UNIVERSIDAD CENTROAMERICANA JOSÉ SIMEÓN CAÑAS
DEPARTAMENTO DE ELECTRÓNICA E INFORMÁTICA

Administración de base de datos

Catedrático: Ing. James Edward Humberstone Morales

Informe sobre proyecto final
db_reserva_gimnasio

Estudiantes:

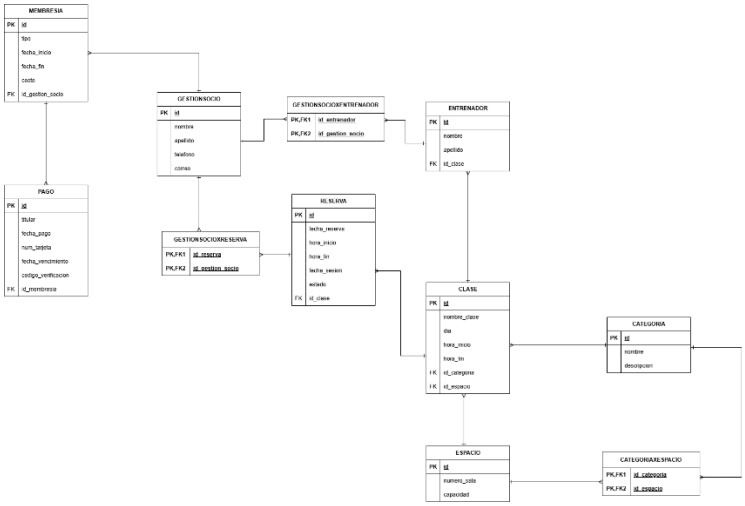
Quezada Hernandez Alejandra Michelle 00066224 sección: 01
Turcios Aparicio Andrea Yesenia 00042524 sección: 01
Villarcorta Salazar Alisson Alessandra 00062224 sección: 01
Zelada Barrientos, Keyri Margarita, 00154324 sección: 01

Fecha de entrega del manual técnico: **Jueves 27 de noviembre de 2025**

Descripción del sistema elegido y modelo de datos

El sistema elegido fue el sistema de reservas de gimnasio donde son incluidas las siguientes tablas: pagos, socios, membresías, entrenadores, clases, espacio, categoría y reserva.

El modelo que se utilizo fue un modelo relacional donde están todas las tablas respectivas y sus cardinalidades descritas con sus respectivas llaves foráneas y llaves primarias a su vez con sus campos y normalizado.



Políticas de seguridad implementadas

En db_reserva_gimnasio, su aplicación es indispensable porque se gestionan datos personales, pagos y operaciones internas; por ello, la seguridad se implementa directamente en el motor de SQL Server mediante controles de acceso por roles, restricciones de integridad y auditorías que aseguran confidencialidad, consistencia y responsabilidad sobre cada operación ejecutada en el sistema.

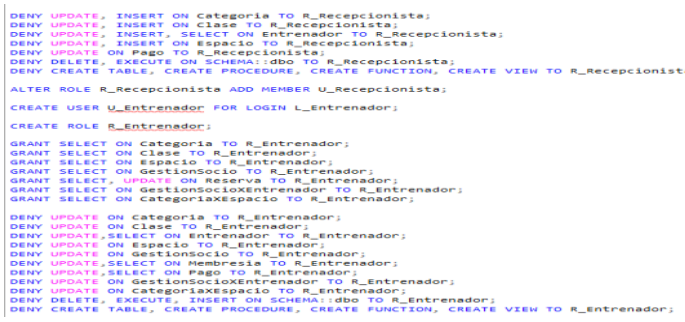
1. Gestión de identidades y control de acceso basado en roles (RBAC)

La primera capa de seguridad se implementa mediante la gestión de identidades en dos niveles: a nivel de servidor (logins) y a nivel de base de datos (usuarios). Esto permite separar la autenticación del acceso físico a la instancia, de la autorización lógica dentro de db_reserva_gimnasio.

```
USE master;
CREATE LOGIN L_Supervisor WITH PASSWORD = 'Supervisor123';
CREATE LOGIN L_Recepcionista WITH PASSWORD = 'Recepcionista123';
CREATE LOGIN L_Entrenador WITH PASSWORD = 'Entrenador123';
CREATE LOGIN L_GestorClases WITH PASSWORD = 'GestorClases123';
CREATE LOGIN L_Administrador WITH PASSWORD = 'Administrador123';
```

