



UNIVERSIDAD CENTROAMERICANA JOSE SIMEON CAÑAS

PROYECTO DE CATEDRA – ADMINISTRACION DE BASES DE DATOS

Informe técnico

Sistema de Gestión de Donaciones para ONG – SQL Server 2022

Estudiantes:

Salguero Mojica, Ana Stephanie

Carné: 00007324

Girón Araujo, Fiorella María

Carné: 00149524

Iraheta García, Valeria Lourdes

Carné: 00002024

Andrea Elizabeth Monterroza Rodriguez

Carné: 00004624

Fecha de entrega del informe: 27 de noviembre del 2025

Descripción del sistema.

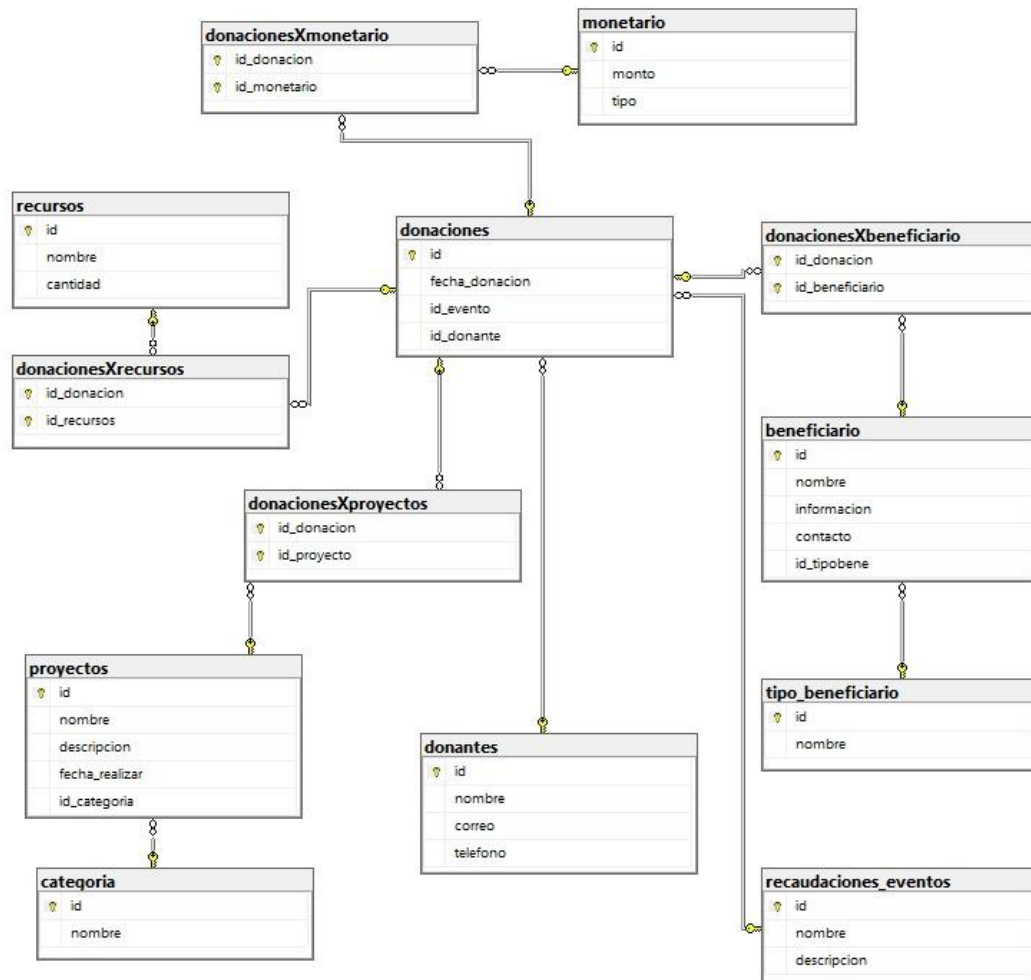
El proyecto **Donaciones_ONG** consiste en el diseño, construcción y administración de una base de datos destinada a gestionar de manera eficiente el proceso de donaciones dentro de una organización no gubernamental. Su finalidad es centralizar y organizar la información relacionada con los donantes, beneficiarios, recursos entregados, aportes monetarios, proyectos apoyados y eventos de recaudación, permitiendo que la ONG mantenga un control claro, seguro y estructurado de todas sus operaciones.

