Tu idea tiene **ventajas y consideraciones importantes**. Aquí te presento un análisis y una solución mejorada que combina lo mejor de ambos enfoques:

**Ventajas de tu propuesta:**

1. **Histórico centralizado**: Permite analizar tendencias y auditorías a largo plazo
2. **Disponibilidad permanente**: Los datos persisten después de reinicios o fallos
3. **Consultas personalizadas**: Posibilidad de generar reportes históricos
4. **Independencia del correo**: Los datos se guardan aunque falle el envío de email

**Consideraciones a tener en cuenta:**

1. Necesidad de mantener y respaldar la BD Monitoreo
2. Posible crecimiento de la tabla según la frecuencia de backups
3. Requiere manejo de retención histórica

**Solución Mejorada (Base de Datos + SP + Job):**

1. **Crear BD y tabla de monitoreo**

CREATE DATABASE [Monitoreo]

ON PRIMARY

(

NAME = N'Monitoreo',

FILENAME = N'H:\Databases\Monitoreo.mdf', -- Cambiar por tu ruta de datos

SIZE = 100MB,

MAXSIZE = UNLIMITED,

FILEGROWTH = 50MB

)

LOG ON

(

NAME = N'Monitoreo\_log',

FILENAME = N'H:\Databases\Monitoreo\_log.ldf', -- Cambiar por tu ruta de logs

SIZE = 50MB,

MAXSIZE = 2GB,

FILEGROWTH = 25MB

);

GO

USE [Monitoreo];

GO

-- Crear tabla (la misma que tenías)

CREATE TABLE Historico\_Backup (

BackupID INT IDENTITY(1,1) PRIMARY KEY,

ServerName NVARCHAR(128),

DatabaseName NVARCHAR(128),

BackupType VARCHAR(10),

BackupSizeMB DECIMAL(10,2),

StartTime DATETIME,

FinishTime DATETIME,

DurationSeconds INT,

Status NVARCHAR(50),

ErrorMessage NVARCHAR(MAX),

BackupFile NVARCHAR(255),

LogDate DATETIME DEFAULT GETDATE()

);

GO

CREATE INDEX IX\_Historico\_Backup\_LogDate ON Historico\_Backup (LogDate);

CREATE INDEX IX\_Historico\_Backup\_DatabaseName ON Historico\_Backup (DatabaseName);

GO

El código integrando tu **dispositivo de backup "Backup\_full"**. Aquí está la versión corregida y completa:

USE [Monitoreo]

GO

CREATE OR ALTER PROCEDURE [dbo].[usp\_BackupAllDatabases]

@BackupType VARCHAR(10) = 'FULL', -- FULL, DIFF, LOG

@EmailRecipients NVARCHAR(MAX) = 'luis.orobio@sisa.com.co',

@Compression BIT = 1,

@VerifyBackup BIT = 1

AS

BEGIN

SET NOCOUNT ON;

DECLARE @HTML NVARCHAR(MAX);

DECLARE @Subject NVARCHAR(255);

DECLARE @ServerName NVARCHAR(128) = @@SERVERNAME;

DECLARE @CurrentDate NVARCHAR(50) = CONVERT(NVARCHAR, GETDATE(), 120);

DECLARE @TotalDBs INT = 0, @SuccessDBs INT = 0, @FailedDBs INT = 0;

DECLARE @BackupPath NVARCHAR(500) = '\\WINVMWAPPRD90\Diario\Laserfiche\Backups\_Full\';

IF UPPER(@BackupType) NOT IN ('FULL', 'DIFF', 'LOG')

BEGIN

RAISERROR('Tipo de backup no válido. Use FULL, DIFF o LOG.', 16, 1);

RETURN;

END

IF OBJECT\_ID('tempdb..#BackupResults') IS NOT NULL

DROP TABLE #BackupResults;

CREATE TABLE #BackupResults (

DatabaseName NVARCHAR(128),

BackupSizeMB DECIMAL(10,2),

StartTime DATETIME,

FinishTime DATETIME,

DurationSeconds INT,

Status NVARCHAR(50),

ErrorMessage NVARCHAR(MAX),

BackupFile NVARCHAR(500)

);

DECLARE @DBList TABLE (DBName NVARCHAR(128), RecoveryModel NVARCHAR(60));