En este bloque se crean identidades a nivel de servidor para cada perfil funcional. El login representa la entidad que puede autenticarse contra SQL Server; responde a “quién puede conectarse” a la instancia. Ninguna operación posterior es posible si el login no existe o no es válido.

```
USE db_reserva_gimnasio;

--Crear el usuario supervisor para el usuario

CREATE USER U_Supervisor FOR LOGIN L_Supervisor;

CREATE ROLE R_Supervisor;

--Dando accesos al rol.

GRANT SELECT, INSERT, EXECUTE ON SCHEMA::dbo TO R_Supervisor;
DENY UPDATE, DELETE ON SCHEMA::dbo TO R_Supervisor;
DENY CREATE TABLE, CREATE PROCEDURE, CREATE FUNCTION, CREATE VIEW TO R_Supervisor;

ALTER ROLE R_Supervisor ADD MEMBER U_Supervisor;

CREATE USER U_Recepcionista FOR LOGIN L_Recepcionista;

CREATE ROLE R_Recepcionista;

GRANT SELECT ON Categoria TO R_Recepcionista;
GRANT SELECT ON Clase TO R_Recepcionista;
GRANT SELECT ON Espacio TO R_Recepcionista;
GRANT SELECT, UPDATE, INSERT ON GestionSocio TO R_Recepcionista;
GRANT SELECT, UPDATE, INSERT ON Membresia TO R_Recepcionista;
GRANT SELECT, INSERT ON Pago TO R_Recepcionista;
GRANT SELECT, UPDATE, INSERT ON Reserva TO R_Recepcionista;
```



```
ALTER ROLE R_Entrenador ADD MEMBER U_Entrenador;

CREATE USER U_GestorClases FOR LOGIN L_GestorClases;

CREATE ROLE R_GestorClases;

GRANT SELECT,UPDATE, DELETE, INSERT ON Categoria TO R_GestorClases;
GRANT SELECT,UPDATE, INSERT ON Clase TO R_GestorClases;
GRANT SELECT,UPDATE, DELETE, INSERT ON Espacio TO R_GestorClases;
GRANT SELECT,UPDATE, DELETE, INSERT ON CategoriaXEspacio TO R_GestorClases;
GRANT SELECT ON Reserva TO R_GestorClases;
GRANT SELECT ON GestionSocioXEntrenador TO R_GestorClases;

DENY DELETE ON Clase TO R_GestorClases;
DENY SELECT, INSERT, UPDATE, DELETE ON Entrenador TO R_GestorClases;
DENY SELECT, INSERT, UPDATE, DELETE ON GestionSocio TO R_GestorClases;
DENY SELECT, INSERT, UPDATE, DELETE ON Membresia TO R_GestorClases;
DENY SELECT, INSERT, UPDATE, DELETE ON Pago TO R_GestorClases;
DENY INSERT, UPDATE, DELETE ON Reserva TO R_GestorClases;
DENY INSERT, UPDATE, DELETE ON GestionSocioXEntrenador TO R_GestorClases;
DENY EXECUTE ON SCHEMA::dbo TO R_GestorClases;
DENY CREATE TABLE, CREATE PROCEDURE, CREATE FUNCTION, CREATE VIEW TO R_GestorClases;

ALTER ROLE R_GestorClases ADD MEMBER U_GestorClases;

CREATE USER U_Administrador FOR LOGIN L_Administrador;

CREATE ROLE R_Administrador;

GRANT SELECT,UPDATE, DELETE, INSERT,EXECUTE ON SCHEMA::dbo TO R_Administrador;
GRANT CREATE TABLE, CREATE PROCEDURE, CREATE FUNCTION, CREATE VIEW TO R_GestorClases;

ALTER ROLE R_Administrador ADD MEMBER U_Administrador;
```

En este caso, la implementación del control de acceso basado en roles (RBAC) se realizó de forma nominativa: para cada perfil de personal se creó un login, un usuario y un rol asociado, y luego se vinculó ese usuario como único miembro de ese rol. Por ejemplo, primero se define U_Supervisor para el login L_Supervisor, luego se crea R_Supervisor y, finalmente, se agregan ambos con ALTER ROLE R_Supervisor ADD MEMBER U_Supervisor. Lo mismo ocurre con U_Recepcionista y R_Recepcionista, y así sucesivamente para entrenador, gestor de clases y administrador. Teóricamente, el esquema sigue siendo RBAC porque los permisos se otorgan al rol y no directamente al usuario; la diferencia es que, en esta versión del sistema, cada rol representa a una persona específica del gimnasio. Si en el futuro se incorporan más usuarios por perfil, bastaría con crear nuevos USER y agregarlos al rol correspondiente, sin modificar la política de permisos.

Además del control de acceso, el script incorpora políticas de integridad referencial mediante claves primarias y foráneas. Aunque se consideran parte del diseño del modelo, funcionan también como barrera de seguridad al impedir que se creen registros incoherentes o que se rompan relaciones esenciales del negocio.

2. Segmentación lógicas mediante esquemas

```
--Esquemas
CREATE SCHEMA Administrador AUTHORIZATION U_Administrador;
CREATE SCHEMA GestorClase AUTHORIZATION U_GestorClases;

ALTER SCHEMA Administrador TRANSFER Pago;
ALTER SCHEMA GestorClase TRANSFER Espacio;
```

En db_reserva_gimnasio los esquemas se usan como medida de seguridad al asignar la propiedad de ciertos objetos a usuarios específicos, de modo que solo ellos puedan administrarlos; por ello se crean con autorización directa a un usuario, por ejemplo: CREATE SCHEMA Administrador AUTHORIZATION U_Administrador; y CREATE SCHEMA GestorClase AUTHORIZATION U_GestorClases;, y posteriormente se transfieren tablas a dichos espacios con ALTER SCHEMA Administrador TRANSFER Pago; y ALTER SCHEMA GestorClase TRANSFER Espacio;, lo que garantiza que información sensible como pagos quede bajo dominio exclusivo del administrador, mientras que la gestión de espacios físicos quede restringida al responsable de clases, reforzando el aislamiento y control sobre datos críticos del sistema.

3. Auditoría y trazabilidad crítica

El script incorpora auditorías a nivel de servidor y especificaciones de auditoría a nivel de base de datos. Desde la teoría de seguridad, estos mecanismos dan soporte a la responsabilidad (accountability): permiten reconstruir quién ejecutó operaciones relevantes, sobre qué objetos y en qué momento.

3.1 Auditoría sobre pagos

```
--Auditorias y especificaciones de auditorias.
USE master;
CREATE SERVER AUDIT Audit_Gimnasio
TO FILE (
FILEPATH = 'C:\SQL_Audit\audit_gimnasio',
MAXSIZE = 100 MB,
MAX_ROLLOVER_FILES = 1000)
WITH (ON_FAILURE = CONTINUE);
ALTER SERVER AUDIT Audit_Gimnasio WITH (STATE = ON);

USE db_reserva_gimnasio;
CREATE DATABASE AUDIT SPECIFICATION Audit_DB_Administrador
FOR SERVER AUDIT Audit_Gimnasio
ADD (DELETE, INSERT ON OBJECT::Administrador.Pago BY PUBLIC)
WITH (STATE = ON);
```

El objeto SERVER AUDIT define el canal físico de salida de los eventos de auditoría, en este caso archivos en disco con configuración de tamaño y rotación. La DATABASE AUDIT SPECIFICATION indica que se auditarán operaciones INSERT y DELETE sobre Administrador.Pago para cualquier usuario (BY PUBLIC).