El sistema tiene como finalidad administrar la información relacionada a:

- **Donantes y sus contribuciones:** Personas que realizan aportes monetarios o en especie
- **Proyectos y categorías:** Iniciativas apoyadas por la ONG, categorizadas según su naturaleza (educación, salud, infraestructura, etc.).
- **Beneficiarios y tipos de beneficiarios:** Personas u organizaciones que reciben donaciones. Incluye su clasificación por tipo de beneficiario.
- **Eventos de recaudación:** Actividades donde se reciben donaciones.
- **Seguimiento de donaciones por proyectos y beneficiarios.**

Modelo de datos.

A continuación, se presenta el modelo Entidad-Relación (ER) del sistema implementado:



Descripción de las entidades:

1. Tabla: donantes

Registra la información básica de cada donante individual

| Campo | Descripción | Tipo | Restricciones |
|----------|--|---------------|------------------------|
| id | Identificador único del donante | INT | PK, IDENTITY, NOT NULL |
| nombre | Nombre del donante | VARCHAR (50) | NULL permitido |
| correo | Correo electrónico del donante | VARCHAR (100) | NULL permitido |
| teléfono | Número telefónico en formato "0000-0000" | VARCHAR (9) | NULL permitido |

2. Tabla: recaudaciones_eventos

Almacena los eventos en los que se realizan actividades de recaudación.

| Campo | Descripción | Tipo | Restricciones |
|-------|-------------|------|---------------|
|-------|-------------|------|---------------|

| | | | |
|-------------|---|---------------|------------------------|
| id | Identificador único del evento de recaudación | INT | PK, IDENTITY, NOT NULL |
| nombre | Nombre del evento | VARCHAR (50) | NULL permitido |
| descripción | Breve descripción del evento | VARCHAR (200) | NULL permitido |

3. Tabla: donaciones

Contiene cada donación registrada, ya sea monetaria o en especie.

| Campo | Descripción | Tipo | Restricciones |
|----------------|--|------|---|
| id | Identificador único de la donación | INT | PK, IDENTITY, NOT NULL |
| fecha_donacion | Fecha en que se realizó la donación | DATE | NULL permitido |
| id_evento | Evento asociado a la donación (opcional) | INT | FK [recaudaciones_eventos (id)], NULL permitido |
| id_donante | Donante que realizó la donación | INT | FK [donantes (id)], NOT NULL |

Notas: Se permiten donaciones fuera de eventos (id_evento NULL)

4. Tabla: recursos

Almacena recursos físicos que pueden ser donados.

| Campo | Descripción | Tipo | Restricciones |
|----------|---------------------------------|--------------|------------------------|
| id | Identificador único del recurso | INT | PK, IDENTITY, NOT NULL |
| nombre | Nombre del recurso donado | VARCHAR (50) | NULL permitido |
| cantidad | Cantidad disponible donada | INT | NOT NULL |

5. Tabla: donacionesXrecursos

Relación N:M entre donaciones y los recursos.

| Campo | Descripción | Tipo | Restricciones |
|-------------|-------------------|------|---------------------------|
| id_donacion | Donación asociada | INT | PK, FK [donaciones (id)], |
| id_recursos | Recurso donado | INT | PK, FK [recursos (id)] |

6. Tabla: monetario

Almacena detalles de montos donados en dinero.

| Campo | Descripción | Tipo | Restricciones |
|-------|---|----------------|------------------------|
| id | Identificador único del aporte monetario | INT | PK, IDENTITY, NOT NULL |
| monto | Monto entregado en dolares | DECIMAL (18,2) | NULL permitido |
| tipo | Tipo de entrega: “efectivo” o “transferencia” | VARCHAR (20) | NULL permitido |

