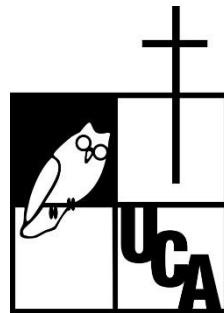


UNIVERSIDAD CENTROAMERICANA

JOSÉ SIMEÓN CAÑAS



Administración de bases de datos

Proyecto de cátedra:

sistema de reservas de gimnasio

Profesor:

James Edward Humberstone Morales

Integrantes:

Violeta Lisbeth Amaya Gonzalez

Edwin Oswaldo Guzman Melendez

Jonathan Rigoberto Martinez Menjivar

Mario Alfredo Tobar Ayala

Diana Michell Velasquez Melendez

Fecha de entrega: noviembre 26 de 2025

1. DEFINICION DEL PROYECTO.

El proyecto de cátedra se centra en el escenario de un Sistema de Reservas de Gimnasio (GIMNASIO_BD), cuyo objetivo principal es la gestión integral de socios, personal, clases, horarios, membresías y transacciones de pago. Este sistema está diseñado para optimizar los procesos operativos de un centro deportivo, asegurando un seguimiento de las reservas y la administración eficiente de los recursos.

El diseño de la base de datos se fundamenta en un modelo relacional robusto, cuya estructura lógica se detalla en el Diagrama Entidad-Relación (ER) adjunto. El modelo se compone de siete entidades principales, interconectadas para garantizar la integridad referencial y la normalización de los datos.

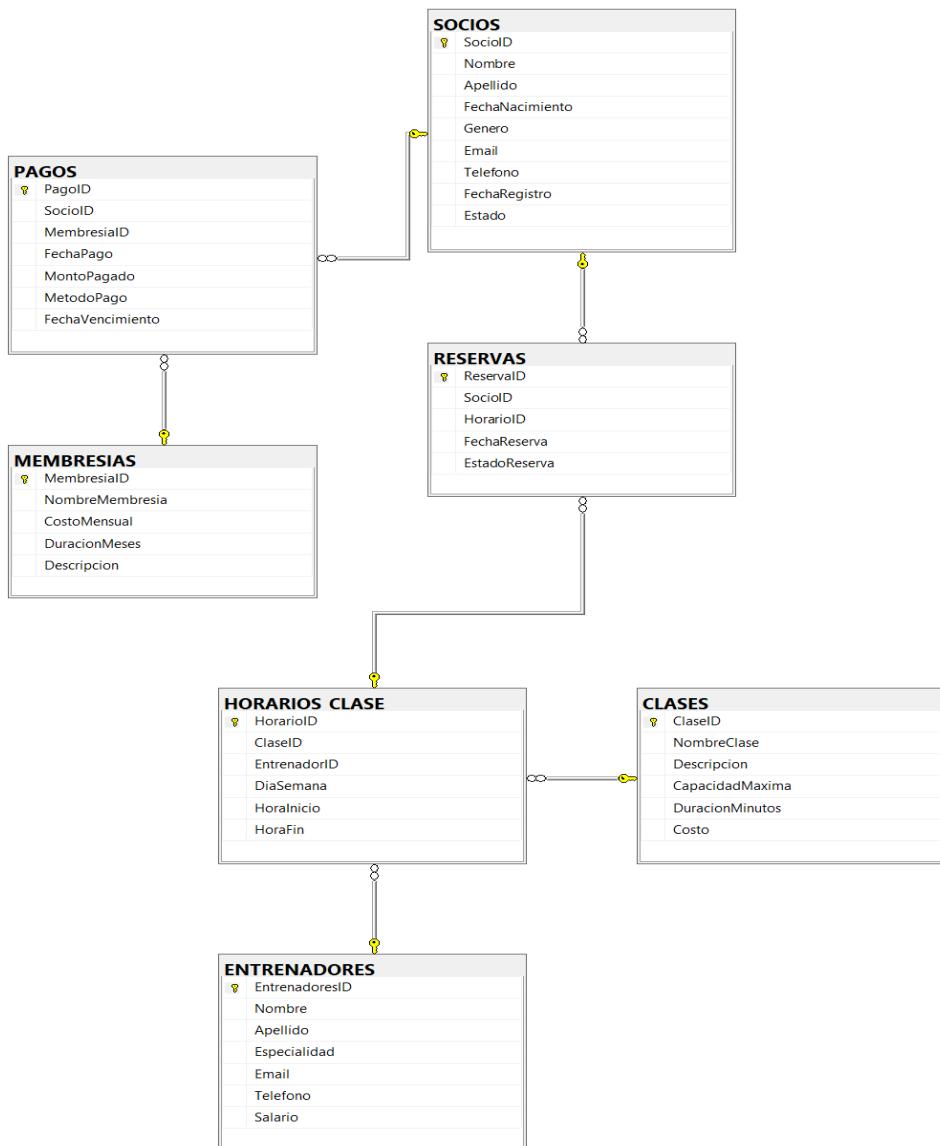


fig 1 Diagrama Entidad-Relación (ER) del Sistema GIMNASIO_BD

1.1 Diccionario de Datos

Se presenta el diccionario de datos con el objetivo de dar a conocer la estructura de la base de datos, detallando cada tabla, columna y tipo de dato

Tabla 1: Socios almacena la información personal y de registro de los socios del gimnasio.

Columna	Tipo de Dato	PK/FK	Nulo	Restricciones	Valor Default	Descripción
SocioID	INT	PK	NO	IDENTITY(1,1)	Auto	Identificador único del socio
Nombre	NVARCHAR(100)		NO			Nombre del socio
Apellido	NVARCHAR(100)		NO			Apellido del socio
FechaNacimiento	DATE		SÍ			Fecha de nacimiento del socio
Genero	CHAR(1)		SÍ	CHECK (Genero IN ('M','F','O'))		Género: M=Masculino, F=Femenino, O=Otro
Email	NVARCHAR(100)		NO	UNIQUE		Correo electrónico único del socio
Telefono	NVARCHAR(20)		SÍ			Número de teléfono de contacto
FechaRegistro	DATE		NO		GETDATE()	Fecha de registro en el gimnasio
Estado	BIT		NO		1	Estado del socio: 1=Activo, 0=Inactivo

Tabla 2: entrenadores contiene la información de los entrenadores que trabajan en el gimnasio.

Columna	Tipo de Dato	PK/FK	Nulo	Restricciones	Valor Default	Descripción
EntrenadoresID	INT	PK	NO	IDENTITY(1,1)	Auto	Identificador único del entrenador
Nombre	NVARCHAR(100)		NO			Nombre del entrenador
Apellido	NVARCHAR(100)		NO			Apellido del entrenador
Especialidad	NVARCHAR(100)		SÍ			Área de especialización (ej: Cardio, Yoga)
Email	NVARCHAR(100)		NO	UNIQUE		Correo electrónico único del entrenador
Telefono	NVARCHAR(20)		SÍ			Número de teléfono de contacto
Salario	DECIMAL(10,2)		SÍ			Salario mensual del entrenador

Tabla 3: clases describe las diferentes clases que se ofrecen en el gimnasio

Columna	Tipo de Dato	PK/FK	Nulo	Restricciones	Valor Default	Descripción
ClaseID	INT	PK	NO	IDENTITY(1,1)	Auto	Identificador único de la clase
NombreClase	NVARCHAR(100)		NO			Nombre de la clase (ej: Zumba, Spinning)
Descripcion	NVARCHAR(500)		SÍ			Descripción detallada de la clase
CapacidadMaxima	INT		NO			Número máximo de participantes
DuracionMinutos	INT		NO			Duración de la clase en minutos
Costo	DECIMAL(8,2)		NO			Costo de la clase

Tablas 4: horarios_clases define los horarios, días y entrenadores asignados para cada clase.

Columna	Tipo de Dato	PK/FK	Nulo	Restricciones	Valor Default	Descripción
HorarioID	INT	PK	NO	IDENTITY(1,1)	Auto	Identificador único del horario
ClaseID	INT	FK	NO	REFERENCES CLASES(ClaseID)		Referencia a la clase que se imparte
EntrenadorID	INT	FK	NO	REFERENCES ENTRENADORES(EntrenadorID)		Referencia al entrenador asignado
DiaSemana	NVARCHAR(15)		NO			Día de la semana (ej: Lunes, Martes)
HoraInicio	TIME		NO			Hora de inicio de la clase
HoraFin	TIME		NO			Hora de finalización de la clase

Tabla 5: reservas registra las reservas de clases hechas por los socios.

Columna	Tipo de Dato	PK/FK	Nulo	Restricciones	Valor Default	Descripción
ReservaID	INT	PK	NO	IDENTITY(1,1)	Auto	Identificador único de la reserva
SocioID	INT	FK	NO	REFERENCES SOCIOS(SocioID)		Referencia al socio que reserva
HorarioID	INT	FK	NO	REFERENCES HORARIOS_CLASE(HorarioID)		Referencia al horario de clase reservado
FechaReserva	DATE		NO	UNIQUE (SocioID, HorarioID, FechaReserva)	CONVERT(DATE, GETDATE())	Fecha de la reserva
EstadoReserva	NVARCHAR(10)		NO	CHECK (EstadoReserva IN ('Confirmada','Cancelada','Completada'))		Estado: Confirmada, Cancelada o Completada

Tabla 6: membresias define los tipos de membresías disponibles en el gimnasio.

Columna	Tipo de Dato	PK/FK	Nulo	Restricciones	Valor Default	Descripción
MembresialID	INT	PK	NO	IDENTITY(1,1)	Auto	Identificador único de la membresía
NombreMembresia	NVARCHAR(50)		NO	UNIQUE		Nombre del tipo de membresía (ej: Básica, Premium)
CostoMensual	DECIMAL(8,2)		NO			Costo mensual de la membresía
DuracionMeses	INT		NO			Duración de la membresía en meses
Descripcion	NVARCHAR(300)		SÍ			Descripción de beneficios de la membresía

Tabla 7: pagos registra los pagos de membresías realizados por los socios.

Columna	Tipo de Dato	PK/FK	Nulo	Restricciones	Valor Default	Descripción
PagoID	INT	PK	NO	IDENTITY(1,1)	Auto	Identificador único del pago
SocioID	INT	FK	NO	REFERENCES SOCIOS(SocioID)		Referencia al socio que realiza el pago
MembresialID	INT	FK	NO	REFERENCES MEMBRESIAS(MembresialID)		Referencia a la membresía pagada
FechaPago	DATE		NO		GETDATE()	Fecha en que se realizó el pago
MontoPagado	DECIMAL(8,2)		NO			Monto exacto pagado por el socio
MetodoPago	NVARCHAR(50)		SÍ			Método de pago (Tarjeta, Efectivo, Transferencia)
FechaVencimiento	DATE		NO			Fecha de vencimiento de la membresía

Relaciones entre tablas

Tabla Origen	Columna FK	Tabla Destino	Columna PK	Tipo Relación	Descripción
HORARIOS_CLASE	ClaseID	CLASES	ClaseID	N:1	Un horario pertenece a una clase
HORARIOS_CLASE	EntrenadorID	ENTRENADORES	EntrenadoresID	N:1	Un horario es impartido por un entrenador
RESERVAS	SocioID	SOCIOS	SocioID	N:1	Una reserva es hecha por un socio
RESERVAS	HorarioID	HORARIOS_CLASE	HorarioID	N:1	Una reserva corresponde a un horario de
PAGOS	SocioID	SOCIOS	SocioID	N:1	Un pago es realizado por un socio
PAGOS	MembresialID	MEMBRESIAS	MembresialID	N:1	Un pago corresponde a una membresía

Restricciones y reglas de negocio

Tabla	Tipo	Restricción	Descripción
SOCIOS	CHECK	Genero IN ('M','F','O')	El género debe ser M (Masculino), F (Femenino) u O (Otro)
SOCIOS	UNIQUE	Email	El correo electrónico debe ser único en el sistema
SOCIOS	DEFAULT	Estado = 1	Por defecto, los socios se registran como activos
SOCIOS	DEFAULT	FechaRegistro = GETDATE()	La fecha de registro se establece automáticamente
ENTRENADORES	UNIQUE	Email	El correo electrónico del entrenador debe ser único
MEMBRESIAS	UNIQUE	NombreMembresia	El nombre de la membresía debe ser único
RESERVAS	CHECK	EstadoReserva IN ('Confirmada','Cancelada','Completada')	El estado solo puede ser Confirmada, Cancelada o Completada
RESERVAS	UNIQUE	(SocioID, HorarioID, FechaReserva)	Un socio no puede reservar la misma clase dos veces en la misma fecha
RESERVAS	DEFAULT	FechaReserva = CONVERT(DATE, GETDATE())	La fecha de reserva se establece automáticamente
PAGOS	DEFAULT	FechaPago = GETDATE()	La fecha de pago se establece automáticamente

2. SEGURIDAD.

La arquitectura de seguridad se fundamenta en el principio de mínimo privilegio y segregación de funciones, garantizando que cada usuario disponga únicamente de los permisos necesarios. La implementación utiliza roles de base de datos con asignación explícita mediante GRANT y restricciones mediante DENY. Todos los logins implementan CHECK_POLICY = ON y CHECK_EXPIRATION = ON

2.1. estructura de roles y usuarios

Rol	Perfil	Login	Usuario BD	Función
rol_administrador	Admin BD	LoginAdminGym	UsuarioAdmin	Control total
rol_subadministrador	Gestión de operaciones	LoginSubAdminGym	UsuarioSubAdmin	Admin operativa
rol_recepcion	Atención al cliente	LoginRecepcion	UsuarioRecepcion	Operaciones diarias
rol_entrenador	Personal instrucción	LoginEntrenador	UsuarioEntrenador	Consulta información
rol_lector	Reportes/Auditoria	LoginLector	UsuarioLector	Ánálisis datos

2.2. Matriz de permisos consolidada

Objeto	Administrador	Subadministrador	Recepción	Entrenador	Lector
Esq.administrador	CONTROL	CRUD	S, I, U	SELECT*	DENY
Esq.contabilidad	CONTROL	CRUD	S, I	-	DENY
Esq.reportes	CONTROL	SELECT	SELECT	-	SELECT
SOCIOS	CRUD	SELECT	S, I, U	-	-
ENTRENADORES	CRUD	CRUD	-	SELECT	-
CLASES	CRUD	CRUD	SELECT	SELECT	-
HORARIOS_CLASE	CRUD	CRUD	SELECT	SELECT	-
RESERVAS	CRUD	CRUD	S, I, U	SELECT	-
MEMBRESIAS	CRUD	CRUD	SELECT	-	-
PAGOS	CRUD	CRUD	S, I	-	-
Vistas reportes	SELECT	SELECT	SELECT	-	SELECT

CRUD = Create, Read, Update, Delete | S = SELECT, I = INSERT, U = UPDATE | *Limitado a: CLASES,

HORARIOS_CLASE, RESERVAS, ENTRENADORES

2.3 Organización en esquemas

Se utiliza una arquitectura basada en esquemas para lograr una segregación lógica, superando el enfoque monolítico del esquema dbo. Los esquemas actúan como contenedores lógicos que agrupan objetos por su función, lo que permite aplicar políticas de seguridad granulares y simplificar la administración de permisos mediante el principio de mínimo privilegio.

2.4 Esquemas implementados

Esquema administración

Propósito: Encapsula las entidades operativas core del negocio.

Objetos: SOCIOS, ENTRENADORES, CLASES, HORARIOS_CLASE, RESERVAS, MEMBRESIAS

Justificación: Agrupa tablas con alta frecuencia de operaciones DML. La cohesión funcional permite aplicar políticas de acceso diferenciadas según perfil de usuario.

Esquema Contabilidad

Propósito: Aísla entidades financieras críticas con controles de auditoría estrictos.

Objetos: pagos

Justificación: Responde a requisitos de cumplimiento normativo (SOX, PCI-DSS) y segregación de funciones. Facilita auditoría selectiva, backup diferencial y trazabilidad de transacciones monetarias.

Esquema Reportes

Propósito: Capa de abstracción de solo lectura mediante vistas.

Objetos: V_SociosActivos, V_IngresosMensuales, V_OcupacionClases

Justificación: Las vistas encapsulan JOINs complejos, ocultan estructura real de tablas y proporcionan estabilidad de interfaz ante cambios estructurales.

2.5 Alineación con seguridad y mantenimiento

La arquitectura de esquemas implementada permite asignar permisos a nivel de contenedor en lugar de objeto individual, reduciendo significativamente la complejidad administrativa. Los permisos se otorgan mediante GRANT a nivel de esquema, complementados con DENY explícitos para implementar una estrategia de defensa en profundidad.

Como se evidencia en la configuración del rol_recepcion:

```
GRANT SELECT, INSERT, UPDATE ON SCHEMA::Administracion TO rol_recepcion;
GRANT SELECT, INSERT ON SCHEMA::Contabilidad TO rol_recepcion;

DENY DELETE ON SCHEMA::Administracion TO rol_recepcion;
DENY DELETE ON SCHEMA::Contabilidad TO rol_recepcion;
```

Este enfoque proporciona ventajas operativas sustanciales, ya que las nuevas tablas añadidas a un esquema heredan automáticamente los permisos configurados previamente. La denegación explícita a nivel de esquema prevalece sobre cualquier permiso individual, garantizando que las restricciones críticas se mantengan consistentemente. Adicionalmente, esta estructura simplifica los procesos de auditoría de permisos, permitiendo consultas más eficientes a sys.database_permissions filtradas por esquema en lugar de revisar permisos tabla por tabla.

2.6 Mantenibilidad

La organización por esquemas facilita significativamente la gestión y evolución de la base de datos mediante una estructura modular. Este enfoque permite una refactorización aislada, donde los cambios realizados en el esquema de Contabilidad no afectan las operaciones del esquema de Administración, manteniendo la integridad funcional de cada dominio. Adicionalmente, el diseño por esquemas posibilita deployments selectivos, ya que los scripts DDL organizados por esquema permiten despliegues parciales y controlados, optimizando los procesos de actualización. En el ámbito de la seguridad y desarrollo, esta organización favorece el testing al permitir que los ambientes de desarrollo repliquen únicamente el esquema de Administración sin incluir datos financieros sensibles, reduciendo riesgos y cumpliendo con políticas de protección de información.

Implementación: La migración de tablas desde el esquema dbo se realiza mediante la sentencia ALTER SCHEMA TRANSFER, incorporando validaciones previas que aseguran la robustez del proceso:

```
IF EXISTS (SELECT * FROM sys.tables WHERE name = 'SOCIOS' AND schema_id = SCHEMA_ID('dbo'))
ALTER SCHEMA Administracion TRANSFER dbo.SOCIOS;

IF EXISTS (SELECT * FROM sys.tables WHERE name = 'PAGOS' AND schema_id = SCHEMA_ID('dbo'))
ALTER SCHEMA Contabilidad TRANSFER dbo.PAGOS;
```

2.7 auditoría

El sistema implementa una estrategia de auditoría comprehensiva que utiliza la funcionalidad nativa de **SQL Server Audit**, estableciendo un marco de monitorización de doble nivel que abarca tanto el ámbito del servidor como el específico de la base de datos GIMNASIO_BD. Esta arquitectura dual permite capturar eventos desde la autenticación inicial hasta las operaciones transaccionales más granulares dentro de la base de datos.

Implementación del sistema de auditoría: Se ha definido un objeto de auditoría a nivel de servidor denominado Audit_GIMNASIO_BD.

```
CREATE SERVER AUDIT Audit_GIMNASIO_BD
TO FILE
(
    FILEPATH = 'C:\SQL_Audit\' ,
    MAXSIZE = 100 MB,
    MAX_ROLLOVER_FILES = 5,
    RESERVE_DISK_SPACE = OFF
)
WITH
(
    ON_FAILURE = CONTINUE,
    QUEUE_DELAY = 1000
);
```

Configuración de Almacenamiento y Retención

El objeto de auditoría se configura para almacenar los registros en archivos físicos con un esquema de rotación controlado.

Parámetros de Configuración:

- FILEPATH = 'C:\SQL_Audit': Directorio destino para archivos de auditoría
- MAXSIZE = 100 MB: Límite por archivo antes de rotación
- MAX_ROLLOVER_FILES = 5: Máximo de archivos históricos retenidos
- RESERVE_DISK_SPACE = OFF: Asignación dinámica de almacenamiento
- ON_FAILURE = CONTINUE: Operaciones continúan si auditoría falla
- QUEUE_DELAY = 1000: Balance entre performance y captura en tiempo real

2.8 Especificaciones de Auditoría

El sistema de auditoría se implementa mediante dos especificaciones complementarias que capturan eventos críticos en diferentes niveles del entorno de base de datos.

A. Especificación a Nivel de Servidor

Esta especificación monitorea la seguridad y operación del servidor SQL Server, capturando eventos fundamentales para la gestión integral del sistema:

Control de Acceso: Registra todos los intentos de conexión al servidor, incluyendo inicios de sesión exitosos (SUCCESSFUL_LOGIN_GROUP) y fallidos (FAILED_LOGIN_GROUP), así como cierres de sesión (LOGOUT_GROUP) para trazabilidad completa de sesiones.

Gestión de Seguridad: Audita modificaciones en la seguridad del servidor, incluyendo cambios en logins (SERVER_PRINCIPAL_CHANGE_GROUP) y alteraciones en membresías de roles de servidor (SERVER_ROLE_MEMBER_CHANGE_GROUP).

Operaciones Críticas: Monitoriza actividades de mantenimiento vitales como operaciones de respaldo y restauración (BACKUP_RESTORE_GROUP) para garantizar la integridad de los procedimientos de recuperación.

B. Especificación a Nivel de Base de Datos

Esta especificación se aplica específicamente a la base de datos GIMNASIO_DB, proporcionando auditoría granular sobre la estructura y los datos:

Cambios Estructurales: Rastrea modificaciones en el esquema de la base de datos, incluyendo creación, alteración o eliminación de objetos (DATABASE_OBJECT_CHANGE_GROUP), cambios en usuarios y permisos (DATABASE_PRINCIPAL_CHANGE_GROUP,

DATABASE_PERMISSION_CHANGE_GROUP), y modificaciones en roles de base de datos (DATABASE_ROLE_MEMBER_CHANGE_GROUP).

Operaciones de Datos por Esquema: Implementa auditoría específica para operaciones DML en esquemas críticos:

Esquema Administración: Audita operaciones SELECT, INSERT, UPDATE y DELETE sobre tablas de gestión operativa

Esquema Contabilidad: Monitoriza todas las operaciones DML en datos financieros sensibles

Esquema Reportes: Registra accesos de consulta (SELECT) a información analítica.

Ejemplo de implementación y resultado de auditoría

The screenshot shows a SQL Server Management Studio (SSMS) interface. On the left, there is a code editor with the following T-SQL script:

```
235
236  SELECT TOP 10 [Fecha/Hora Evento], [Usuario (Login)], [Descripcion Accion], [Objeto Afectado], [Sentencia SQL Ejecutada]
237  FROM dbo.V_Auditoria_Eventos_Criticos
238  ORDER BY [Fecha/Hora Evento] DESC;
239
```

The status bar at the bottom indicates "No se encontraron problemas." (No errors found). Below the code editor, there are two tabs: "Resultados" and "Mensajes". The "Resultados" tab is selected and displays a table with the following data:

	Fecha/Hora Evento	Usuario (Login)	Descripcion Accion	Objeto Afectado	Sentencia SQL Ejecutada
1	2025-11-25 16:25:31.4036378	sa	Login Exitooso		-- network protocol: LPC set quoted_identifier ...
2	2025-11-25 16:25:31.4026013	sa	Logout		
3	2025-11-25 16:25:31.3289897	sa	Login Exitooso		-- network protocol: LPC set quoted_identifier ...
4	2025-11-25 16:22:54.3083641	NT Service\SQLTELEMETRY	Login Exitooso		-- network protocol: LPC set quoted_identifier ...
5	2025-11-25 16:22:54.3073647	NT Service\SQLTELEMETRY	Logout		
6	2025-11-25 16:22:54.0377373	NT Service\SQLTELEMETRY	Login Exitooso		-- network protocol: LPC set quoted_identifier ...
7	2025-11-25 16:22:46.8577258	NT Service\SQLTELEMETRY	Logout		
8	2025-11-25 16:18:46.1473335	sa	Logout		
9	2025-11-25 16:17:53.9908082	NT Service\SQLTELEMETRY	Logout		
10	2025-11-25 16:17:53.9908082	NT Service\SQLTELEMETRY	Login Exitooso		-- network protocol: LPC set quoted_identifier ...

Consulta de logs implementadas

Para facilitar el análisis de los logs generados, el script implementa cinco vistas de base de datos que consultan directamente los archivos de auditoría mediante la función sys.fn_get_audit_file. Estas vistas actúan como filtros predefinidos para diferentes necesidades de reporte:

Vista de Auditoría	Propósito Principal	Eventos/Filtros Clave
<u>V Auditoria Eventos Criticos</u>	Reporte consolidado de eventos de seguridad (logins) y cambios DDL/DML en <u>GIMNASIO BD</u> .	Logins (Exitosos/Fallidos/Logout) y todas las acciones en <u>GIMNASIO BD</u> . Incluye una columna de descripción para el <u>action_id</u> .
<u>V Auditoria Logins</u>	Seguimiento detallado de la actividad de conexión y desconexión.	Logins (Exitosos/Fallidos) y Logouts.
<u>V Auditoria DML</u>	Monitoreo de operaciones de manipulación de datos.	<u>SELECT, INSERT, UPDATE, DELETE</u> dentro de <u>GIMNASIO BD</u> .
<u>V Auditoria Pagos</u>	Auditoría de transacciones de alta sensibilidad.	<u>SELECT, INSERT, UPDATE, DELETE</u> específicamente en la tabla <u>PAGOS</u> del esquema <u>Contabilidad</u> en <u>GIMNASIO BD</u> .
<u>V Auditoria Cambios Seguridad</u>	Rastreo de modificaciones en la configuración de seguridad.	Cambios en roles de BD/Servidor, permisos y principales.

2.9 Dimensionamiento de la base de datos

Tabla clases, con un número de registros de 2,151 número de columnas 6

Columna	Tipo de Dato	Tamaño (bytes)
ClaseID	INT	4
NombreClase	NVARCHAR(100)	202
Descripcion	NVARCHAR(500)	1002
CapacidadMaxima	INT	4
DuracionMinutos	INT	4
Costo	DECIMAL(8,2)	5
tamaño de fila promedio		1019

Cálculos del espacio total
(datos + overhead):

Tamaño de tabla = (tamaño fila
+ overhead) * filas estimadas

$$\text{Tamaño de tabla} = (1019 + 7) * 2151 = 2206926$$

$$\text{En MB} = 2206926 / 1024 * 1024 = 2.1046 \text{ MB}$$

Tabla entrenadores, número de registros 2,151 número de columnas 7

Columna	Tipo de Dato	Tamaño (bytes)
EntrenadoresID	INT	4
Nombre	NVARCHAR(100)	202
Apellido	NVARCHAR(100)	202
Especialidad	NVARCHAR(100)	202
Email	NVARCHAR(100)	202
Telefono	NVARCHAR(20)	42
Salario	DECIMAL(10,2)	9
tamaño total de fila		863

Cálculos del espacio total (datos + overhead):

Tamaño de tabla = (tamaño fila + overhead) * filas estimadas

Tamaño de tabla = $(863+7) * 2151 =$

$$\text{En MB} = \frac{1871370}{1048576} = 1.7846 \text{ MB}$$

Tabla horario_clases, número de registros 2,154 número de columnas 6

Columna	Tipo de Dato	Tamaño (bytes)
HorarioID	INT	4
ClaseID	INT	4
EntrenadorID	INT	4
DiaSemana	NVARCHAR(15)	32
HorarioInicio	TIME	5
HorarioFin	TIME	5
tamaño de fila		54

Cálculos del espacio total (datos + overhead):

Tamaño de tabla = (tamaño fila + overhead) * filas estimadas

Tamaño de tabla = $(54+7) * 2154 =$

$$\text{En MB} = \frac{131394}{1048576} = 0.1253 \text{ MB}$$

Tabla membresías, número de registros 2,151, número de columnas 5

Columna	Tipo de Dato	Tamaño (bytes)
MembresiaID	INT	4
NombreMembresia	NVARCHAR(50)	102
CostoMensual	DECIMAL(8)	9
DuracionMeses	INT	4
Descripcion	NVARCHAR(300)	602
Tamaño de la fila		721

Cálculos del espacio total (datos + overhead):

Tamaño de tabla = (tamaño fila + overhead) * filas estimadas

$$\text{Tamaño de tabla} = (721+7) * 2151 = \text{En MB} = \frac{910728}{1048576} = 0.8685 \text{ MB}$$

Tabla pagos, número de registros 2153, número de columnas 7

Columna	Tipo de Dato	Tamaño (bytes)
PagoID	INT	4
SocioID	INT	4
MembresiaID	INT	4
FechaPago	DATE	3
MontoPagado	DECIMAL(8,2)	5
MetodoPago	NVARCHAR(50)	102
FechaVencimiento	DATE	3
Tamaño de Fila		125

Cálculos del espacio total (datos + overhead):

Tamaño de tabla = (tamaño fila + overhead) * filas estimadas

Tamaño de tabla = (125+7) * 2153 =

$$\text{En MB} = \frac{284196}{1048576} = 0.2710 \text{ MB}$$

Tabla reservas, número de registros 2153, número de columnas 5

Columna	Tipo de Dato	Tamaño (bytes)
ReservaID	INT	4
SocioID	INT	4
HorarioID	INT	4
FechaReserva	DATE	3
EstadoReserva	NVARCHAR(10)	22
tamaño de fila		37

Cálculos del espacio total (datos + overhead):

Tamaño de tabla = (tamaño fila + overhead) * filas estimadas

Tamaño de tabla = (37+7) * 2153 =

$$\text{En MB} = \frac{94732}{1048576} = 0.0903 \text{ MB}$$

Tabla socios, número de registros 2151, número de columnas 9

Columna	Tipo de Dato	Tamaño (bytes)
SocioID	INT	4
Nombre	NVARCHAR(50)	102
Apellido	NVARCHAR(50)	102
FechaNacimiento	DATE	3
Genero	CHAR(1)	1
Email	NVARCHAR(100)	102
Telefono	NVARCHAR(20)	22
FechaRegistro	DATE	3
Estado	BIT	1
tamaño de fila		340

Cálculos del espacio total (datos + overhead):

Tamaño de tabla = (tamaño fila + overhead) * filas estimadas

Tamaño de tabla = (340+7) * 2151

$$= \text{En MB} = \frac{746397}{1048576} = 0.7118 \text{ MB}$$

Tamaño de tablas por factor de índices, paginación y crecimiento a futuro

Tablas	Índices (MB)	Paginación / fragmentación(MB)	Crecimiento a futuro (MB)	Tamaño Total Estimado (MB)
CLASES	2.7360	2.3151	2.7360	7.7870
ENTRENADORES	2.3200	1.9631	2.3200	6.6030
HORARIO CLASES	0.1629	0.1378	0.1629	0.4636
MEMEBRESIAS	1.1291	0.9554	1.1291	3.2135
PAGOS	0.3523	0.2981	0.3523	1.0027
RESERVAS	0.1174	0.9933	0.1174	1.2281
SOCIOS	0.9253	0.7830	0.9253	2.6337
Total estimado				22.9315

El dimensionamiento proyectado para la base de datos arroja un consumo total estimado de 22.93 MB, un valor considerablemente bajo para un sistema transaccional con más de 2,100 registros por entidad. Lo cual es favorable para la operación del sistema. Este reducido volumen, que incluye datos, índices y un margen de crecimiento, se inclina a garantizar un alto rendimiento al minimizar la E/S de disco y agilizar las operaciones frecuentes como consultas y reservas.

3. BACKUP

3.1 plan de backup y restauración

La estrategia de respaldo implementada está diseñada metódicamente para garantizar la continuidad operativa del sistema del gimnasio, minimizando tanto el Objetivo de Punto de Recuperación (RPO) como el Objetivo de Tiempo de Recuperación (RTO). Esta aproximación combina respaldos completos (FULL), diferenciales y de transacciones (log) en un ciclo diario estructurado, ofreciendo redundancia y agilidad en la restauración de datos críticos como pagos, reservas y registros de asistencia.

Tipo de respaldo	Frecuencia	Ventana de respaldo	RPO	RTO	Justificación
FULL diario: L–D 4:00 AM	Diario	15–25 min	Máx. 24 horas	Medio	Proporciona un punto completo del sistema todos los días, reduciendo la dependencia del diferencial.
Diferencial: L–D 4:00 PM	Diario	5–10 min	6–12 horas	Bajo	Captura todos los cambios desde el FULL de la mañana, acelerando la restauración y reduciendo pasos.
Log: 10:00 AM	Diario	1–3 min	6 horas	Muy bajo	Permite recuperar transacciones entre el FULL y el diferencial de media tarde.
Log: 10:00 PM	Diario	1–3 min	4 horas	Bajo	Garantiza un punto de recuperación nocturno previo al cierre del día.

3.2 estrategia del plan de respaldo

Plan de Respaldo

El plan de respaldo para la base de datos del gimnasio se basa en cuatro puntos diarios que garantizan un equilibrio adecuado entre seguridad, velocidad de recuperación y uso eficiente del almacenamiento. Cada día inicia con un respaldo FULL a las 4:00 AM, que sirve como base sólida para cualquier restauración. A las 10:00 AM se realiza un respaldo del log que captura los cambios transaccionales de la mañana; a las 4:00 PM se genera un respaldo diferencial que consolida todos los cambios desde el FULL del día; y finalmente, a las 10:00 PM se ejecuta un segundo respaldo del log que asegura que ninguna transacción del día quede sin protección. Esta estrategia mantiene un RPO máximo de 6 horas y un RTO muy reducido, ya que para restaurar solo se requiere combinar el FULL del día con el diferencial y los logs correspondientes según la hora del fallo.

Día	Lunes	Martes	Miércoles	Jueves	Viernes	Sábado	Domingo
04:00	FULL						
10:00	LOG						
17:00	DIFERENCIAL						
22:00	LOG						

Creacion Backup full diario

```
--BACKUP FULL - DIARIO A LAS 4:00 AM
[ ] BACKUP DATABASE GIMNASIO_DB
[ ] TO DISK = 'C:\backups\GIMNASIO_FULL.bak'
[ ] WITH INIT, STATS = 10;
% Messages
10 percent processed.
21 percent processed.
32 percent processed.
42 percent processed.
53 percent processed.
64 percent processed.
74 percent processed.
85 percent processed.
92 percent processed.
100 percent processed.
Processed 1208 pages for database 'GIMNASIO_DB', file 'GIMNASIO_DB' on file 1.
Processed 1 pages for database 'GIMNASIO_DB', file 'GIMNASIO_DB_log' on file 1.
BACKUP DATABASE successfully processed 1209 pages in 0.087 seconds (108.493 MB/sec).

Completion time: 2025-11-24T15:54:42.5010828-06:00
```

creación de Backup diferencial diario

```
--BACKUP DIFERENCIAL - DIARIO A LAS 4:00 PM
[ ] BACKUP DATABASE GIMNASIO_DB
[ ] TO DISK = 'C:\backups\GIMNASIO_DIFERENCIAL.bak'
[ ] WITH DIFFERENTIAL, STATS = 10;
% Messages
32 percent processed.
43 percent processed.
54 percent processed.
76 percent processed.
98 percent processed.
100 percent processed.
Processed 88 pages for database 'GIMNASIO_DB', file 'GIMNASIO_DB' on file 1.
Processed 1 pages for database 'GIMNASIO_DB', file 'GIMNASIO_DB_log' on file 1.
BACKUP DATABASE WITH DIFFERENTIAL successfully processed 89 pages in 0.040 seconds (17.224 MB/sec).

Completion time: 2025-11-24T15:56:12.4522509-06:00
```

Creacion de Backup log

```
--BACKUP DEL LOG - 10:00 AM
[ ] BACKUP LOG GIMNASIO_DB
[ ] TO DISK = 'C:\backups\GIMNASIO_LOG_10AM.trn'
[ ] WITH STATS = 10;
6 Messages
100 percent processed.
Processed 11 pages for database 'GIMNASIO_DB', file 'GIMNASIO_DB_log' on file 1.
BACKUP LOG successfully processed 11 pages in 0.007 seconds (11.509 MB/sec).

Completion time: 2025-11-24T15:57:10.7409996-06:00
```

```
--BACKUP DEL LOG - 10:00 PM
[ ] BACKUP LOG GIMNASIO_DB
[ ] TO DISK = 'C:\backups\GIMNASIO_LOG_10PM.trn'
[ ] WITH STATS = 10;
% Messages
100 percent processed.
Processed 1 pages for database 'GIMNASIO_DB', file 'GIMNASIO_DB_log' on file 1.
BACKUP LOG successfully processed 1 pages in 0.017 seconds (0.114 MB/sec).

Completion time: 2025-11-24T15:57:54.4779998-06:00
```

Restauración completa de los Backups

```
-- RESTAURACION

-- 1 Restaurar el FULL 4 AM
RESTORE DATABASE GIMNASIO_DB
FROM DISK = 'C:\backups\GIMNASIO_FULL.bak'
WITH NORECOVERY;

-- 2 Restaurar el DIFERENCIAL 4 PM
RESTORE DATABASE GIMNASIO_DB
FROM DISK = 'C:\backups\GIMNASIO_DIFERENCIAL.bak'
WITH NORECOVERY;

-- 3 Restaurar LOG de 10 AM
RESTORE LOG GIMNASIO_DB
FROM DISK = 'C:\backups\GIMNASIO_LOG_10AM.trn'
WITH NORECOVERY;

-- 4 Restaurar LOG de 10 PM
RESTORE LOG GIMNASIO_DB
FROM DISK = 'C:\backups\GIMNASIO_LOG_10PM.trn'
WITH RECOVERY;
```

Creación de los archivos.bk en el disco C:

The screenshot shows a Windows File Explorer window with the following details:

- Path: Este equipo > Disco local (C:) > backups >
- File List:
 - Nombre: backups clase, Fecha de modificación: 24/11/2025 12:09, Tipo: Carpeta de archivos
 - Nombre: GIMNASIO_DIFERENCIAL.bak, Fecha de modificación: 24/11/2025 15:56, Tipo: Archivo BAK, Tamaño: 2,138 KB
 - Nombre: GIMNASIO_FULL.bak, Fecha de modificación: 24/11/2025 15:54, Tipo: Archivo BAK, Tamaño: 11,354 KB
 - Nombre: GIMNASIO_LOG_10AM.trn, Fecha de modificación: 24/11/2025 15:57, Tipo: Archivo TRN, Tamaño: 150 KB
 - Nombre: GIMNASIO_LOG_10PM.trn, Fecha de modificación: 24/11/2025 15:57, Tipo: Archivo TRN, Tamaño: 86 KB

3.3 Jobs

Para garantizar la ejecución consistente y automatizada de la estrategia de respaldo definida, se implementaron los siguientes Jobs programados en el SQL Server Agent. Estos jobs aseguran que los respaldos se realicen según la frecuencia establecida, minimizando la intervención manual y reduciendo el riesgo de omisión en las copias de seguridad críticas.

A continuación, se detalla la configuración y propósito de cada Job:

Job	Frecuencia	Objetivo
JOB_FULL_DIARIO_4AM	Diario 4:00 AM	Crear el respaldo FULL del día. Base para todas las restauraciones.
JOB_LOG_10AM	Diario 10:00 AM	Capturar transacciones de la mañana y reducir ventana de pérdida.
JOB_LOG_10PM	Diario 10:00 PM	Registrar transacciones finales del día
JOB_LOG_DIFERENCIAL_4PM	Diario 4:00 PM	Capturar todos los cambios desde el último respaldo FULL (4 AM) para reducir el tiempo de restauración en caso de fallo.

El proceso de recuperación ante una contingencia sigue una secuencia estructurada y eficiente. Se inicia con la restauración del respaldo completo (FULL) más reciente, continúa con la aplicación del último respaldo diferencial disponible para reducir el número de transacciones a recuperar, y finaliza con la aplicación secuencial de los respaldos de logs transaccionales. Esta metodología permite recuperar la base de datos hasta un momento específico justo antes de la falla, garantizando la integridad de la información y cumpliendo de manera rigurosa con los objetivos de RPO y RTO establecidos.

3.4 Evidencia de ejecución

A continuación, se adjuntan capturas que evidencian la correcta configuración y ejecución de los Jobs en el SQL Server Agent, así como el monitoreo a través del Job Activity Monitor:

	name
1	JOB_DIF_4PM
2	JOB_FULL_4AM
3	JOB_LOG_10AM_10PM
4	reporte de indices fragmentados
5	syspolicy_purge_history

Ejecución de Schedules correctamente

Date	Step ID	Server	Job Name	Step Name
24/11/2025 20:55...		DESKTOP-7E02UUU	JOB_LOG_10AM_10PM	
24/11/2025 20:42...		DESKTOP-7E02UUU	JOB_DIF_4PM	
24/11/2025 20:39...		DESKTOP-7E02UUU	JOB_FULL_4AM	

4. Rendimiento

La estrategia de optimización se basó en el diseño de índices no agrupados específicamente orientados a las operaciones críticas de las consultas analíticas: filtrado, uniones, y las fases de agregación y ordenamiento inherentes a las funciones ventana. El objetivo principal fue eliminar los barridos completos de tabla y reducir los barridos de índice agrupado, permitiendo que el motor de base de datos accediera directamente a los datos mediante operaciones de búsqueda eficientes.

La implementación de índices especializados demostró una mejora importante en el rendimiento de las consultas con funciones ventana. La siguiente tabla resume las métricas clave obtenidas antes y después de la optimización:

Métrica de Rendimiento	Antes de la Indexación	Después de la Indexación	Justificación
Lecturas Lógicas	Alta. Recorrido de muchas páginas de datos	Baja. Lectura solo de páginas necesarias	El índice permite la búsqueda directa de los datos.
Tiempo de CPU	Alto. Mayor trabajo interno, Sort y Hash Aggregate	Bajo. Operaciones de agregación y ordenamiento más rápidas	El índice pre-ordena los datos, reduciendo el trabajo del motor.
Costo Estimado	Mayor. No hay ayuda para encontrar datos rápido	Menor. Ruta de ejecución más eficiente	El optimizador elige un plan de consulta más ligero.

4.1 consultas

Ranking de horarios preferidos

```
--Funciones Ventana
--1-Ranking de horarios preferidos
SELECT H.HoraInicio,
COUNT(R.ReservaID) AS TotalReservas,
RANK() OVER (ORDER BY COUNT(R.ReservaID) DESC) AS RankingHorario
FROM HORARIOS_CLASE H
JOIN RESERVAS R ON H.HorarioID = R.HorarioID
WHERE R.EstadoReserva = 'Confirmada'
GROUP BY H.HoraInicio
ORDER BY RankingHorario;
```

	HoraInicio	TotalReservas	RankingHorario
1	07:00:00.000000	23	1
2	08:45:00.000000	22	2
3	10:00:00.000000	19	3
4	13:15:00.000000	18	4
5	09:00:00.000000	18	4
6	21:00:00.000000	18	4
7	09:30:00.000000	17	7
8	12:30:00.000000	16	8
9	21:15:00.000000	16	8
10	18:15:00.000000	16	8
11	19:15:00.000000	16	8
12	19:30:00.000000	15	12
13	21:45:00.000000	15	12
14	18:45:00.000000	15	12
15	14:15:00.000000	15	12
16	09:45:00.000000	15	12
17	06:00:00.000000	14	17
18	06:30:00.000000	14	17
19	12:45:00.000000	14	17
20	13:00:00.000000	14	17
21	10:45:00.000000	14	17
22	08:30:00.000000	14	17
23	19:45:00.000000	14	17
24	18:30:00.000000	13	24
25	17:30:00.000000	13	24
26	15:45:00.000000	13	24
27	11:15:00.000000	13	24
28	11:30:00.000000	13	24
29	14:45:00.000000	13	24
30	10:15:00.000000	13	24
31	06:15:00.000000	12	31
32	13:30:00.000000	12	31
33	16:30:00.000000	12	31
34	16:45:00.000000	12	31
35	17:45:00.000000	12	31
36	20:00:00.000000	11	36

Query executed successfully.

DESKTOP-7E02UUF (16.0 RTM) | DESKTOP-7E02UUF\vic

El análisis de esta consulta identifica los horarios de clase más populares clasificándolos según el número de reservas confirmadas. Combina las tablas HORARIOS_CLASE y RESERVAS, filtra por estado 'Confirmada' y agrupa por hora de inicio. Utiliza RANK() sobre el conteo de reservas en orden descendente para asignar posiciones. El resultado prioriza horarios por demanda. La estrategia permite identificar tanto los horarios de alta demanda como las oportunidades en franjas menos concurridas.