3.2 Auditoría sobre espacios

```
USE master;
CREATE SERVER AUDIT Audit_Reserva_Gimnasio
TO FILE (
FILEPATH = 'C:\SQL_Audit\audit_gimnasio',
MAXSIZE = 100 MB,
MAX_ROLLOVER_FILES = 1000)
WITH (ON_FAILURE = CONTINUE);
ALTER SERVER AUDIT Audit_Reserva_Gimnasio WITH (STATE = ON);

USE db_reserva_gimnasio;
CREATE DATABASE AUDIT SPECIFICATION Audit_DB_GestorClase
FOR SERVER AUDIT Audit_Reserva_Gimnasio
ADD (DELETE, INSERT ON OBJECT::GestorClase.Espacio BY PUBLIC)
WITH (STATE = ON);
```

En este caso se auditan las operaciones de inserción y eliminación sobre GestorClase.Espacio. Las salas y espacios físicos determinan la capacidad del gimnasio y la programación de clases; cambios no controlados en este ámbito pueden impactar directamente en la operación y en la experiencia de los socios.

Evidencia de consultas optimizadas e índices implementados.

En la base de datos db_reserva_gimnasio, se realizaron tres consultas implementando funciones ventana.

Como primera consulta analiza la asistencia mensual a clases, agrupando por año, mes y clases específicas, calculando los totales acumulados de asistencias confirmadas.

Results Messages

	año	mes	Numero_clase	nombre_clase	asistentes_mes	total_asistentes_mes
1	2025	11	1	Spinning	20	100
2	2025	11	5	Boxeo	20	100
3	2025	11	7	Funcional	20	100
4	2025	11	8	HIIT	20	100
5	2025	11	9	TRX	20	100
6	2025	12	7	Funcional	60	260
7	2025	12	10	Yoga	40	260
8	2025	12	6	Cardio	40	260
9	2025	12	1	Spinning	40	260
10	2025	12	2	Crossfit	20	260
11	2025	12	3	Pilates	20	260
12	2025	12	4	Zumba	20	260
13	2025	12	5	Boxeo	20	260

En base a la consulta se creo un índice, llamado “IX_Reserva_Clase_Estado” referenciando a la tabla reserva, este índice organiza los datos, primero por el id_clase dentro de cada clase y luego por estado, permitiendo escanear toda la tabla con facilidad. También por esta consulta se creó el índice

“IX_Categoria_nombre” referenciando a la tabla categoría, este índice ayuda a agrupar los nombres de cada clase, permite que se organice mejor la tabla.

```
CREATE NONCLUSTERED INDEX IX_Reserva_Clase_Estado
ON Reserva(id_clase, estado);

CREATE NONCLUSTERED INDEX IX_Categoria_nombre ON Categoria(nombre);
```

Como segunda consulta, se hizo una para identificar las frecuencias de reservas por cada día de semana por su respectiva clase.

Results | Messages

	Id Clase	Dia	reservas_por_clase	Total de reservas por dia
1	2	Jueves	5	2
2	4	Jueves	5	2
3	6	Martes	5	2
4	7	Martes	5	2
5	5	Miércoles	5	2
6	9	Miércoles	5	2
7	1	Viernes	5	4
8	3	Viernes	5	4
9	8	Viernes	5	4
10	10	Viernes	5	4

En base a esta consulta se crearon los índices “IX_Clase_dia”, que permite agrupar por semana, sin este índice los datos no se agruparían bien con su respectivo día, también se creó el índice “IX_Reserva_Clase” referenciando a la tabla de Reserva, este índice lo que hace es acelerar la unión entre la tabla reserva con la tabla clase, logrando hacer un escaneo de ambas tablas, buscando en reserva el id_clase para poder relacionarlas.

```
CREATE NONCLUSTERED INDEX IX_Clase_dia ON Clase(dia);

CREATE NONCLUSTERED INDEX IX_Reserva_Clase ON Reserva(id_clase);
```

En la última consulta se utilizó una función RANK, que clasifica las membresías por popularidad, basándose en la frecuencia con la que lo piden los clientes.

Results | Messages

	membresia	total_pagos	ingresos_totales	ranking_frecuencia_pagos
1	Anual	238	47600.00	4
2	Trimestral	244	24400.00	3
3	Mensual	251	12550.00	2
4	Semestral	267	40050.00	1

Para esta consulta se creó un índice llamado “IX_Membresia_Tipo” referenciando a la tabla Membresía, acá en este índice se agrupan por el tipo de membresía, ya sea anual, trimestral, mensual o semestral, que permite que la operación de ranking y su agrupación se facilite.

```
CREATE NONCLUSTERED INDEX IX_Membresia_Tipo ON Membresia(Tipo);
```

Estrategia de dimensionamiento, respaldo y recuperación.

Dimensionamiento de la base de datos db_reserva_gimnasio

En la base de datos se encuentran 11 tablas con sus diferentes campos en cada una para el dimensionamiento fue necesario calcular individualmente el tamaño de cada tabla y por último su tamaño total de la base de datos db_reserva_gimnasio con cada uno de los cálculos de las tablas.

Se comenzó con la tabla de “Gestion Socio” donde esta posee 5 campos y uno extra que es la cabecera de fila dando un total de 326. Este total se multiplica por la cantidad de filas de la tabla y ya que esta no posee ningún índice solo se hizo la conversión a KB.

Dimensionamiento tabla Gestion Socio		
Campos	Tipo	Bytes
id	INT	4
nombre	VARCHAR	102
apellido	VARCHAR	102
telefono	CHAR	9
correo	VARCHAR	102
Cabecera de fila		7
	Total	326

Titulo	Total por numero de filas en byte	Tamaño en KB	Tamaño en KB con indices
Total	326000	318,359375	318,359375

Siguiendo con la tabla de “Pago” donde esta posee 7 campos y uno extra que es la cabecera de fila dando un total de 151. Este total se multiplica por la cantidad de filas de la tabla y ya que esta no posee ningún índice solo se hizo la conversión a KB.