7. Tabla: donacionesXmonetario

Relación puente para conectar una donación con aportes monetarios

| Campo | Descripción | Tipo | Restricciones |
|--------------|---------------------------|------|--------------------------|
| id_donacion | Donación asociada | INT | PK, FK [donaciones (id)] |
| id_moentario | Aporte monetario asociado | INT | PK, FK [monetario (id)] |

8. Tabla: categoría

Clasifica los proyectos según el tipo de actividad (educación, salud, infraestructura, etc)

| Campo | Descripción | Tipo | Restricciones |
|--------|-------------------------------------|--------------|------------------------|
| id | Identificador único de la categoría | INT | PK, IDENTITY, NOT NULL |
| nombre | Nombre de la categoría | VARCHAR (50) | NULL permitido |

9. Tabla: proyectos.

Proyectos de la ONG que pueden ser financiados o apoyados mediante donaciones.

| Campo | Descripción | Tipo | Restricciones |
|----------------|--|---------------|--------------------------------|
| id | Identificador único del proyecto | INT | PK, IDENTITY, NOT NULL |
| nombre | Nombre del proyecto | VARCHAR (100) | NULL permitido |
| descripcion | Breve descripción general del proyecto | VARCHAR (200) | NULL permitido |
| fecha_realizar | Fecha estimada o programada de realización | DATETIME | NULL permitido |
| id_categoria | Categoría del proyecto | INT | FK [categoria (id)], NOT NULL |

10. Tabla: donacionesXproyectos

Relación N:M entre proyectos y donaciones que los financian.

| Campo | Descripción | Tipo | Restricciones |
|-------------|----------------------|------|--------------------------|
| id_donacion | Donación asociada | INT | PK, FK [donaciones (id)] |
| id_proyecto | Proyecto beneficiado | INT | PK, FK [proyectos (id)] |

11. Tabla: tipo_beneficiario

Tipos de beneficiarios que pueden recibir ayuda (familias, comunidades, individuos, etc.)

| Campo | Descripción | Tipo | Restricciones |
|--------|--|--------------|------------------------|
| id | Identificador único del tipo de beneficiario | INT | PK, IDENTITY, NOT NULL |
| nombre | Nombre del tipo | VARCHAR (50) | NULL permitido |

12. Tabla: beneficiario

Información de los beneficiarios que reciben ayuda por los proyectos o donaciones.

| Campo | Descripción | Tipo | Restricciones |
|-------------|--|---------------|-----------------------------|
| id | Identificador único del beneficiario | INT | PK, IDENTITY, NOT NULL |
| nombre | Nombre completo del beneficiario | VARCHAR (50) | NOT NULL |
| informacion | Información adicional | VARCHAR (200) | NULL permitido |
| contacto | Número telefónico en formato "0000-0000" | VARCHAR (9) | NULL permitido |
| Id_tipobene | Tipo de beneficiario | INT | FK [tipo_beneficiario (id)] |

13. Tabla: donacionesXbeneficiario

Relación N:M entre donaciones y beneficiarios

| Campo | Descripción | Tipo | Restricciones |
|-----------------|----------------------------------|------|----------------------------|
| Id_donacion | Donación entregada | INT | PK, FK [donaciones (id)] |
| Id_beneficiario | Beneficiario que recibe la ayuda | INT | PK, FK [beneficiario (id)] |

Políticas de seguridad implementadas.

Para garantizar una administración segura de los datos, el proyecto implementa una base de datos autocontenida con usuarios internos y roles personalizados, siguiendo el principio de privilegios mínimos. Esto permite delimitar funciones entre administradores, consultores y encargados de mantenimiento, asegurando que cada usuario cuente únicamente con los permisos necesarios.