Ranking de horario por tipo de clase

```
--2-Ranking de horarios por tipo de clase
SELECT C.NombreClase, H.DiaSemana, H.HoraInicio,
COUNT(R.ReservaID) AS TotalReservas,
RANK() OVER (PARTITION BY C.ClaseID ORDER BY COUNT(R.ReservaID) DESC) AS RankingPorClase
FROM CLASES C
JOIN HORARIOS_CLASE H ON C.ClaseID = H.ClaseID
JOIN RESERVAS R ON H.HorarioID = R.HorarioID
WHERE R.EstadoReserva = 'Confirmada'
GROUP BY C.NombreClase, C.ClaseID, H.DiaSemana, H.HoraInicio
ORDER BY C.NombreClase, RankingPorClase;
```

NombreClase	DiaSemana	HoraInicio	TotalReservas	RankingPorClase
1 Boxeo Fitness 1042	Miércoles	14:30:00.000000	1	1
2 Boxeo Fitness 1043	Lunes	10:45:00.000000	1	1
3 Boxeo Fitness 1061	Domingo	15:45:00.000000	1	1
4 Boxeo Fitness 1130	Viernes	14:00:00.000000	1	1
5 Boxeo Fitness 1143	Sábado	20:45:00.000000	2	1
6 Boxeo Fitness 1180	Lunes	11:45:00.000000	1	1
7 Boxeo Fitness 119	Jueves	11:15:00.000000	1	1
8 Boxeo Fitness 12	Sábado	21:00:00.000000	1	1
9 Boxeo Fitness 1200	Lunes	10:45:00.000000	2	1
10 Boxeo Fitness 1201	Sábado	10:45:00.000000	2	1
11 Boxeo Fitness 1227	Viernes	14:00:00.000000	1	1
12 Boxeo Fitness 1263	Sábado	07:45:00.000000	1	1
13 Boxeo Fitness 1264	Sábado	12:15:00.000000	2	1
14 Boxeo Fitness 1312	Viernes	13:00:00.000000	1	1
15 Boxeo Fitness 139	Domingo	18:15:00.000000	1	1
16 Boxeo Fitness 1405	Martes	11:30:00.000000	2	1
17 Boxeo Fitness 1405	Jueves	10:45:00.000000	1	2
18 Boxeo Fitness 1413	Viernes	08:00:00.000000	1	1
19 Boxeo Fitness 1418	Sábado	06:30:00.000000	1	1
20 Boxeo Fitness 1455	Jueves	12:30:00.000000	1	1
21 Boxeo Fitness 1479	Miércoles	16:15:00.000000	1	1
22 Boxeo Fitness 1490	Viernes	18:45:00.000000	1	1
23 Boxeo Fitness 1526	Domingo	06:30:00.000000	1	1
24 Boxeo Fitness 1535	Jueves	12:30:00.000000	2	1
25 Boxeo Fitness 1538	Martes	11:45:00.000000	1	1
26 Boxeo Fitness 1556	Viernes	08:00:00.000000	1	1
27 Boxeo Fitness 1618	Jueves	12:00:00.000000	1	1
28 Boxeo Fitness 162	Sábado	19:15:00.000000	1	1
29 Boxeo Fitness 162	Sábado	19:45:00.000000	1	1
30 Boxeo Fitness 1701	Miércoles	21:15:00.000000	1	1
31 Boxeo Fitness 1741	Lunes	07:30:00.000000	1	1
32 Boxeo Fitness 1843	Viernes	13:00:00.000000	2	1
33 Boxeo Fitness 1855	Jueves	10:45:00.000000	1	1
34 Boxeo Fitness 1893	Miércoles	08:00:00.000000	1	1
35 Boxeo Fitness 1896	Viernes	08:30:00.000000	1	1
36 Boxeo Fitness 191	Miércoles	20:00:00.000000	2	1

Query executed successfully.

Esta consulta genera un ranking de popularidad de horarios dentro de cada tipo de clase. Combina tres tablas para contar reservas confirmadas agrupadas por clase, día y hora. La función RANK() con PARTITION BY clasifica los horarios de mayor a menor demanda para cada clase específica. El resultado permite identificar los horarios preferidos por actividad, optimizando la programación semanal.

Ranking de clases preferidas

```
--3-Clases preferidas
SELECT C.NombreClase,
COUNT(R.ReservaID) AS TotalReservas,
RANK() OVER (ORDER BY COUNT(R.ReservaID) DESC) AS RankingClase
FROM CLASES C
JOIN HORARIOS_CLASE H ON C.ClaseID = H.ClaseID
JOIN RESERVAS R ON R.HorarioID = H.HorarioID
WHERE R.EstadoReserva = 'Confirmada'
GROUP BY C.NombreClase
ORDER BY RankingClase;
```

The screenshot shows a SQL query results window with the following details:

- Results**: The tab is selected.
- Messages**: No messages are present.
- Client Statistics**: Not applicable for this screenshot.
- Table Headers**: NombreClase, TotalReservas, RankingClase.
- Data Rows** (approximate values):

NombreClase	TotalReservas	RankingClase
Boxeo Fitness 1200	4	1
CrossFit 1544	4	1
Estramiento Profundo 1883	4	1
Yoga Flow 1897	4	1
Yoga Flow 2049	4	1
Yoga Flow 1965	3	6
Yoga Flow 1936	3	6
Yoga Flow 1706	3	6
Yoga Flow 675	3	6
Yoga Flow 1184	3	6
Yoga Flow 1341	3	6
Yoga Flow 1657	3	6
Spinning	3	6
Spinning 1676	3	6
Spinning 1696	3	6
Spinning 276	3	6
Zumba 1068	3	6
Zumba 451	3	6
Zumba 797	3	6
Zumba 982	3	6
CrossFit 882	3	6
Estramiento Profundo 2039	3	6
Estramiento Profundo 2074	3	6
Estramiento Profundo 566	3	6
Estramiento Profundo 819	3	6
Estramiento Profundo 99	3	6
HIIT Extremo 1176	3	6
HIIT Extremo 1797	3	6
HIIT Extremo 216	3	6
HIIT Extremo 301	3	6
HIIT Extremo 661	3	6
Pilates Mat 1703	3	6
Pilates Mat 1889	3	6
Pilates Mat 203	3	6
CrossFit 1773	3	6
CrossFit 1734	3	6
- Status Bar**: Query executed successfully.
- Icon**: DESKTO icon.

La consulta identifica las clases más populares según las reservas confirmadas. Cuenta el total de reservas por clase usando COUNT() y GROUP BY. La función RANK() ordena las clases de mayor a menor demanda. El resultado muestra un ranking claro de preferencias para orientar decisiones de programación y marketing. La información ayuda a optimizar la oferta de clases según la demanda real.

4.2 optimización e indexación de consultas ventanas

La optimización del rendimiento se basó en el diseño estratégico de índices compuestos orientados a cubrir las columnas utilizadas en operaciones críticas de filtrado, uniones, agrupación y ordenamiento. Esta aproximación permitió transformar operaciones costosas de escaneo completo en búsquedas eficientes mediante índices.

Para la consulta de Ranking por Total Pagado, se implementó un índice no agrupado en la tabla PAGOS sobre las columnas SocioID y MontoPagado. Este índice acelera significativamente la agregación mediante SUM y GROUP BY, además de proporcionar un cubrimiento total que evita accesos adicionales a la tabla base.

En el caso del Ranking por Tipo de Clase, que involucra las tablas RESERVAS y HORARIOS_CLASE, se crearon índices en RESERVAS(HorarioID) y en HORARIOS_CLASE(HorarioID, ClaseID). Estos índices optimizan las uniones entre tablas y la fase de agrupamiento previa a la aplicación de la función ventana, reduciendo sustancialmente el costo de las operaciones JOIN.

Para la consulta de Variación Mensual con LAG(), se implementó un índice en RESERVAS(FechaReserva) que mejora la eficiencia de la agregación por intervalos de fecha y facilita el ordenamiento temporal requerido por la función LAG(). Este índice permite un acceso rápido a los datos ordenados cronológicamente, esencial para el cálculo de comparaciones entre períodos.

La implementación de estos índices permitió que el optimizador de SQL Server generara planes de ejecución más eficientes, donde las funciones ventana operan sobre conjuntos de datos previamente filtrados, ordenados y agrupados, traduciéndose en una reducción significativa del consumo de recursos y una mejora notable en los tiempos de respuesta.

5. EXPLOTACIÓN

Para optimizar el rendimiento y simplificar el modelo de datos en Power BI, se implementó una vista consolidada directamente en SQL Server. Esta vista, denominada v_AnalisisRendimiento, integra en una única estructura todas las tablas relevantes, como Pagos, Socios, Clases y Reservas.

¿Por qué se utilizó una vista?

Rendimiento: Se delega el trabajo pesado de procesamiento y unión de datos a SQL Server, que está optimizado para estas tareas. Esto hace que el dashboard en Power BI sea más rápido y responsive.

Simplicidad: En lugar de cargar múltiples tablas en Power BI y crear complejas relaciones, solo se carga una única tabla (la vista). Esto simplifica el modelo de datos y reduce la posibilidad de errores.

Mantenimiento: Cualquier cambio en la lógica de negocio se puede realizar directamente en la vista SQL, sin necesidad de modificar el reporte de Power BI.

Script de la Vista

A continuación, se muestra el script SQL utilizado para crear la vista v_AnalisisRendimiento:

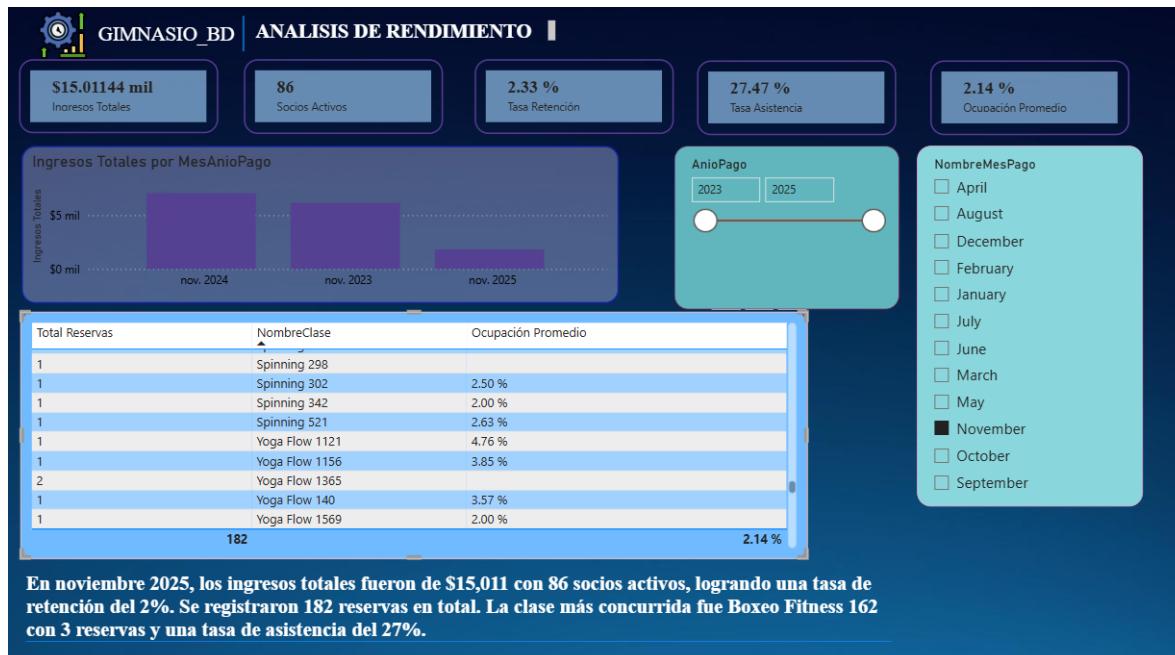
```

77
78  SELECT TOP 10 * FROM dbo.v_AnalisisRendimiento ORDER BY FechaPago DESC;

```

PagoID	FechaPago	Ingresos	MetodoPago	FechaVencimiento	AnioPago	MesPago	NombreMesPago	MesAnioPago	SocioID	NombreSocio	FechaNacimiento	EdadSocio	RangoEdad	Genero	Email	FechaRegis	
1	2154	2025-11-21	500.00	Tarjeta	2025-12-21	2025	11	November	nov. 2025	1	Ana Lopez	1995-03-15	30	26-35	F	ana.lopez@mail.com	2025-11-04
2	2154	2025-11-21	500.00	Tarjeta	2025-12-21	2025	11	November	nov. 2025	1	Ana Lopez	1995-03-15	30	26-35	F	ana.lopez@mail.com	2025-11-04
3	2154	2025-11-21	500.00	Tarjeta	2025-12-21	2025	11	November	nov. 2025	1	Ana Lopez	1995-03-15	30	26-35	F	ana.lopez@mail.com	2025-11-04
4	3	2025-11-04	80.00	Transferencia	2026-05-04	2025	11	November	nov. 2025	4	Pedro Sanchez	1975-01-25	50	46-55	M	pedro.sanchez@mail.com	2025-11-04
5	3	2025-11-04	80.00	Transferencia	2026-05-04	2025	11	November	nov. 2025	4	Pedro Sanchez	1975-01-25	50	46-55	M	pedro.sanchez@mail.com	2025-11-04
6	1	2025-11-04	50.00	Tarjeta de Credito	2026-02-04	2025	11	November	nov. 2025	1	Ana Lopez	1995-03-15	30	26-35	F	ana.lopez@mail.com	2025-11-04
7	1	2025-11-04	50.00	Tarjeta de Credito	2026-02-04	2025	11	November	nov. 2025	1	Ana Lopez	1995-03-15	30	26-35	F	ana.lopez@mail.com	2025-11-04
8	1	2025-11-04	50.00	Tarjeta de Credito	2026-02-04	2025	11	November	nov. 2025	1	Ana Lopez	1995-03-15	30	26-35	F	ana.lopez@mail.com	2025-11-04
9	1488	2025-10-28	54.71	Efectivo	2026-10-23	2025	10	October	oct. 2025	56	Candelario Daza	1988-12-07	37	36-45	O	merinocinco@example.net	2023-12-06
10	222	2025-10-28	37.48	PayPal	2025-11-27	2025	10	October	oct. 2025	343	Tito Bustos	2004-01-09	21	18-25	O	mnoquera@example.net	2025-10-27

Análisis del Dashboard



Componentes y Utilidad

1.KPIs Principales:

Utilidad: Ofrecen una lectura inmediata de la salud del negocio. Métricas como Ingresos Totales, Socios Activos y Tasa de Retención son útiles para la gestión diaria y estratégica.

2.Filtros Interactivos:

Permiten al usuario segmentar la información por Año y Mes. Esta interactividad es útil para descubrir tendencias, comparar períodos y realizar análisis más profundos sin necesidad de múltiples reportes.

3.Ingresos Totales por Mes:

Muestra la evolución y estacionalidad de los ingresos, permitiendo identificar los meses de mayor y menor rendimiento para planificar campañas de marketing o promociones.

4.Ranking de Clases:

Identifica qué clases son las más populares y cuáles tienen mejor rendimiento en términos de ocupación. Es una herramienta útil para optimizar la oferta de clases y la asignación de recursos.

5.Narrativa Ejecutiva:

Nos brinda un resumen en lenguaje natural de los hallazgos más importantes del período seleccionado. Esto es útil para directivos o usuarios que necesitan una conclusión rápida sin tener que interpretar cada gráfico individualmente.