INSERT INTO @DBList

SELECT

name,

recovery\_model\_desc

FROM sys.databases

WHERE state = 0 -- ONLINE

AND name NOT IN ('tempdb') -- Excluir tempdb

AND is\_in\_standby = 0 -- No bases en standby

AND (UPPER(@BackupType) != 'LOG' OR recovery\_model\_desc != 'SIMPLE'); -- Solo BD en FULL recovery para LOG backups

SET @TotalDBs = @@ROWCOUNT;

DECLARE @DBName NVARCHAR(128);

DECLARE @RecoveryModel NVARCHAR(60);

DECLARE @SQL NVARCHAR(MAX);

DECLARE @StartTime DATETIME;

DECLARE @FinishTime DATETIME;

DECLARE @ErrorMsg NVARCHAR(MAX);

DECLARE @BackupFile NVARCHAR(500);

DECLARE DBCursor CURSOR FOR

SELECT DBName, RecoveryModel FROM @DBList;

OPEN DBCursor;

FETCH NEXT FROM DBCursor INTO @DBName, @RecoveryModel;

WHILE @@FETCH\_STATUS = 0

BEGIN

SET @ErrorMsg = NULL;

SET @StartTime = GETDATE();

SET @BackupFile = @BackupPath + @DBName + '\_' +

CONVERT(NVARCHAR(8), GETDATE(), 112) + '\_' +

REPLACE(CONVERT(NVARCHAR(8), GETDATE(), 108), ':', '') +

CASE WHEN UPPER(@BackupType) = 'LOG' THEN '.trn' ELSE '.bak' END;

BEGIN TRY

IF UPPER(@BackupType) = 'DIFF'

SET @SQL = 'BACKUP DATABASE [' + @DBName + '] TO DISK = ''' + @BackupFile + ''' WITH DIFFERENTIAL, ';

ELSE IF UPPER(@BackupType) = 'LOG'

SET @SQL = 'BACKUP LOG [' + @DBName + '] TO DISK = ''' + @BackupFile + ''' WITH ';

ELSE

SET @SQL = 'BACKUP DATABASE [' + @DBName + '] TO DISK = ''' + @BackupFile + ''' WITH ';

SET @SQL = @SQL + 'STATS = 10, CHECKSUM';

IF @Compression = 1

SET @SQL = @SQL + ', COMPRESSION';

PRINT 'Ejecutando: ' + @SQL;

EXEC sp\_executesql @SQL;

IF @VerifyBackup = 1

BEGIN

SET @SQL = 'RESTORE VERIFYONLY FROM DISK = ''' + @BackupFile + '''';

EXEC sp\_executesql @SQL;

END

SET @FinishTime = GETDATE();

INSERT INTO #BackupResults

VALUES (

@DBName,

NULL,

@StartTime,

@FinishTime,

DATEDIFF(SECOND, @StartTime, @FinishTime),

'Exitoso',

NULL,

@BackupFile

);

SET @SuccessDBs = @SuccessDBs + 1;

END TRY

BEGIN CATCH

SET @FinishTime = GETDATE();

SET @ErrorMsg = ERROR\_MESSAGE();

INSERT INTO #BackupResults

VALUES (

@DBName,

0,

@StartTime,

@FinishTime,

DATEDIFF(SECOND, @StartTime, @FinishTime),

'Fallido',

@ErrorMsg,

@BackupFile

);

SET @FailedDBs = @FailedDBs + 1;

END CATCH

FETCH NEXT FROM DBCursor INTO @DBName, @RecoveryModel;

END

CLOSE DBCursor;

DEALLOCATE DBCursor;

UPDATE br

SET br.BackupSizeMB = bs.backup\_size/1048576.0

FROM #BackupResults br

JOIN msdb.dbo.backupset bs ON br.DatabaseName = bs.database\_name

JOIN msdb.dbo.backupmediafamily bmf ON bs.media\_set\_id = bmf.media\_set\_id

WHERE br.Status = 'Exitoso'

AND bs.backup\_finish\_date BETWEEN br.StartTime AND br.FinishTime

AND bmf.physical\_device\_name = br.BackupFile;

INSERT INTO Monitoreo.dbo.Historico\_Backup (

ServerName,

DatabaseName,

BackupType,

BackupSizeMB,

StartTime,

FinishTime,

DurationSeconds,

Status,

ErrorMessage,

BackupFile,

LogDate

)