| | |
|--|---|
| Creación de usuarios: | Creación de roles: |
| <pre>CREATE USER adminUser WITH PASSWORD = 'adminUser1234'; CREATE USER readerUser WITH PASSWORD = 'readerUser1234'; CREATE USER dbaUser WITH PASSWORD = 'dbaUser1234'; GO</pre> | <pre>CREATE ROLE rol_admin CREATE ROLE rol_reader CREATE ROLE rol_mantenimiento</pre> |
| Asignación de permisos: | Asignación de roles: |
| <pre>GRANT SELECT, INSERT, UPDATE, DELETE ON SCHEMA :: dbo TO rol_admin; GRANT SELECT ON SCHEMA :: dbo TO rol_reader; GRANT ALTER, CONTROL, REFERENCES, VIEW DEFINITION ON SCHEMA :: dbo TO rol_mantenimiento;</pre> | <pre>ALTER ROLE rol_admin ADD MEMBER adminUser; ALTER ROLE rol_reader ADD MEMBER readerUser; ALTER ROLE rol_mantenimiento ADD MEMBER dbaUser;</pre> |

Esquemas:

El diseño de esquemas se realizó con el objetivo de garantizar una organización lógica, mejorar la administración del sistema y facilitar la implementación de políticas de seguridad y mantenimiento. Se definieron cinco esquemas principales, cada uno agrupando entidades que comparten una misma función dentro del modelo de negocio de la ONG:

| | |
|---|---|
| Creación de esquemas: | Inserción de tablas en esquemas: |
| <pre>CREATE SCHEMA donaciones; GO CREATE SCHEMA aporte; GO CREATE SCHEMA eventos; GO CREATE SCHEMA proyectos; GO CREATE SCHEMA personas; GO</pre> | <pre>--Donaciones ALTER SCHEMA donaciones TRANSFER dbo.donaciones ALTER SCHEMA donaciones TRANSFER dbo.donacionesxMonetario ALTER SCHEMA donaciones TRANSFER dbo.donacionesxRecursos ALTER SCHEMA donaciones TRANSFER dbo.donacionesxProyectos ALTER SCHEMA donaciones TRANSFER dbo.donacionesxBeneficiario --Aporte ALTER SCHEMA aporte TRANSFER dbo.recursos ALTER SCHEMA aporte TRANSFER dbo.monetario --Eventos ALTER SCHEMA eventos TRANSFER dbo.recaudaciones_eventos --Proyectos ALTER SCHEMA proyectos TRANSFER dbo.proyectos ALTER SCHEMA proyectos TRANSFER dbo.categoria --Personas ALTER SCHEMA personas TRANSFER dbo.donantes ALTER SCHEMA personas TRANSFER dbo.beneficiario ALTER SCHEMA personas TRANSFER dbo.tipo_beneficiario</pre> |
| Asignación de permisos: | |

```

--rol_reader
GRANT SELECT ON SCHEMA::donaciones TO rol_reader;
GRANT SELECT ON SCHEMA::aporte TO rol_reader;
GRANT SELECT ON SCHEMA::eventos TO rol_reader;
GRANT SELECT ON SCHEMA::proyectos TO rol_reader;
GRANT SELECT ON SCHEMA::personas TO rol_reader;
--rol_mantenimiento
GRANT SELECT, INSERT, UPDATE, DELETE ON SCHEMA::donaciones TO rol_mantenimiento;
GRANT SELECT, INSERT, UPDATE, DELETE ON SCHEMA::aporte TO rol_mantenimiento;
GRANT SELECT, INSERT, UPDATE, DELETE ON SCHEMA::eventos TO rol_mantenimiento;
GRANT SELECT, INSERT, UPDATE, DELETE ON SCHEMA::proyectos TO rol_mantenimiento;
GRANT SELECT, INSERT, UPDATE, DELETE ON SCHEMA::personas TO rol_mantenimiento;
--rol_admin
GRANT CONTROL ON SCHEMA::donaciones TO rol_admin;
GRANT CONTROL ON SCHEMA::aporte TO rol_admin;
GRANT CONTROL ON SCHEMA::eventos TO rol_admin;
GRANT CONTROL ON SCHEMA::proyectos TO rol_admin;
GRANT CONTROL ON SCHEMA::personas TO rol_admin;
GRANT CREATE TABLE TO rol_admin;

```

El diseño de los esquemas cumple con los principios clave de seguridad y mantenimiento:

- Gracias a los esquemas, cada rol recibe únicamente los permisos necesarios sin comprometer la seguridad del resto del sistema.
- Separar información por funciones evita cruces indebidos, facilita organización y mejora la trazabilidad.
- Cada esquema puede ser modificado, ampliado o respaldado de manera independiente.
- En caso de expansión (más tipos de aportes, nuevos proyectos, más categorías), el diseño permite crecer sin afectar el núcleo.

