULL	8388608	2199023255552	2	0	0	A712FCFA-BB37-4077-BB57-DC02CFA1B9F7	0	0	0	4096	0

100 % No se encontraron problemas. Línea: 35 Carácter: 1 TABULACIONES CRLF														
Resultados														
DropLSN	Uniqueld	ReadOnlyLSN	ReadWriteLSN	BackupSizeInBytes	SourceBlockSize	FileGroupID	LogGroupGUID	DifferentialBaseLSN	DifferentialBaseGUID	IsReadOnly	IsPresent	TDEThumbprint	SnapshotLH	
1	0	1914E1CA-A467-4CBA-97DB-3B09683CD2B1	0	0	4128768	4096	1	NULL	0	00000000-0000-0000-0000-000000000000	0	1	NULL	NULL
2	0	A712FCFA-BB37-4077-BB57-DC02CFA1B9F7	0	0	0	4096	0	NULL	0	00000000-0000-0000-0000-000000000000	0	1	NULL	NULL

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.

Proyecto 9: Uso práctico de BACKUP DATABASE y RESTORE DATABASE con MOVE (cambio de archivos)

1. Enunciado del ejercicio

Simular un escenario donde necesitas mover los archivos físicos (.mdf/.ldf) a otra unidad (p.ej. por falta de espacio): hacer backup y restaurar la base en la nueva ruta usando RESTORE ... WITH MOVE.

SCRIPT DE LA SOLUCION

```
-- Paso 1: Realizar el respaldo completo de la base de datos
BACKUP DATABASE QhatuPeru
TO DISK = 'E:\Backups\QhatuPeru.bak'
WITH FORMAT, INIT, NAME = 'Backup completo de QhatuPeru';

-- Paso 2: Obtener las rutas actuales de los archivos físicos
RESTORE FILELISTONLY
FROM DISK = 'E:\Backups\QhatuPeru.bak';

-- Paso 3: Restaurar la base de datos en una nueva ruta (ejemplo: D:\Data\)
RESTORE DATABASE QhatuPeru_Movida
FROM DISK = 'E:\Backups\QhatuPeru.bak'
WITH MOVE 'QhatuPeru' TO 'D:\Data\QhatuPeru_Movida.mdf',
      MOVE 'QhatuPeru_log' TO 'D:\Data\QhatuPeru_Movida.ldf',
      RECOVERY, REPLACE;
```

RESULTADO

100 %															No se encontraron problemas.															Línea: 1															Carácter: 1														
Resultados															Messages																																												
		LogicalName		PhysicalName		Type		FileGroupName		Size		MaxSize		Field		CreateLSN		DropLSN		Uniqueld		ReadOnlyLSN		ReadWriteLSN		BackupSizeInBytes		SourceBlockSize		FileGroupId																													
1		QhatuPeru		D:\SQLData\QhatuPeru.mdf		D		PRIMARY		8388608		35184372080640		1		0		0		1914E1CA-A467-4CBA-97DB-3B09683CD2B1		0		0		4128768		4096		1																													
2		QhatuPeru_log		D:\SQLLogs\QhatuPeru_log.ldf		L		NULL		8388608		2199023255552		2		0		0		A712FCFA-BB37-4077-BB57-DC02CFA1B9F7		0		0		0		4096		0																													

100 %															No se encontraron problemas.															Línea: 1															Carácter: 1															SPC															CRLF														
Resultados															Messages																																																																										
		DropLSN		Uniqueld		ReadOnlyLSN		ReadWriteLSN		BackupSizeInBytes		SourceBlockSize		FileGroupId		LogGroupGUID		DifferentialBaseLSN		DifferentialBaseGUID		IsReadOnly		IsPresent		TDEThumbprint		SnapshotURL																																																													
1		0		1914E1CA-A467-4CBA-97DB-3B09683CD2B1		0		0		4128768		4096		1		NULL		39000000112000001		EAC08370-5908-44B2-8135-88891A199CCE		0		1		NULL		NULL																																																													
2		0		A712FCFA-BB37-4077-BB57-DC02CFA1B9F7		0		0		0		4096		0		NULL		0		00000000-0000-0000-0000-000000000000		0		1		NULL		NULL																																																													

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.

Proyecto 10: Verificación y mantenimiento de backups: automatización y chequeos

2. Enunciado del ejercicio

Crear un job que: (a) ejecute RESTORE VERIFYONLY para todos los archivos .bak en la carpeta de backups, (b) compare con msdb.backupset y (c) envíe un resumen (se inserta en tabla BackupAudit).