Dimensionamiento tabla Pago		
Campos	Tipo	Bytes
id	INT	4
titular	VARCHAR	102
fecha_pago	DATE	3
num_tarjeta	VARCHAR	22
fecha_vencimiento	DATE	3
codigo_verificacion	CHAR	6
id_membresia	INT	4
cabecera de fila		7
	Total	151

Titulo	Total por numero de filas en byte	Tamaño en KB	Tamaño en KB con indices
Total	4000	3,90625	3,90625

Luego con la tabla de “Entrenador” donde esta posee 4 campos y uno extra que es la cabecera de fila dando un total de 219. Este total se multiplica por la cantidad de filas de la tabla y ya que esta no posee ningún índice solo se hizo la conversión a KB.

Dimensionamiento tabla Entrenador		
Campos	Tipo	Bytes
id	INT	4
nombre	VARCHAR	102
apellido	VARCHAR	102
id_clase	INT	4
cabecera de fila		7
	Total	219

Titulo	Total por numero de filas en byte	Tamaño en KB	Tamaño en KB con indices
Total	219000	213,8671875	213,8671875

Luego se encuentra la tabla de “Clase” donde esta posee 5 campos y uno extra que es la cabecera de fila dando un total de 143. Este total se multiplica por la cantidad de filas de la tabla y ya que esta posee índice que se multiplica por la cantidad de índices que tiene en este caso solo posee uno y de ultimo se hizo la conversión a KB.

Dimensionamiento tabla Clase		
Campos	Tipo	Bytes
id	INT	4
nombre	VARCHAR	102
dia	VARCHAR	22
id_categoria	INT	4
id_espacio	INT	4
cabecera de fila		7
	Total	143

Titulo	Total por numero de filas en bytes	Tamaño en KB	Tamaño en KB con indices
Total	1430	1,396484375	18,15429688

Después la tabla de “Membresia” donde esta posee 6 campos y uno extra que es la cabecera de fila dando un total de 51. Este total se multiplica por la cantidad de filas de la tabla y ya que esta posee índice que se multiplica por la cantidad de índices que tiene en este caso solo posee uno y de ultimo se hizo la conversión a KB.

Dimensionamiento tabla Membresia		
Campos	Tipo	Bytes
id	INT	4
tipo	VARCHAR	22
fecha_inicio	DATE	3
fecha_fin	DATE	3
costo	MONEY	8
id_gestion_socio	INT	4
cabecera de fila		7
	Total	51

Titulo	Total por numero de filas en bytes	Tamaño en KB	Tamaño en KB con indices
Total	51000	49,8046875	64,74609375

Siguiendo con la tabla de “Categoria” donde esta posee 3 campos y uno extra que es la cabecera de fila dando un total de 215. Este total se multiplica por la cantidad de filas de la tabla y ya que esta posee índice que se multiplica por la cantidad de índices que tiene en este caso solo posee uno y de ultimo se hizo la conversión a KB.

Dimensionamiento tabla Categoria		
Campos	Tipo	Bytes
id	INT	4
nombre	VARCHAR	102
descripcion	VARCHAR	102
cabecera de fila		7
	Total	215

Titulo	Total por numero de filas en bytes	Tamaño en KB	Tamaño en KB con indices
Total	215000	209,9609375	279500

Luego con la tabla de “Espacio” donde esta posee 3 campos y uno extra que es la cabecera de fila dando un total de 135. Este total se multiplica por la cantidad de filas de la tabla y ya que esta no posee ningún índice solo se hizo la conversión a KB.

Dimensionamiento tabla Espacio		
Campos	Tipo	Bytes
id	INT	4
num_sala	INT	102
capacidad	INT	22
cabecera de fila		7
	Total	135

Titulo	Total por numero de filas en bytes	Tamaño en KB	Tamaño en KB con indices
Total	1350	1,318359375	1,318359375

Después con la tabla “Reserva” donde esta posee 7 campos y uno extra que es la cabecera de fila dando un total de 139. Este total se multiplica por la cantidad de filas de la tabla y ya que esta posee índice que se multiplica por la cantidad de índices que tiene en este caso posee 4 índices y de ultimo se hizo la conversión a KB.

Dimensionamiento tabla Reserva		
Campos	Tipo	Bytes
id	INT	4
fecha_reserva	DATE	3
hora_inicio	DATETIME	8
hora_fin	DATETIME	8
fecha_sesion	DATE	3
estado	VARCHAR(100)	102
id_clase	INT	4
cabecera de fila		7
	Total	139

Titulo	Total por numero de filas en bytes	Tamaño en KB	Tamaño en KB con indices
Total	6950	6,787109375	19,38466309

Luego se encuentran las tablas cruzadas donde solo contienen las llaves primarias de las tablas a las que pertenecen, cada una tiene 2 campos y uno extra que es la cabecera de fila, estas tablas son “CategoriaXEspacio”, “GestionSocioXEntrenador” y “GestionSocioXReserva” además ninguna de estas posee índices solo se encuentra su conversión en KB.

Dimensionamiento tabla CategoriaXEspacio		
Campos	Tipo	Bytes
id_categoria	INT	4
id_espacio	INT	4
cabecera de fila		7
	Total	15

Titulo	Total por numero de filas en bytes	Tamaño en KB	Tamaño en KB con indices
Total	150	0,146484375	0,146484375

Dimensionamiento tabla GestionSocioXEntrenador		
Campos	Tipo	Bytes
id_gestion_socio	INT	4
id_entrenador	INT	4
cabecera de fila		7
	Total	15

Titulo	Total por numero de filas en bytes	Tamaño en KB	Tamaño en KB con indices
Total	15000	14,6484375	14,6484375

Dimensionamiento tabla GestionSocioXReserva		
Campos	Tipo	Bytes
id_gestion_socio	INT	4
id_reserva	INT	4
cabecera de fila		7
	Total	15

Titulo	Total por numero de filas en bytes	Tamaño en KB	Tamaño en KB con indices
Total	15000	14,6484375	14,6484375

Y su tamaño total de toda la base al hacer la suma de todos los tamaños en KB es de:

Tamaño de toda la base de datos
280169,1796

Respaldo y recuperación.

El modelo de recuperación que utiliza la base de datos db_reserva_gimnasio es un modelo full ya que registra todas las transacciones y no elimina los registros del log hasta que se haga un respaldo de log.