Auditorías:

La implementación de auditorías se diseñó para garantizar la trazabilidad, integridad y seguridad de la información sensible manejada por la organización. Para ello, se configuró una auditoría a nivel de servidor y posteriormente una Auditoría de Base de Datos que supervisa operaciones críticas sobre los esquemas que contienen datos confidenciales.

| | |
|---|---|
| Creación de auditoría a nivel servidor: | Generar una auditoria para cada apartado con información considerada sensible |
|---|---|

| | |
|--|---|
| <pre>USE master; CREATE SERVER AUDIT Audit_Donaciones TO FILE (FILEPATH = 'C:\auditoria\', MAXSIZE = 50 MB,MAX_FILES = 100) WITH (QUEUE_DELAY = 1000, ON_FAILURE = CONTINUE); GO ALTER SERVER AUDIT Audit_Donaciones WITH (STATE = ON);</pre> | <pre>USE Donaciones_ONG; CREATE DATABASE AUDIT SPECIFICATION Audit_DonacionesONG FOR SERVER AUDIT Audit_Donaciones --Para esquema de donaciones ADD (INSERT ON SCHEMA::donaciones BY PUBLIC), ADD (UPDATE ON SCHEMA::donaciones BY PUBLIC), ADD (DELETE ON SCHEMA::donaciones BY PUBLIC), -- Para el esquema de personas ADD (INSERT ON SCHEMA::personas BY PUBLIC), ADD (UPDATE ON SCHEMA::personas BY PUBLIC), ADD (DELETE ON SCHEMA::personas BY PUBLIC), --Para el esquema de Aporte ADD (INSERT ON SCHEMA::Aporte BY PUBLIC), ADD (UPDATE ON SCHEMA::Aporte BY PUBLIC), ADD (DELETE ON SCHEMA::Aporte BY PUBLIC), -- Para esquema proyecto ADD (INSERT ON SCHEMA::proyectos BY PUBLIC), ADD (UPDATE ON SCHEMA::proyectos BY PUBLIC), ADD (DELETE ON SCHEMA::proyectos BY PUBLIC), -- Para esquema eventos ADD (INSERT ON SCHEMA::eventos BY PUBLIC), ADD (UPDATE ON SCHEMA::eventos BY PUBLIC), ADD (DELETE ON SCHEMA::eventos BY PUBLIC) WITH (STATE = ON);</pre> |
| Comprobación de auditorias | |
| <pre>SELECT * FROM sys.fn_get_audit_file('C:\auditoria*', DEFAULT, DEFAULT);</pre> | |

Consultas optimizadas e índices aplicados.

Se aplicaron índices sobre claves foráneas y campos comúnmente utilizados en consultas, optimizando la velocidad de acceso y el rendimiento en operaciones de análisis.

Índices aplicados en las tablas:

| | |
|---|--|
| Donaciones: | Donantes: |
| <pre>CREATE INDEX idx_donaciones_iddonante ON donaciones(id_donante); CREATE INDEX idx_donaciones_idevento ON donaciones(id_evento);</pre> | <pre>CREATE INDEX idx_donantes_nombre ON donantes(nombre); CREATE INDEX idx_donantes_correo ON donantes(correo);</pre> |
| Recaudacions_eventos: | DonacionesXRecursos: |
| <pre>CREATE INDEX idx_eventos_nombre ON recaudacions_eventos(nombre);</pre> | <pre>CREATE INDEX idx_dxr_iddonacion ON donacionesXrecursos(id_donacion); CREATE INDEX idx_dxr_idrecurso ON donacionesXrecursos(id_recursos);</pre> |
| DonacionesXMonetario: | DonacionesXProyectos: |