SCRIPT DE LA SOLUCION

```
-- Paso 1: Crear tabla de auditoria
USE master;
GO
IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'BackupAudit')
BEGIN
    CREATE TABLE BackupAudit (
        Id INT IDENTITY(1,1) PRIMARY KEY,
        BackupFile NVARCHAR(260),
        DatabaseName NVARCHAR(128),
        BackupStart DATETIME,
        BackupFinish DATETIME,
        VerifyStatus NVARCHAR(50),
        AuditDate DATETIME DEFAULT GETDATE()
    );
END
GO

-- Paso 2: Script que será usado por el Job
DECLARE @BackupFile NVARCHAR(260);
DECLARE @Command NVARCHAR(MAX);
DECLARE @DatabaseName NVARCHAR(128);
DECLARE @BackupStart DATETIME;
DECLARE @BackupFinish DATETIME;

DECLARE backup_cursor CURSOR FOR
SELECT physical_device_name
FROM msdb.dbo.backupmediafamily
WHERE physical_device_name LIKE 'E:\Backups\%.bak';

OPEN backup_cursor;
FETCH NEXT FROM backup_cursor INTO @BackupFile;

WHILE @@FETCH_STATUS = 0
BEGIN
    BEGIN TRY
        -- Verificación del backup
        SET @Command = 'RESTORE VERIFYONLY FROM DISK = N''' + @BackupFile + ''';
        EXEC(@Command);

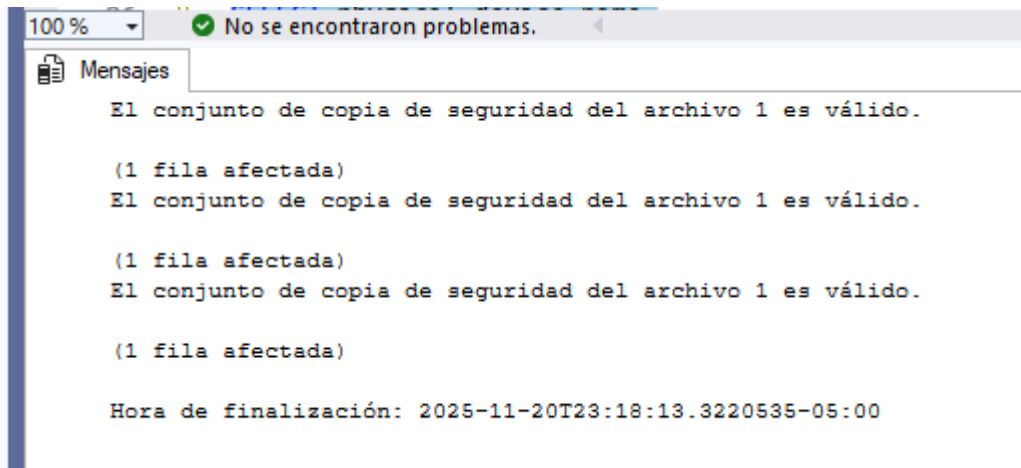
        -- Obtener metadatos del backup
        SELECT TOP 1
            @DatabaseName = b.database_name,
            @BackupStart = b.backup_start_date,
            @BackupFinish = b.backup_finish_date
        FROM msdb.dbo.backupset b
        JOIN msdb.dbo.backupmediafamily m ON b.media_set_id = m.media_set_id
        WHERE m.physical_device_name = @BackupFile
        ORDER BY b.backup_finish_date DESC;
    END TRY
    BEGIN CATCH
        -- Manejo de errores
    END CATCH
END
```

```

49
50      -- Insertar resultado exitoso
51      INSERT INTO master.dbo.BackupAudit (BackupFile, DatabaseName, BackupStart, BackupFinish, VerifyStatus)
52      VALUES (@BackupFile, @DatabaseName, @BackupStart, @BackupFinish, 'VERIFIED');
53  END TRY
54  BEGIN CATCH
55      -- Insertar resultado fallido
56      INSERT INTO master.dbo.BackupAudit (BackupFile, DatabaseName, BackupStart, BackupFinish, VerifyStatus)
57      VALUES (@BackupFile, NULL, NULL, NULL, 'FAILED');
58  END CATCH;
59
60      FETCH NEXT FROM backup_cursor INTO @BackupFile;
61  END
62
63  CLOSE backup_cursor;
64  DEALLOCATE backup_cursor;
65

```

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.

Proyecto 11: Plan de mantenimiento: limpiezas y retención usando msdb

1. Enunciado del ejercicio

Crear un plan de mantenimiento básico (mediante scripts) que elimine historiales de backup antiguos (sp_delete_backuphistory), mantenga registro de backup y limpie archivos físicos con más de X días.

SCRIPT

DE LA SOLUCION

```
-- Habilitar el uso de xp_cmdshell
EXEC sp_configure 'show advanced options', 1;
RECONFIGURE;

EXEC sp_configure 'xp_cmdshell', 1;
RECONFIGURE;

-- Paso 1: Eliminar historial antiguo de backups en msdb (por ejemplo, mayor a 30 días)
DECLARE @FechaLimite DATETIME = DATEADD(DAY, -30, GETDATE());
EXEC msdb.dbo.sp_delete_backuphistory @FechaLimite;

-- Paso 2: Registrar los backups actuales en una tabla de auditoría
USE master;
IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'BackupRetentionLog')
BEGIN
    CREATE TABLE BackupRetentionLog (
        Id INT IDENTITY(1,1) PRIMARY KEY,
        BackupFile NVARCHAR(260),
        FileDate DATETIME,
        ActionTaken NVARCHAR(50),
        AuditDate DATETIME DEFAULT GETDATE()
    );
END

-- Paso 3: Eliminar archivos físicos .bak con más de X días (ejemplo: 30 días)
DECLARE @cmd NVARCHAR(1000);
CREATE TABLE #Archivos (Linea NVARCHAR(4000));

-- Listar archivos con fecha
INSERT INTO #Archivos
EXEC xp_cmdshell 'forfiles /p "E:\Backups" /s /m *.bak /d -30 /c "cmd /c echo @path @fdate"';

-- Procesar y eliminar
DECLARE @linea NVARCHAR(4000), @archivo NVARCHAR(260), @fecha NVARCHAR(50);
DECLARE cursor_archivos CURSOR FOR SELECT Linea FROM #Archivos WHERE Linea IS NOT NULL;

OPEN cursor_archivos;
FETCH NEXT FROM cursor_archivos INTO @linea;

WHILE @@FETCH_STATUS = 0
BEGIN
    IF CHARINDEX('E:\Backups\', @linea) > 0
    BEGIN
        SET @archivo = LEFT(@linea, CHARINDEX(' ', @linea) - 1);
        SET @fecha = RIGHT(@linea, LEN(@linea) - CHARINDEX(' ', @linea));

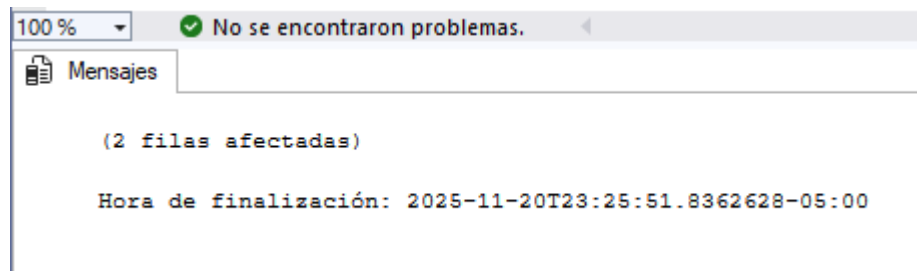
        -- Eliminar archivo
        SET @cmd = 'del "' + @archivo + '"';
        EXEC xp_cmdshell @cmd;

        -- Registrar acción
        INSERT INTO BackupRetentionLog (BackupFile, FileDate, ActionTaken)
        VALUES (@archivo, TRY_CONVERT(DATETIME, @fecha, 103), 'ELIMINADO');
    END

    FETCH NEXT FROM cursor_archivos INTO @linea;
END

CLOSE cursor_archivos;
DEALLOCATE cursor_archivos;
DROP TABLE #Archivos;
```