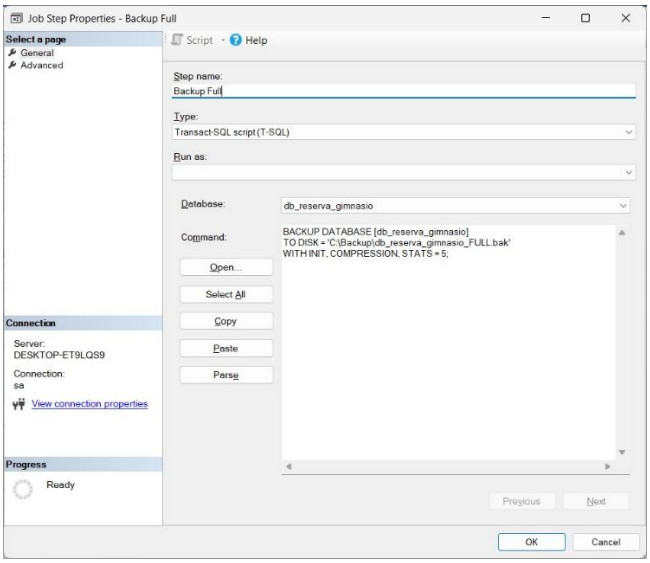
```
ALTER DATABASE db_reserva_gimnasio SET
RECOVERY FULL;
```

Como el modelo recuperación es full la base de datos permite realizar respaldos FULL, DIFFERENTIAL Y LOG.

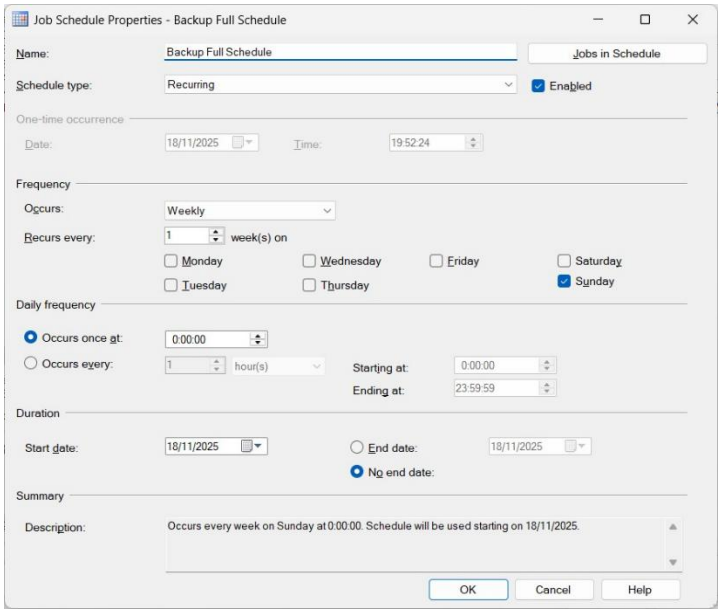
```
BACKUP DATABASE db_reserva_gimnasio
 TO DISK = 'C:\Backup\db_reserva_gimnasio_FULL.bak'
 WITH INIT, COMPRESSION, STATS = 5;

BACKUP DATABASE db_reserva_gimnasio
 TO DISK = 'C:\Backup\db_reserva_gimnasio_DIFF.bak'
 WITH DIFFERENTIAL, COMPRESSION, STATS= 5;

BACKUP LOG db_reserva_gimnasio
 TO DISK = 'C:\Backup\db_reserva_gimnasio_LOG.trn'
 WITH INIT, COMPRESSION, STATS = 5;
```

Para cada plan de respaldo se realizó un Job con su step que es la tarea que este realizara donde se le indica que cree el respaldo en este caso es el Backup FULL donde se realiza cada domingo a las 00:00 de la madrugada una vez por semana.

Step del Backup FULL

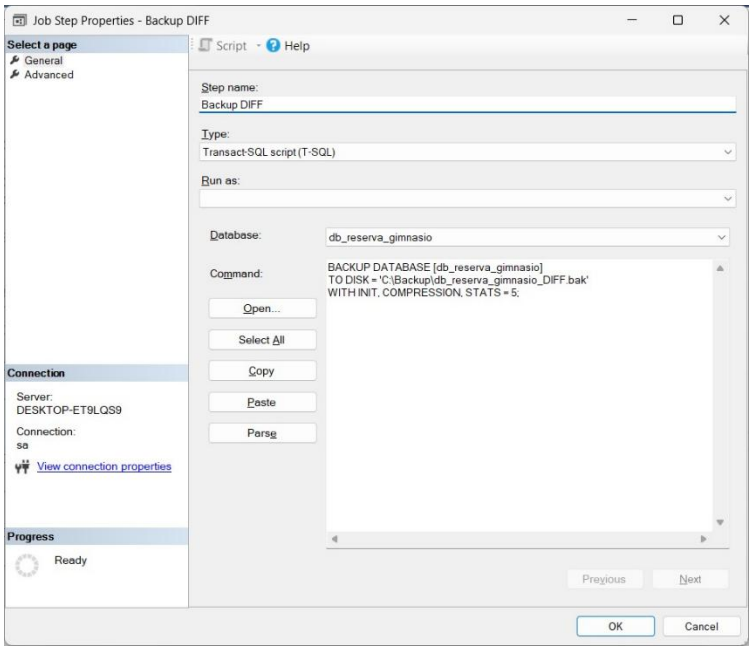


Horario del Backup FULL

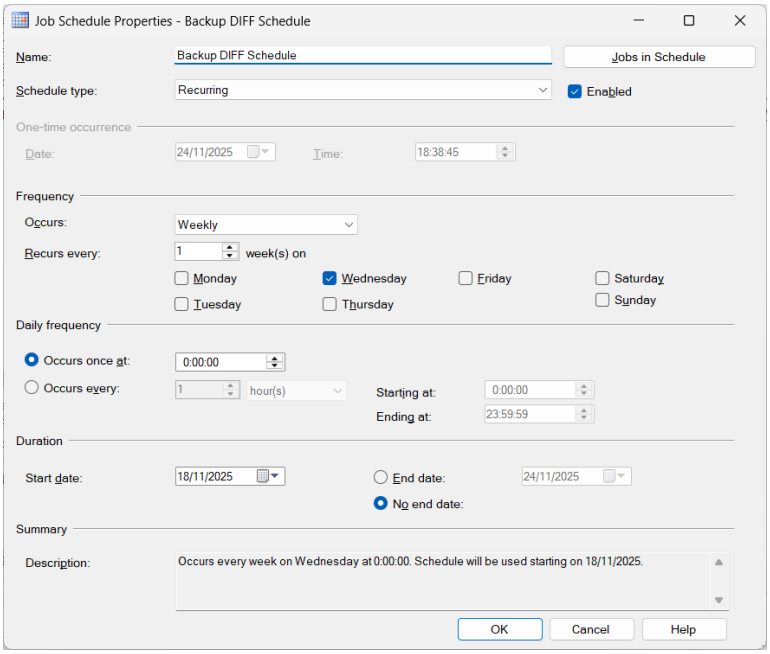


Luego está el plan de respaldo Differential donde se incluyen los cambios realizados desde el ultimo FULL y no utiliza demasiado espacio, en su step se encuentra que debe hacer la copia de seguridad y este tendrá una recurrencia de los miércoles a las 00:00 de la madrugada.

Step del Backup DIFFERENTIAL

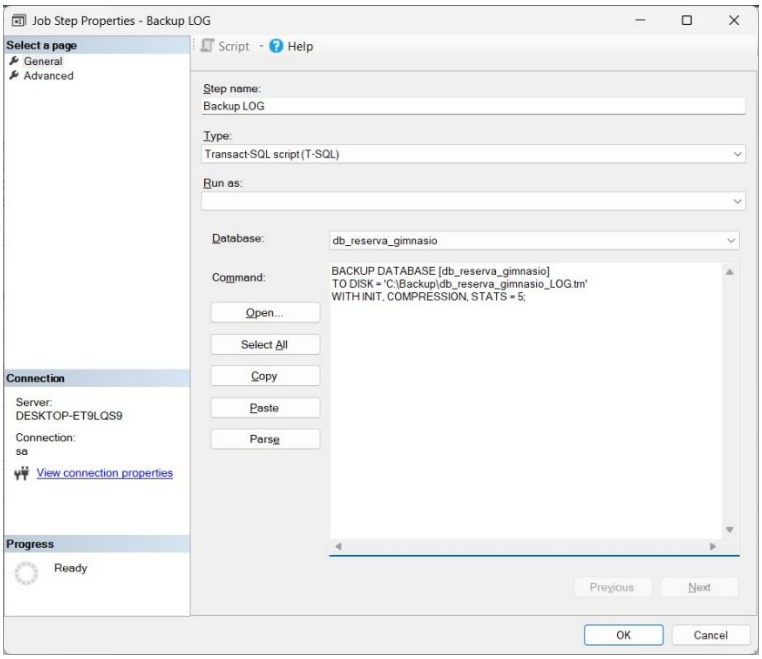


Horario de Backup DIFFERENTIAL

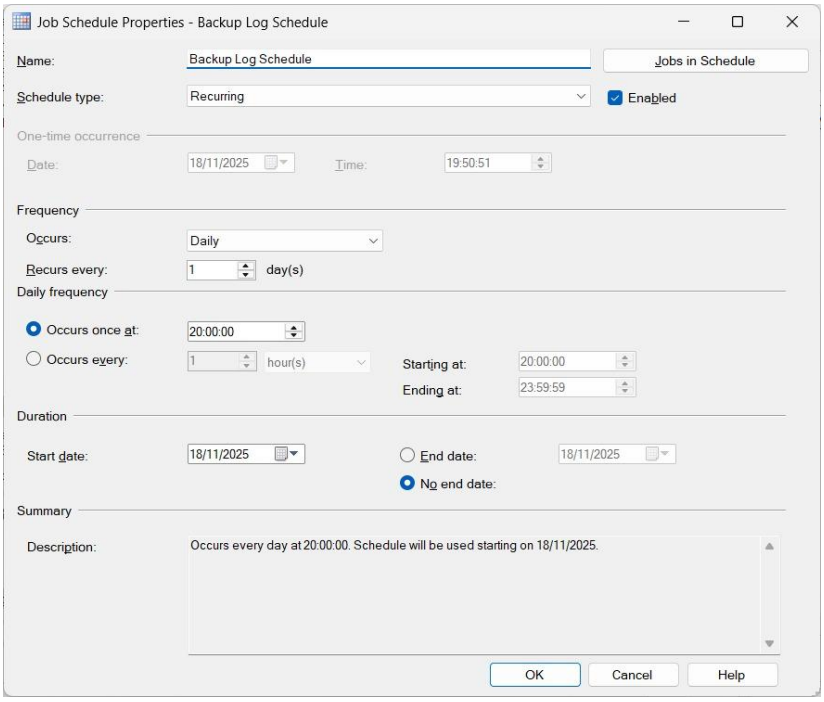


Y por último el Backup LOG que guarda los datos que han sido modificados desde la última copia de seguridad ya sea FULL o DIFFERENTIAL entonces solo procesa archivos nuevos o que han sido modificados, su step es que debe realizar el Backup LOG todos los días a las 20:00 p.m. de la noche.

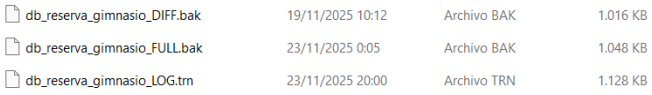
Step del Backup LOG



Horario del Backup LOG



Al momento de realizar estos Jobs para la realización de copias de seguridad se muestran de la siguiente manera.



Visualización de resultados en Power BI conectado a SQL Server.

1. CONFIGURACIÓN DE LA CONEXIÓN A BASE DE DATOS

Proceso de Conexión:

Se estableció una conexión directa entre Power BI Desktop y la base de datos SQL Server mediante los siguientes parámetros:

- **Servidor:** [Nombre del servidor que tenga la persona]
- **Base de datos:** db_reserva_gimnasio
- Tablas importadas:
    - Reserva (contiene el registro de reservas)
    - Clase (catálogo de clases y horarios)
    - Membresia (tipos y estados de membresías)
    - GestionSocio (información de socios)

Nota: Se puede importar todas las tablas y colocar sus relaciones para que de esta manera se pueda realizar más visualizaciones.

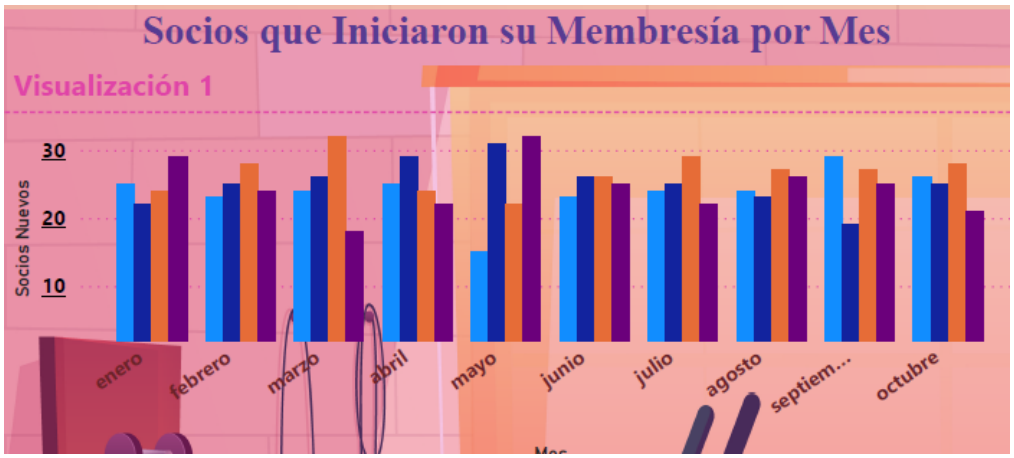
Configuración Técnica:

La conexión se realizó mediante el conector nativo de SQL Server en Power BI, permitiendo la actualización periódica de datos y asegurando que la información mostrada refleje el estado actual de la base de datos operacional.

2. VISUALIZACIONES IMPLEMENTADAS

2.1 Análisis de Crecimiento de Socios por Mes

Tipo de Visualización: Gráfico de columnas agrupadas y apiladas



Configuración Técnica:

- **Eje X:** Membresia.fecha_inicio (agrupado por mes)
- **Valores Y:** Medida = DISTINCTCOUNT(GestionSocio[id])
- **Leyenda:** Membresia.tipo (Anual, Mensual, Semestral, Trimestral)

Propósito Empresarial:

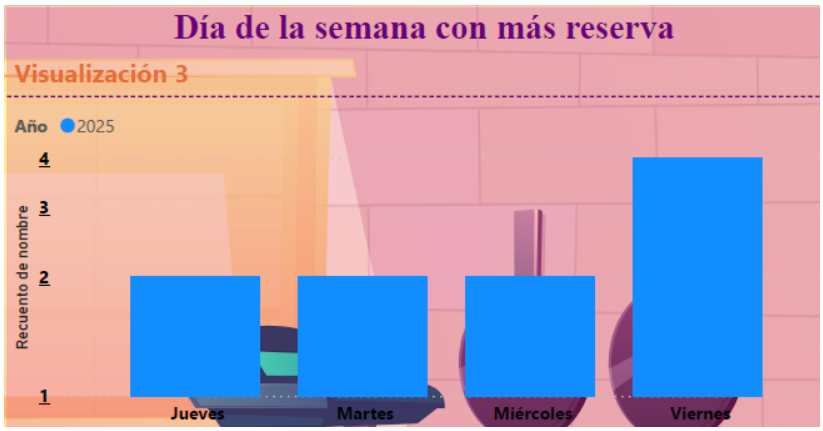
Esta visualización permite identificar tendencias mensuales en la captación de nuevos socios y analizar la preferencia por tipo de membresía. Facilita la planificación de estrategias de marketing y la proyección de ingresos.

Explicación Técnica:

Se utilizó **DISTINCTCOUNT** en lugar de **COUNT** para garantizar que cada socio se cuente una sola vez, evitando duplicados cuando un socio tiene múltiples membresías históricas.

2.2 Análisis de Días con Mayor Afluencia de Reservas

Tipo de Visualización: Gráfico de barras horizontales



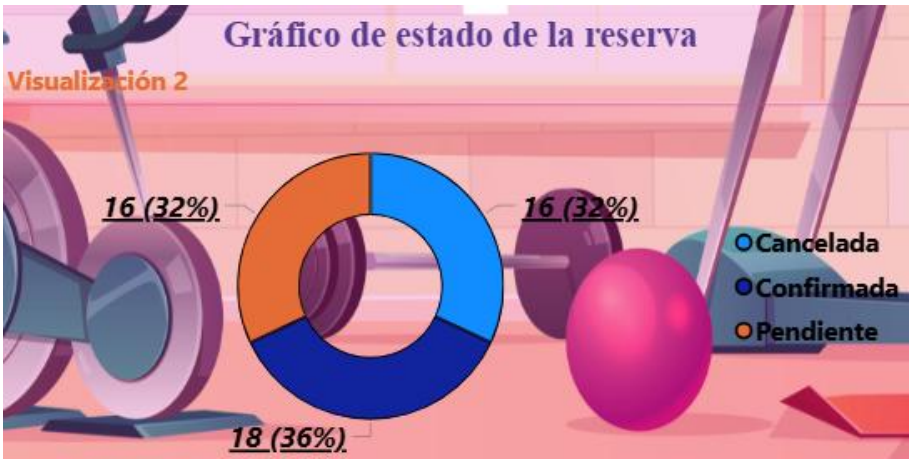
Configuración Técnica:

- **Eje Y:** Clase.dia (días de la semana)
- **Valores X:** Total Reservas por Día = COUNTROWS(Reserva)
- **Ordenamiento:** Descendente por volumen de reservas
- **Relación explotada:** Reserva.id_clase → Clase.id

Propósito Operativo:
Identifica los días de la semana con mayor demanda de clases, permitiendo optimizar la asignación de entrenadores, la gestión de espacios y la programación horaria. Los datos revelan patrones de comportamiento de los socios.

2.3 Dashboard de Estados de Reserva

Tipo de Visualización: Gráfico de anillos (Donut Chart)



Configuración Técnica:

- **Leyenda:** Reserva.estado (Confirmada, Pendiente, Cancelada)
- **Valores:** COUNT(Reserva.id)
- **Porcentajes:** Calculados automáticamente por Power BI

Propósito de Monitoreo:
Proporciona una vista instantánea del ciclo de vida de las reservas, permitiendo monitorear la efectividad del proceso de confirmación y identificar oportunidades de mejora en la experiencia del usuario.

3. KPIs IMPLEMENTADOS

3.1 Tarjeta de Total de Socios

Configuración Técnica:

- **Valor:** Medida = DISTINCTCOUNT(GestionSocio[id])
- **Formato:** Número entero

Propósito Empresarial:
Proporciona el número único de socios activos en el sistema, permitiendo monitorear la base de clientes total.

3.2 Tarjeta de Reservas Este Mes

Configuración Técnica:

- **Valor:** Medida = CALCULATE(COUNTROWS(Reserva), DATESMTD(Reserva[fecha_sesion]))
- **Formato:** Número entero

Propósito Operativo:
Muestra el volumen de reservas del mes en curso, facilitando el seguimiento de la demanda actual.

3.3 Tarjeta de Ingresos Mensuales

Configuración Técnica:

- **Valor:** Medida = CALCULATE(SUM(Membresia[costo]), DATESMTD(Membresia[fecha_inicio]))
- **Formato:** Moneda

Propósito Financiero:
Cuantifica los ingresos generados por membresías en el mes actual, apoyando el análisis de rendimiento económico.

3.4 Tarjeta de Tasa de Confirmación

Configuración Técnica:

- **Valor:** Medida = DIVIDE(CALCULATE(COUNTROWS(Reserva), Reserva[estado] = "Confirmada"), COUNTROWS(Reserva), 0)
- **Formato:** Porcentaje

Propósito de Calidad:
Evalúa la efectividad del proceso de reservas, identificando oportunidades para mejorar la experiencia del usuario.

4. BENEFICIOS OBTENIDOS

Toma de Decisiones Basada en Datos:

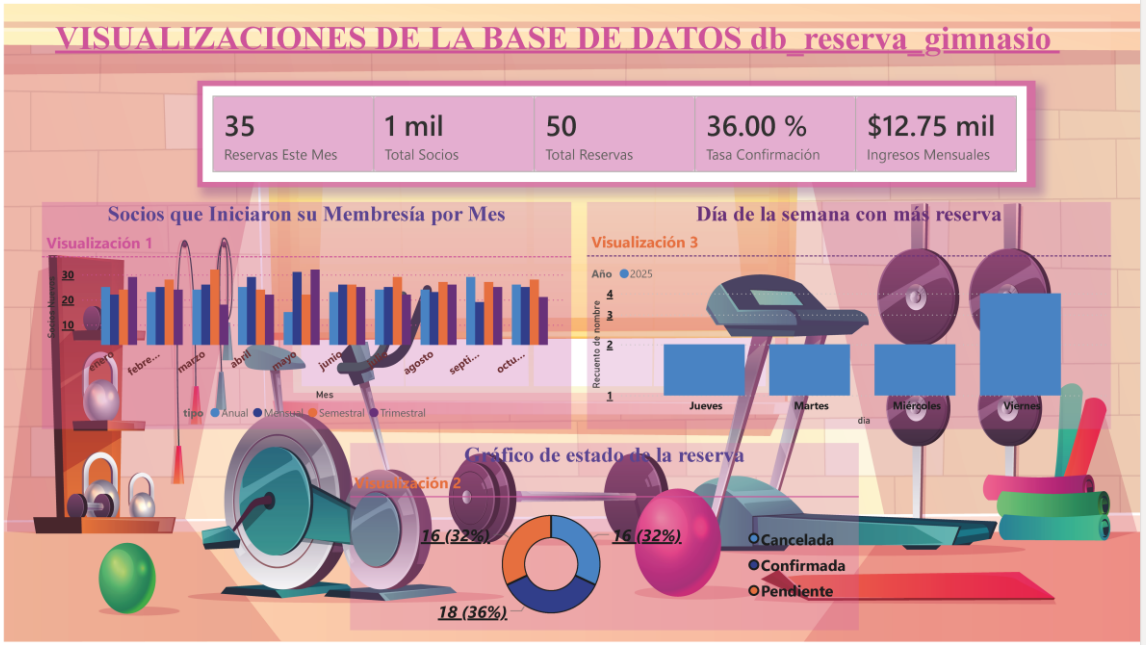
- Identificación de meses de alta y baja captación de socios
- Optimización de recursos en días de mayor demanda
- Monitoreo en tiempo real del desempeño operacional

Eficiencia Operativa:

- Reducción del tiempo de generación de reportes
- Acceso inmediato a métricas clave del negocio

Ventajas Competitivas:

- Capacidad de anticiparse a tendencias estacionales
- Mejora en la asignación de recursos humanos y físicos


VISUALIZACIONES DE LA BASE DE DATOS db_reserva_gimnasio
35
Reservas Este Mes
1 mil
Total Socios
50
Total Reservas
36.00 %
Tasa Confirmación
$12.75 mil
Ingresos Mensuales
Socios que Iniciaron su Membresía por Mes
Visualización 1
Mes
tipo Anual Mensual Semestral Trimestral
Día de la semana con más reserva
Visualización 3
Año 2025
Recuento de nombre
Jueves
Martes
Miércoles
Viernes
día
Gráfico de estado de la reserva
Visualización 2
16 (32%)
16 (32%)
18 (36%)
Cancelada
Confirmada
Pendiente