| | |
|---|---|
| <pre>CREATE INDEX idx_dxm_iddonacion ON donacionesXmonetario(id_donacion); CREATE INDEX idx_dxm_idmonetario ON donacionesXmonetario(id_monetario);</pre> | <pre>CREATE INDEX idx_dxp_iddonacion ON donacionesXproyectos(id_donacion); CREATE INDEX idx_dxp_idproyecto ON donacionesXproyectos(id_proyecto);</pre> |
| DonacionesXBeneficiario: | Proyectos: |
| <pre>CREATE INDEX idx_dxb_iddonacion ON donacionesXbeneficiario(id_donacion); CREATE INDEX idx_dxb_idbeneficiario ON donacionesXbeneficiario(id_beneficiario);</pre> | <pre>CREATE INDEX idx_proyectos_idcategoria ON proyectos(id_categoria);</pre> |
| Beneficiario: | |
| <pre>CREATE INDEX idx_beneficiario_tipobene ON beneficiario(id_tipobene); CREATE INDEX idx_benef_nombre ON beneficiario(nombre);</pre> | |

El sistema incorpora consultas avanzadas mediante funciones ventana, las cuales permiten obtener totales, rankings y porcentajes de manera más eficiente.

Consultas usando funciones ventana:

| | |
|--|--|
| Total de donaciones por donante: | Porcentaje total de donaciones por cada donante, referente al total de donaciones |
| <pre>SELECT d.nombre, dn.id, SUM(m.monto) OVER (PARTITION BY d.id) AS TotalDonadoPorDonante FROM donantes AS d JOIN donaciones AS dn ON dn.id_donante = d.id JOIN donacionesXmonetario dxm ON dxm.id_donacion = dn.id JOIN monetario m ON m.id = dxm.id_monetario;</pre> | <pre>SELECT d.id, m.monto, FORMAT(m.monto * 100.0 / SUM(m.monto) OVER (), 'N2') AS PorcentajeDelTotal FROM donaciones dn JOIN donacionesXmonetario AS dxm ON dxm.id_donacion = dn.id JOIN monetario m ON m.id = dxm.id_monetario JOIN donantes AS d ON d.id = dn.id_donante;</pre> |
| Ranking de donantes por cantidad donada | |
| <pre>SELECT d.nombre, SUM(m.monto) AS TotalDonado, RANK() OVER (ORDER BY SUM(m.monto) DESC) AS RankingDonantes FROM donantes AS d JOIN donaciones AS dn ON dn.id_donante = d.id JOIN donacionesXmonetario AS dxm ON dxm.id_donacion = dn.id JOIN monetario AS m ON m.id = dxm.id_monetario GROUP BY d.nombre</pre> | |

Estrategia de dimensionamiento, respaldo y recuperación de datos.

Dimensionamiento de almacenamiento:

Permite estimar el espacio que ocupará la base de datos considerando:

- 1. Tamaño por fila
- 2. Número estimado de filas por tabla
- 3. Overhead interno de SQL Server (7 bytes adicionales por fila)
- 4. Factor de crecimiento (1.3)
- 5. Espacio aproximado usado por índices

Ejemplo de cálculo en la tabla recaudaciones_eventos:

```
CREATE TABLE recaudaciones_eventos (  
  id INT PRIMARY KEY IDENTITY(1,1) NOT NULL, --4 bytes  
  nombre VARCHAR(50), --50 + 2 = 52  
  descripcion VARCHAR(200) -- 200 + 2 = 202 bytes  
);  
GO  
-- Filas estimadas 1,000  
--Tamaño Filas: 4+52+202 : 258 bytes  
--Tamaño total = (258 + 7) * 1000 / (1024*1024) = 0.252 MB  
--Factor crecimiento 1.3  
-- 0.252 * 1.3 = 0.328 MB
```

Factor por índices: PK (10%):