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.

Proyecto 12: Uso avanzado de BACKUP DATABASE y RESTORE DATABASE con opciones

2. Enunciado del ejercicio

Enseñar y practicar las opciones avanzadas de BACKUP y RESTORE: `COPY_ONLY`, `FORMAT`, `WITH CHECKSUM`, `WITH RETAIN_DAYS`, `WITH STATS`, `WITH MOVE`, `WITH REPLACE`.

SCRIPT DE LA SOLUCION

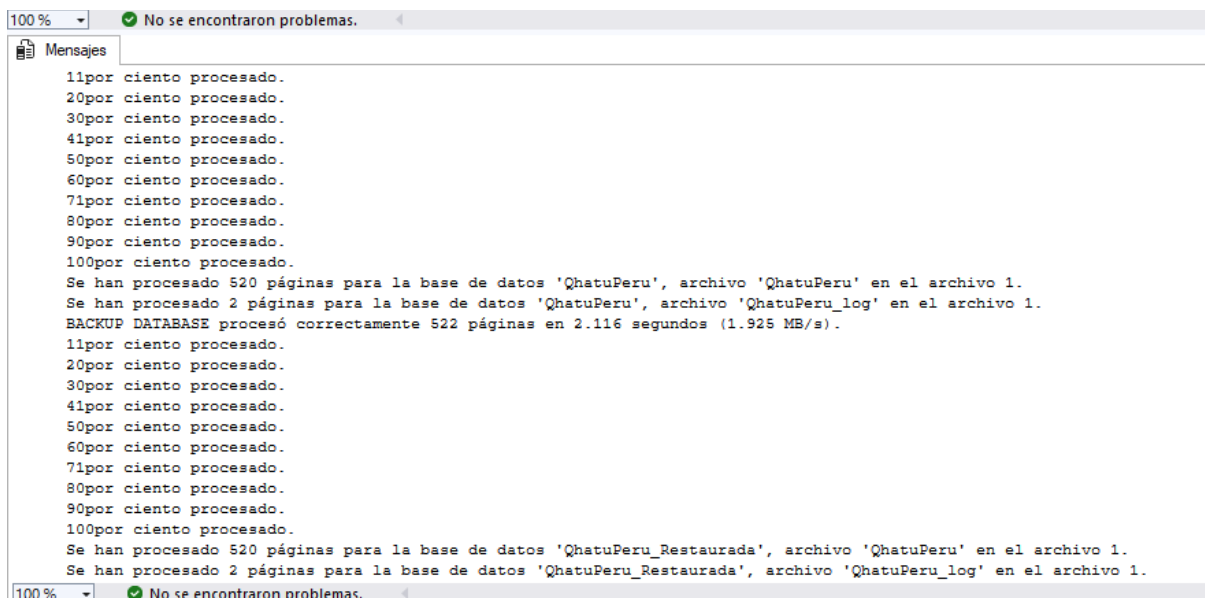
```

-- Backup avanzado con opciones
BACKUP DATABASE QhatuPeru
TO DISK = 'E:\Backups\QhatuPeru_Advanced.bak'
WITH COPY_ONLY,           -- No interfiere con la cadena de backups diferenciales
    FORMAT,               -- Crea un nuevo conjunto de medios
    INIT,                 -- Sobrescribe el archivo si existe
    CHECKSUM,             -- Verifica integridad durante el backup
    RETAINDAYS = 7,       -- Previene eliminación por mantenimiento durante 7 días
    STATS = 10,           -- Muestra progreso cada 10%
    NAME = 'Backup avanzado de QhatuPeru';

-- Restauración avanzada con MOVE y REPLACE
RESTORE DATABASE QhatuPeru_Restaurada
FROM DISK = 'E:\Backups\QhatuPeru_Advanced.bak'
WITH MOVE 'QhatuPeru' TO 'D:\Data\QhatuPeru_Restaurada.mdf',
    MOVE 'QhatuPeru_log' TO 'D:\Data\QhatuPeru_Restaurada.ldf',
    REPLACE,              -- Sobrescribe si ya existe
    STATS = 10;           -- Muestra progreso cada 10%

```

RESULTADO



100 % No se encontraron problemas.

Mensajes

```

11por ciento procesado.
20por ciento procesado.
30por ciento procesado.
41por ciento procesado.
50por ciento procesado.
60por ciento procesado.
71por ciento procesado.
80por ciento procesado.
90por ciento procesado.
100por ciento procesado.
Se han procesado 520 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru' en el archivo 1.
Se han procesado 2 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru_log' en el archivo 1.
BACKUP DATABASE procesó correctamente 522 páginas en 2.116 segundos (1.925 MB/s).
11por ciento procesado.
20por ciento procesado.
30por ciento procesado.
41por ciento procesado.
50por ciento procesado.
60por ciento procesado.
71por ciento procesado.
80por ciento procesado.
90por ciento procesado.
100por ciento procesado.
Se han procesado 520 páginas para la base de datos 'QhatuPeru_Restaurada', archivo 'QhatuPeru' en el archivo 1.
Se han procesado 2 páginas para la base de datos 'QhatuPeru_Restaurada', archivo 'QhatuPeru_log' en el archivo 1.

```

100 % No se encontraron problemas.

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.

Proyecto 13: Restauración punto en el tiempo (recuperar una fila borrada accidentalmente)

1. Enunciado del ejercicio

Simular borrado accidental de filas en tabla ARTICULO. Recuperarlas usando restauración punto-en-el-tiempo hacia otra base temporal y extraer los datos perdidos.

SCRIPT DE LA SOLUCION

```
-- Paso 1: Crear respaldo con marca de tiempo antes del borrado
BACKUP DATABASE QhatuPeru
TO DISK = 'E:\Backups\QhatuPeru_PreBorrado.bak'
WITH FORMAT, INIT, NAME = 'Backup antes del borrado';

-- Paso 2: Simular borrado accidental
DELETE FROM QhatuPeru.dbo.ARTICULO
WHERE IdArticulo = 101; -- Ejemplo de fila borrada

-- Paso 3: Restaurar punto en el tiempo en base temporal
-- Obtener fecha exacta del backup
-- (puedes usar RESTORE HEADERONLY si necesitas verificarla)

RESTORE DATABASE QhatuPeru_Recuperacion
FROM DISK = 'E:\Backups\QhatuPeru_PreBorrado.bak'
WITH MOVE 'QhatuPeru' TO 'D:\Data\QhatuPeru_Recuperacion.mdf',
      MOVE 'QhatuPeru_log' TO 'D:\Data\QhatuPeru_Recuperacion.ldf',
      RECOVERY, REPLACE;

-- Paso 4: Extraer la fila perdida
SELECT *
INTO #FilaRecuperada
FROM QhatuPeru_Recuperacion.dbo.ARTICULO
WHERE IdArticulo = 101;

-- Paso 5: Reinsertar en la base original
INSERT INTO QhatuPeru.dbo.ARTICULO
SELECT * FROM #FilaRecuperada;

-- Limpieza
DROP TABLE #FilaRecuperada;
```

RESULTADO

100 % 5 0

Mensajes

Se han procesado 520 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru' en el archivo 1.
Se han procesado 2 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru_log' en el archivo 1.
BACKUP DATABASE procesó correctamente 522 páginas en 2.685 segundos (1.517 MB/s).

100 % 5 0

Resultados Mensajes

	IdArticulo	Nombre	Categoria	Precio	Stock	FechaRegistro
1	101	Laptop Lenovo ThinkPad	Tecnología	3200.00	10	2025-11-20 23:42:48.557

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.

Proyecto 14: Planes de mantenimiento: chequeos de integridad y tareas automáticas

2. Enunciado del ejercicio

Crear un plan de mantenimiento que ejecute semanalmente: DBCC CHECKDB, reconstrucción de índices fragmentados > 30%, actualización de estadísticas y limpieza de archivos de backup antiguos (retención configurable).

SCRIPT DE LA SOLUCION

```

-- Paso 1: Verificación de integridad
DBCC CHECKDB ('QhatuPeru') WITH NO_INFOMSGS, ALL_ERRORMSG;

-- Paso 2: Reconstrucción de índices fragmentados > 30%
DECLARE @sql NVARCHAR(MAX) = '';

SELECT @sql += '
ALTER INDEX [' + i.name + '] ON [' + s.name + '].[' + o.name + '] REBUILD WITH (ONLINE = OFF);
'
FROM sys.dm_db_index_physical_stats(DB_ID('QhatuPeru'), NULL, NULL, NULL, 'LIMITED') AS stats
JOIN sys.indexes i ON stats.object_id = i.object_id AND stats.index_id = i.index_id
JOIN sys.objects o ON i.object_id = o.object_id
JOIN sys.schemas s ON o.schema_id = s.schema_id
WHERE stats.avg_fragmentation_in_percent > 30 AND i.type_desc IN ('CLUSTERED', 'NONCLUSTERED');

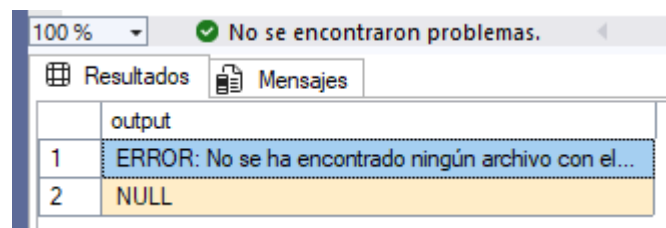
EXEC sp_executesql @sql;

-- Paso 3: Actualización de estadísticas
EXEC sp_MSforeachtable 'UPDATE STATISTICS ? WITH FULLSCAN';

-- Paso 4: Limpieza de archivos .bak antiguos (ejemplo: más de 15 días)
-- Requiere xp_cmdshell habilitado
DECLARE @cmd NVARCHAR(1000);
SET @cmd = 'forfiles /p "E:\Backups" /s /m *.bak /d -15 /c "cmd /c del @path"';
EXEC xp_cmdshell @cmd;

```

RESULTADO



	output
1	ERROR: No se ha encontrado ningún archivo con el...
2	NULL

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.

Proyecto 15: Políticas de retención y compresión de backups (ahorro de espacio)

1. Enunciado del ejercicio

Implementar backups con **compresión** y una política de retención (ejemplo: conservar FULL 4 semanas, DIFF 7 días, LOG 48 horas). Incluir script para eliminar backups fuera de retención.

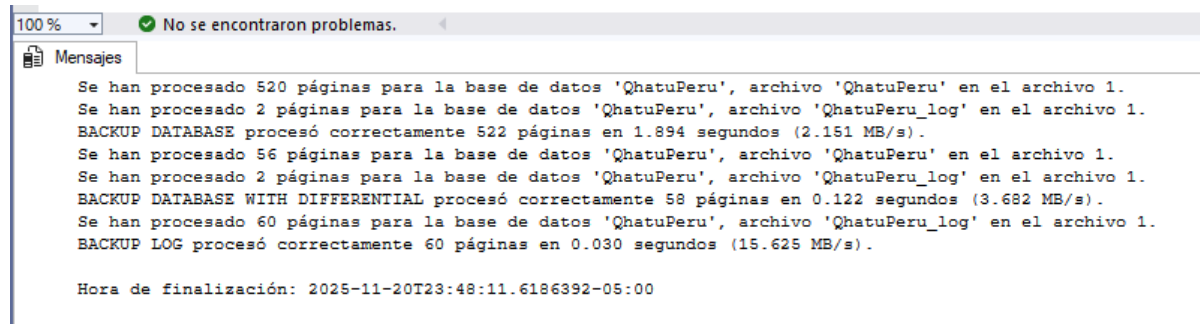
SCRIPT DE LA SOLUCION

```
-- FULL backup con compresión
✓ BACKUP DATABASE QhatuPeru
  TO DISK = 'E:\Backups\QhatuPeru_FULL.bak'
  WITH COMPRESSION, INIT, NAME = 'FULL Backup QhatuPeru';

-- DIFF backup con compresión
✓ BACKUP DATABASE QhatuPeru
  TO DISK = 'E:\Backups\QhatuPeru_DIFF.bak'
  WITH DIFFERENTIAL, COMPRESSION, INIT, NAME = 'DIFF Backup QhatuPeru';

-- LOG backup con compresión
✓ BACKUP LOG QhatuPeru
  TO DISK = 'E:\Backups\QhatuPeru_LOG.trn'
  WITH COMPRESSION, INIT, NAME = 'LOG Backup QhatuPeru';
```

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.

Proyecto 16: Políticas de retención de backups (implementación y reporte)

2. Enunciado del ejercicio

Diseñar e implementar una política de retención: Full (30 días), Diferencial (14 días), Log (7 días).

Producir un reporte automático que muestre cumplimiento de la política.

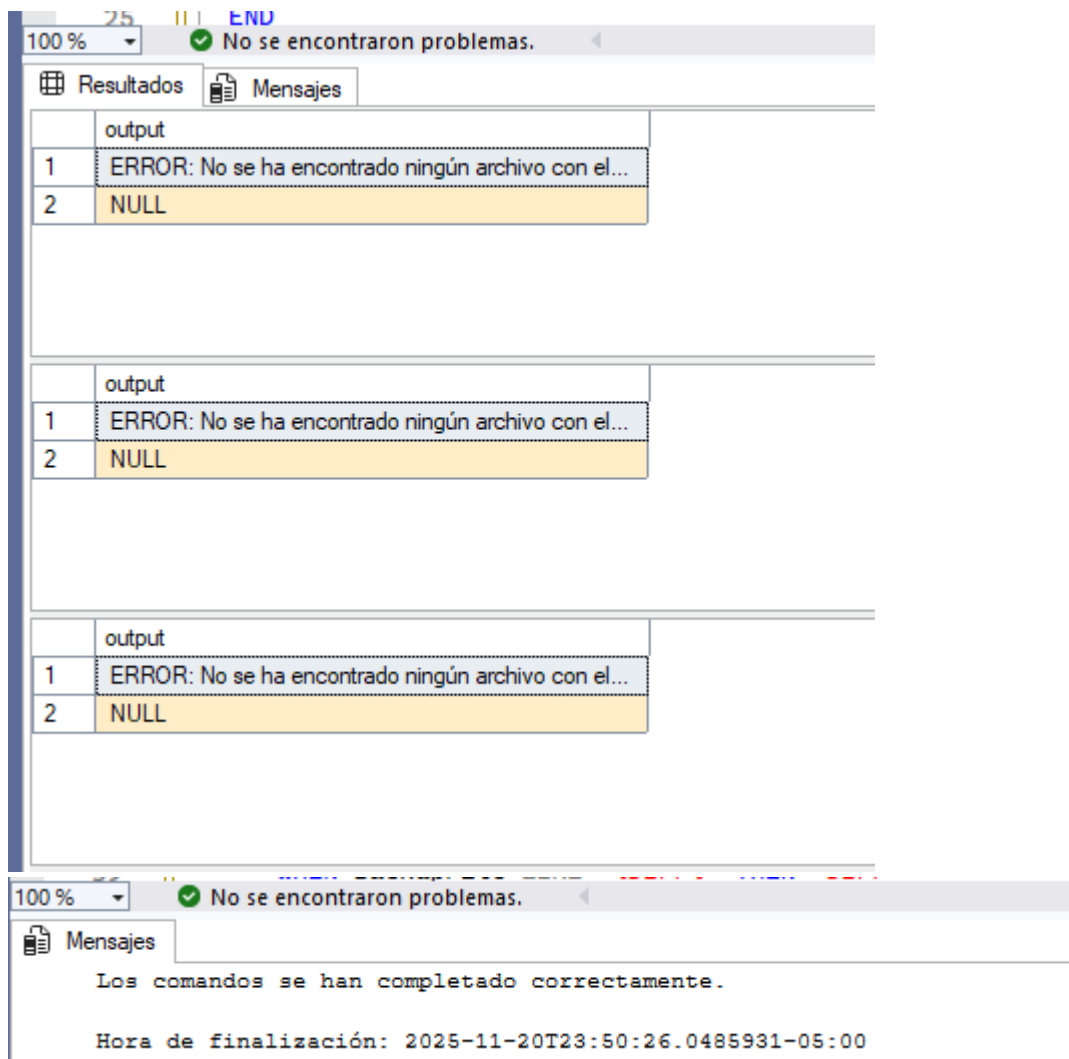
SCRIPT DE LA SOLUCION

```

1  --A
2  -- Requiere xp_cmdshell habilitado
3
4  -- FULL: eliminar si > 30 días
5  EXEC xp_cmdshell 'forfiles /p "E:\Backups" /s /m *FULL.bak /d -30 /c "cmd /c del @path"';
6
7  -- DIFF: eliminar si > 14 días
8  EXEC xp_cmdshell 'forfiles /p "E:\Backups" /s /m *DIFF.bak /d -14 /c "cmd /c del @path"';
9
10 -- LOG: eliminar si > 7 días
11 EXEC xp_cmdshell 'forfiles /p "E:\Backups" /s /m *LOG.trn /d -7 /c "cmd /c del @path"';
12
13 --B
14 USE master;
15 IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'BackupRetentionReport')
16 BEGIN
17     CREATE TABLE BackupRetentionReport (
18         Id INT IDENTITY(1,1) PRIMARY KEY,
19         BackupFile NVARCHAR(260),
20         TipoBackup NVARCHAR(20),
21         FechaCreacion DATETIME,
22         Estado NVARCHAR(20),
23         AuditDate DATETIME DEFAULT GETDATE()
24     );
25 END
26
27 --C
28 -- Limpieza previa
29 DELETE FROM BackupRetentionReport;
30
31 -- Escaneo de archivos actuales
32 INSERT INTO BackupRetentionReport (BackupFile, TipoBackup, FechaCreacion, Estado)
33 EXEC xp_cmdshell 'forfiles /p "E:\Backups" /s /m *.bak /c "cmd /c echo @path @fdate @ftime"';
34
35 -- Clasificación y evaluación
36 UPDATE BackupRetentionReport
37 SET TipoBackup = CASE
38     WHEN BackupFile LIKE '%FULL%' THEN 'FULL'
39     WHEN BackupFile LIKE '%DIFF%' THEN 'DIFF'
40     ELSE 'LOG'
41 END;
42
43 UPDATE BackupRetentionReport
44 SET FechaCreacion = TRY_CONVERT(DATETIME, RIGHT(BackupFile, 17), 103);
45
46 UPDATE BackupRetentionReport
47 SET Estado = CASE
48     WHEN TipoBackup = 'FULL' AND DATEDIFF(DAY, FechaCreacion, GETDATE()) <= 30 THEN 'VIGENTE'
49     WHEN TipoBackup = 'DIFF' AND DATEDIFF(DAY, FechaCreacion, GETDATE()) <= 14 THEN 'VIGENTE'
50     WHEN TipoBackup = 'LOG' AND DATEDIFF(DAY, FechaCreacion, GETDATE()) <= 7 THEN 'VIGENTE'
51     ELSE 'FUERA DE POLÍTICA'
52 END;
53

```

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.

Proyecto 17: Verificación automática post-backup y notificación por correo (simulada)

1. Enunciado del ejercicio

Implementar un procedimiento que: realice backup, verifique (RESTORE VERIFYONLY), registre el resultado y (si falla) envíe una notificación por correo (en laboratorio, registrar la notificación en una tabla o simular envío).

SCRIPT DE LA SOLUCION

```
1  --A
2  -- Auditoría de verificación
3  USE QhatuPeru;
4  IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'BackupVerificationLog')
5  BEGIN
6      CREATE TABLE BackupVerificationLog (
7          Id INT IDENTITY(1,1) PRIMARY KEY,
8          BackupFile NVARCHAR(260),
9          Resultado NVARCHAR(50),
10         Fecha DATETIME DEFAULT GETDATE()
11     );
12 END
13
14 -- Simulación de notificación por correo
15 IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'SimulatedEmailAlerts')
16 BEGIN
17     CREATE TABLE SimulatedEmailAlerts (
18         Id INT IDENTITY(1,1) PRIMARY KEY,
19         Asunto NVARCHAR(100),
20         Mensaje NVARCHAR(MAX),
21         Fecha DATETIME DEFAULT GETDATE()
22     );
23 END
24
25 --B
26 DECLARE @BackupFile NVARCHAR(260) = 'E:\Backups\QhatuPeru_Verificado.bak';
27 DECLARE @Resultado NVARCHAR(50);
28
29 -- Paso 1: Realizar backup
30 BACKUP DATABASE QhatuPeru
31 TO DISK = @BackupFile
32 WITH COMPRESSION, INIT, NAME = 'Backup verificado de QhatuPeru';
33
34 -- Paso 2: Verificar integridad
35 BEGIN TRY
36     RESTORE VERIFYONLY FROM DISK = @BackupFile;
37     SET @Resultado = 'VERIFICADO';
38 END TRY
39 BEGIN CATCH
40     SET @Resultado = 'FALLO';
41 END CATCH;
42
43 -- Paso 3: Registrar resultado
44 INSERT INTO BackupVerificationLog (BackupFile, Resultado)
45 VALUES (@BackupFile, @Resultado);
46
47 -- Paso 4: Simular notificación si falla
48 IF @Resultado = 'FALLÓ'
49 BEGIN
50     INSERT INTO SimulatedEmailAlerts (Asunto, Mensaje)
51     VALUES (
52         'Fallo en verificación de backup',
53         'El archivo ' + @BackupFile + ' no pasó la verificación. Revisar integridad del respaldo.'
54     );
55 END
56
```

RESULTADO

100 % ✓ No se encontraron problemas.

Mensajes

Se han procesado 544 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru' en el archivo 1.
 Se han procesado 2 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru_log' en el archivo 1.
 BACKUP DATABASE procesó correctamente 546 páginas en 1.168 segundos (3.648 MB/s).
 El conjunto de copia de seguridad del archivo 1 es válido.

(1 fila afectada)

Hora de finalización: 2025-11-20T23:58:29.8174578-05:00

100 % ✓ No se encontraron problemas.

Resultados Mensajes

	Id	BackupFile	Resultado	Fecha
1	1	E:\Backups\QhatuPeru_Verificado.bak	VERIFICADO	2025-11-20 23:58:29.800

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.

Proyecto 18: Simulacro de desastre y recuperación completa (DR drill)

2. Enunciado del ejercicio

Realizar un **simulacro de desastre**: simular pérdida total del servidor de base de datos y documentar paso a paso la restauración en un servidor alternativo usando backups disponibles (full + diff + logs). Registrar tiempo de recuperación y lecciones aprendidas.

SCRIPT DE LA SOLUCION

```

-- Paso 1: Restaurar FULL con NORECOVERY
RESTORE DATABASE QhatuPeru_Recuperada
FROM DISK = 'E:\Backups\QhatuPeru_FULL.bak'
WITH MOVE 'QhatuPeru' TO 'D:\Data\QhatuPeru_Recuperada.mdf',
      MOVE 'QhatuPeru_log' TO 'D:\Data\QhatuPeru_Recuperada.ldf',
      NORECOVERY;

-- Paso 2: Restaurar DIFERENCIAL con NORECOVERY
RESTORE DATABASE QhatuPeru_Recuperada
FROM DISK = 'E:\Backups\QhatuPeru_DIFF.bak'
WITH NORECOVERY;

-- Paso 3: Restaurar LOG con RECOVERY
RESTORE LOG QhatuPeru_Recuperada
FROM DISK = 'E:\Backups\QhatuPeru_LOG.trn'
WITH RECOVERY;

```

RESULTADO

100 % ✓ No se encontraron problemas.

Mensajes

Se han procesado 520 páginas para la base de datos 'QhatuPeru_Recuperada', archivo 'QhatuPeru' en el archivo 1.
 Se han procesado 2 páginas para la base de datos 'QhatuPeru_Recuperada', archivo 'QhatuPeru_log' en el archivo 1.
 RESTORE DATABASE procesó correctamente 522 páginas en 0.124 segundos (32.856 MB/s).
 Se han procesado 56 páginas para la base de datos 'QhatuPeru_Recuperada', archivo 'QhatuPeru' en el archivo 1.
 Se han procesado 2 páginas para la base de datos 'QhatuPeru_Recuperada', archivo 'QhatuPeru_log' en el archivo 1.
 RESTORE DATABASE procesó correctamente 58 páginas en 0.119 segundos (3.774 MB/s).
 Se han procesado 0 páginas para la base de datos 'QhatuPeru_Recuperada', archivo 'QhatuPeru' en el archivo 1.
 Se han procesado 60 páginas para la base de datos 'QhatuPeru_Recuperada', archivo 'QhatuPeru_log' en el archivo 1.
 RESTORE LOG procesó correctamente 60 páginas en 0.006 segundos (78.125 MB/s).

Hora de finalización: 2025-11-21T00:01:45.1992077-05:00

100 % ✓ No se encontraron problemas.

Resultados Mensajes

	name	state_desc
1	QhatuPeru_Recuperada	ONLINE

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 alternativo:** 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 alternativo 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.

Proyecto 19: Simulación de recuperación en DR (Disaster Recovery): copiar backup a ubicación remota y restaurar

1. Enunciado del ejercicio

Simular un plan de Disaster Recovery: generar backups y copiar los archivos a una ubicación remota (p.ej. carpeta compartida en otra máquina). En el servidor DR restaurar desde esos backups. Mostrar script de copia y restauración.

SCRIPT DE LA SOLUCION

```
-- FULL backup
BACKUP DATABASE QhatuPeru
TO DISK = 'E:\Backups\QhatuPeru_FULL_DR.bak'
WITH COMPRESSION, INIT, NAME = 'FULL Backup DR';

-- DIFF backup
BACKUP DATABASE QhatuPeru
TO DISK = 'E:\Backups\QhatuPeru_DIFF_DR.bak'
WITH DIFFERENTIAL, COMPRESSION, INIT, NAME = 'DIFF Backup DR';

-- LOG backup
BACKUP LOG QhatuPeru
TO DISK = 'E:\Backups\QhatuPeru_LOG_DR.trn'
WITH COMPRESSION, INIT, NAME = 'LOG Backup DR';

-- Requiere xp_cmdshell habilitado
-- Supongamos que la carpeta remota compartida es \\DRServer\RespaldoDR

EXEC xp_cmdshell 'copy "E:\Backups\QhatuPeru_FULL_DR.bak" "\\DRServer\RespaldoDR\"';
EXEC xp_cmdshell 'copy "E:\Backups\QhatuPeru_DIFF_DR.bak" "\\DRServer\RespaldoDR\"';
EXEC xp_cmdshell 'copy "E:\Backups\QhatuPeru_LOG_DR.trn" "\\DRServer\RespaldoDR\"';
```

```

-- Paso 1: Restaurar FULL con NORECOVERY
RESTORE DATABASE QhatuPeru_DR
FROM DISK = '\\DRServer\RespaldoDR\QhatuPeru_FULL_DR.bak'
WITH MOVE 'QhatuPeru' TO 'E:\DRData\QhatuPeru_DR.mdf',
      MOVE 'QhatuPeru_log' TO 'E:\DRData\QhatuPeru_DR.ldf',
      NORECOVERY;

-- Paso 2: Restaurar DIFERENCIAL con NORECOVERY
RESTORE DATABASE QhatuPeru_DR
FROM DISK = '\\DRServer\RespaldoDR\QhatuPeru_DIFF_DR.bak'
WITH NORECOVERY;

-- Paso 3: Restaurar LOG con RECOVERY
RESTORE LOG QhatuPeru_DR
FROM DISK = '\\DRServer\RespaldoDR\QhatuPeru_LOG_DR.trn'
WITH RECOVERY;

```

RESULTADO

100 % No se encontraron problemas.

Mensajes

Se han procesado 568 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru' en el archivo 1.
 Se han procesado 2 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru_log' en el archivo 1.
 BACKUP DATABASE procesó correctamente 570 páginas en 0.267 segundos (16.663 MB/s).
 Se han procesado 88 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru' en el archivo 1.
 Se han procesado 2 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru_log' en el archivo 1.
 BACKUP DATABASE WITH DIFFERENTIAL procesó correctamente 90 páginas en 0.143 segundos (4.889 MB/s).
 Se han procesado 67 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru_log' en el archivo 1.
 BACKUP LOG procesó correctamente 67 páginas en 0.198 segundos (2.623 MB/s).

Hora de finalización: 2025-11-21T00:06:03.2545854-05:00

100 % No se encontraron problemas.

Resultados Mensajes

	output
1	No se ha encontrado la ruta de acceso de la red.
2	0 archivo(s) copiado(s).
3	NULL

	output
1	No se ha encontrado la ruta de acceso de la red.
2	0 archivo(s) copiado(s).
3	NULL

	output
1	No se ha encontrado la ruta de acceso de la red.
2	0 archivo(s) copiado(s).
3	NULL

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.

Proyecto 20 — Automatización completa: script único que hace full, differential, log y genera reporte

2. Enunciado del ejercicio

Crear un **script maestro** que: detecte el último full, determine si toca hacer full o differential, ejecute backup de acuerdo a política, ejecute backup de log si corresponde, verifique y genere un reporte (inserta en BackupAudit).

SCRIPT DE LA SOLUCION

```
1      USE master;
2      GO
3
4      -- Paso 1: Crear tabla de auditoría si no existe
5      IF NOT EXISTS (SELECT * FROM sys.tables WHERE name = 'BackupAudit')
6      BEGIN
7          CREATE TABLE BackupAudit (
8              Id INT IDENTITY(1,1) PRIMARY KEY,
9              TipoBackup NVARCHAR(20),
10             Archivo NVARCHAR(260),
11             EstadoVerificacion NVARCHAR(50),
12             Fecha DATETIME DEFAULT GETDATE()
13         );
14     END
15
16     -- Paso 2: Variables
17     DECLARE @UltimoFull DATETIME;
18     DECLARE @Hoy DATETIME = GETDATE();
19     DECLARE @Tipo NVARCHAR(20);
20     DECLARE @Archivo NVARCHAR(260);
21     DECLARE @Verificacion NVARCHAR(50);
22
23     -- Paso 3: Detectar último FULL backup
24     SELECT TOP 1 @UltimoFull = backup_finish_date
25     FROM msdb.dbo.backupset
26     WHERE database_name = 'QhatuPeru' AND type = 'D'
27     ORDER BY backup_finish_date DESC;
```



```

29 -- Paso 4: Determinar tipo de backup
30 IF @UltimoFull IS NULL OR DATEDIFF(DAY, @UltimoFull, @Hoy) >= 7
31     SET @Tipo = 'FULL';
32 ELSE
33     SET @Tipo = 'DIFF';
34
35 -- Paso 5: Ejecutar backup según tipo
36 IF @Tipo = 'FULL'
37 BEGIN
38     SET @Archivo = 'E:\Backups\QhatuPeru_FULL_AUTO.bak';
39     BACKUP DATABASE QhatuPeru
40     TO DISK = @Archivo
41     WITH COMPRESSION, INIT, NAME = 'FULL AUTO Backup';
42 END
43 ELSE
44 BEGIN
45     SET @Archivo = 'E:\Backups\QhatuPeru_DIFF_AUTO.bak';
46     BACKUP DATABASE QhatuPeru
47     TO DISK = @Archivo
48     WITH DIFFERENTIAL, COMPRESSION, INIT, NAME = 'DIFF AUTO Backup';
49 END
50
51 -- Paso 6: Ejecutar backup de LOG si aplica
52 IF DATABASEPROPERTYEX('QhatuPeru', 'Recovery') = 'FULL'
53 BEGIN
54     BACKUP LOG QhatuPeru
55     TO DISK = 'E:\Backups\QhatuPeru_LOG_AUTO.trn'
56     WITH COMPRESSION, INIT, NAME = 'LOG AUTO Backup';
57 END
58
59 -- Paso 7: Verificar backup
60 BEGIN TRY
61     RESTORE VERIFYONLY FROM DISK = @Archivo;
62     SET @Verificacion = 'VERIFICADO';
63 END TRY
64 BEGIN CATCH
65     SET @Verificacion = 'FALLÓ';
66 END CATCH;
67
68 -- Paso 8: Insertar en BackupAudit
69 INSERT INTO BackupAudit (TipoBackup, Archivo, EstadoVerificacion)
70 VALUES (@Tipo, @Archivo, @Verificacion);
71
72 SELECT * FROM BackupAudit

```

RESULTADO

100 % 3 0

Mensajes

Se han procesado 88 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru' en el archivo 1.
 Se han procesado 2 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru_log' en el archivo 1.
 BACKUP DATABASE WITH DIFFERENTIAL procesó correctamente 90 páginas en 0.065 segundos (10.757 MB/s).
 Se han procesado 6 páginas para la base de datos 'QhatuPeru', archivo 'QhatuPeru_log' en el archivo 1.
 BACKUP LOG procesó correctamente 6 páginas en 0.024 segundos (1.953 MB/s).
 El conjunto de copia de seguridad del archivo 1 es válido.

Hora de finalización: 2025-11-21T00:09:39.5155417-05:00

100 % 4 0

Resultados Mensajes

	Id	BackupFile	DatabaseName	BackupStart	BackupFinish	VerifyStatus	AuditDate
1	1	E:\Backups\QhatuPeru_FullBackup.bak	QhatuPeru	2025-11-20 22:21:34.000	2025-11-20 22:21:35.000	VERIFIED	2025-11-20 23:18:12.733
2	2	E:\Backups\QhatuPeru.bak	QhatuPeru	2025-11-20 23:04:21.000	2025-11-20 23:04:23.000	VERIFIED	2025-11-20 23:18:13.023
3	3	E:\Backups\QhatuPeru.bak	QhatuPeru	2025-11-20 23:04:21.000	2025-11-20 23:04:23.000	VERIFIED	2025-11-20 23:18:13.307

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.