SELECT

@ServerName,

DatabaseName,

UPPER(@BackupType),

BackupSizeMB,

StartTime,

FinishTime,

DurationSeconds,

Status,

ErrorMessage,

BackupFile,

GETDATE()

FROM #BackupResults;

SET @Subject = 'Resumen Backup ' + UPPER(@BackupType) + ' - ' + @ServerName + ' - ' + @CurrentDate;

SET @HTML =

N'<html><body>

<h2>Resumen de Backup ' + UPPER(@BackupType) + ' - ' + @ServerName + '</h2>

<p><strong>Fecha del reporte:</strong> ' + @CurrentDate + '</p>

<p><strong>Ruta de backup:</strong> ' + @BackupPath + '</p>

<p><strong>Total bases de datos:</strong> ' + CAST(@TotalDBs AS NVARCHAR) + '</p>

<p><strong>Backups exitosos:</strong> <span style="color:green;">' + CAST(@SuccessDBs AS NVARCHAR) + '</span></p>

<p><strong>Backups fallidos:</strong> <span style="color:red;">' + CAST(@FailedDBs AS NVARCHAR) + '</span></p>

<table border="1" cellpadding="5" cellspacing="0">

<tr style="background-color:#f2f2f2;">

<th>Base de Datos</th>

<th>Tamaño (MB)</th>

<th>Inicio</th>

<th>Fin</th>

<th>Duración (s)</th>

<th>Estado</th>

<th>Archivo</th>

</tr>' +

CAST((

SELECT

td = DatabaseName, '',

td = ISNULL(CAST(BackupSizeMB AS NVARCHAR(20)), 'N/A'), '',

td = CONVERT(NVARCHAR(20), StartTime, 120), '',

td = CONVERT(NVARCHAR(20), FinishTime, 120), '',

td = CAST(DurationSeconds AS NVARCHAR(10)), '',

td = CASE WHEN Status = 'Exitoso' THEN

'<span style="color:green;">✓ Exitoso</span>'

ELSE

'<span style="color:red;">✗ Fallido<br/>' + ISNULL(ErrorMessage, '') + '</span>'

END, '',

td = ISNULL(BackupFile, 'N/A')

FROM #BackupResults

ORDER BY Status, DatabaseName

FOR XML PATH('tr'), TYPE

) AS NVARCHAR(MAX)) +

N'</table>

<p>Este es un mensaje automático, por favor no responder.</p>

</body></html>';

IF @EmailRecipients IS NOT NULL AND @EmailRecipients != ''

AND EXISTS (SELECT 1 FROM msdb.dbo.sysmail\_profile WHERE name = 'CuentaOffice365')

BEGIN

BEGIN TRY

EXEC msdb.dbo.sp\_send\_dbmail

@profile\_name = 'CuentaOffice365',

@recipients = @EmailRecipients,

@subject = @Subject,

@body = @HTML,

@body\_format = 'HTML',

@importance = 'HIGH';

END TRY

BEGIN CATCH

PRINT 'Error enviando correo: ' + ERROR\_MESSAGE();

END CATCH

END

SELECT \* FROM #BackupResults ORDER BY Status, DatabaseName;

DROP TABLE #BackupResults;

END

GO

-- Ejemplo 1: Backup completo con configuración básica

EXEC [dbo].[usp\_BackupAllDatabases]

@BackupType = 'FULL',

@EmailRecipients = 'admin@empresa.com';

-- Ejemplo 2: Backup diferencial sin compresión

EXEC [dbo].[usp\_BackupAllDatabases]

@BackupType = 'DIFF',

@Compression = 0,

@EmailRecipients = 'dba@empresa.com';

-- Ejemplo 3: Backup de logs con verificación

EXEC [dbo].[usp\_BackupAllDatabases]

@BackupType = 'LOG',

@VerifyBackup = 1,

@EmailRecipients = 'equipo.it@empresa.com';

-- Consultar backups fallidos en últimos 7 días

SELECT \* FROM Monitoreo.dbo.Historico\_Backup

WHERE Status = 'Fallido'

AND LogDate >= DATEADD(DAY, -7, GETDATE());

-- Eliminar registros mayores a 1 año

DELETE FROM Monitoreo.dbo.Historico\_Backup

WHERE LogDate < DATEADD(YEAR, -1, GETDATE());