$0.328 * 0.10 = 0.032 \text{ MB}$

Tamaño final = $0.328 + 0.032 = 360 \text{ MB}$

Resumen del dimensionamiento:

| Tabla | tam año | tabla | tam año | tabla | tam año | tabla | tam año |
|--|------------|----------------------|------------|-----------------------|------------|--------------------------|------------|
| donantes | 0.478 | recursos | 0.091 | categoria | 0.086 | Tipo_beneficiario | 0.086 |
| Recaudaciones eventos | 0.360 | donaciones Xrecursos | 0.034 | proyectos | 0.463 | beneficiario | 0.418 |
| donaciones | 0.032 | monetario | 0.057 | Donaciones Xproyectos | 0.034 | donacionesX beneficiario | 0.034 |
| Tamaño total estimado de la base = 2.20 MB | | | | | | | |

Estrategia de respaldo:

El sistema de gestión de donaciones de la ONG almacena información crítica como donantes, beneficiarios, proyectos, aportes y registros de donaciones. Debido a que estos datos son esenciales para auditorías, seguimiento de aportes y cumplimiento legal, se implementó el siguiente plan de respaldo y recuperación estructurado:

Objetivos de recuperación:

- **RPO (Recovery Point Objective): 6 horas**
La ONG está dispuesta a tolerar una pérdida máxima de 6 horas de información

en caso de desastre. Esto es adecuado ya que las donaciones no se procesan minuto a minuto, pero sí varias veces al día.

- **RTO (Recovery Time Objective): 1–2 horas**

El objetivo es que la base de datos pueda ser restaurada y operativa en un periodo máximo de 2 horas, suficiente para garantizar continuidad administrativa.

Para cumplir con los objetivos anteriores se estableció una estrategia combinada de **Backups Full, Diferenciales y Log Backups**, aprovechando el modelo de recuperación **FULL**.

| Tipo de Backup | Frecuencia | Motivo |
|---------------------|---|---|
| Full backup | Cada domingo a las 02:00 AM | Garantiza un punto base semanal desde el cual se construye toda la cadena de restauración. Se ejecuta fuera de horario laboral para evitar impacto en el rendimiento. |
| Differential backup | Todos los días a las 12:00 AM (medianoche) | Captura todos los cambios desde el último Full sin crear cadenas largas. Permite restaurar rápidamente sin depender de múltiples logs consecutivos. |
| Log backup | Cada 6 horas (06:00 AM, 12:00PM, 06:00 PM, 12: 00AM) | Garantiza un RPO de 6 horas, cumpliendo la tolerancia máxima de pérdida de datos requerida. |

Se crearon Jobs automatizados para garantizar la ejecución constante de los respaldos y evitar errores humanos.

Los Jobs creados fueron:



NOTA: En la captura aparecen deshabilitados, pero muestran claramente que el equipo configuró los tres Jobs necesarios para cumplir con la política de respaldo.

Migración e importación de datos.

El proceso de migración e importación de datos se desarrolló con el objetivo de garantizar una transición segura, íntegra y eficiente de la información hacia la base de datos. Este proceso se abordó en tres etapas principales: análisis de los archivos físicos, importación desde fuentes externas (CSV), migración desde una base anterior, y carga manual para completar catálogos.

| | |
|---|---|
| Verificación de almacenamiento de datos: | Importación de datos desde archivos CSV |
| <pre>SELECT name AS FileName, size/128 AS SizeMB, max_size/128 AS MaxSizeMB, physical_name FROM sys.database_files;</pre> | <pre>BULK INSERT donantes FROM 'C:\import\donantes.csv' WITH (FIRSTROW = 2, FIELDTERMINATOR = ',', ROWTERMINATOR = '\n', TABLOCK); GO</pre> |
| Migración de datos desde otra base. Ejemplo | Importación manual de datos. Ejemplo |
| <pre>INSERT INTO donaciones (fecha_donacion, id_evento, id_donante) SELECT fecha_donacion, id_evento, id_donante FROM ONG_OLD.dbo.donaciones;</pre> | <pre>INSERT INTO recursos (nombre, cantidad) VALUES ('nombre', 1111)</pre> |

Visualización en Power BI.

Se creó un dashboard conectado directamente a SQL Server, con visualizaciones útiles:

