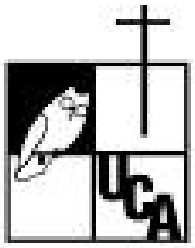


Universidad Centroamericana
José Simeón Cañas

Proyecto de catedra – Administración de base de datos

Ing. James Edward Humberstone Morales

André Alberto Calidonio Campos – 00363918

Kalet Adonay Chavez Alas - 00369924

Jose Ernesto Gonzalez Castillo – 00061524

Eyler Alejandro Guido Urbina – 00139823

Fecha: jueves 27 de noviembre de 2025

1. DESCRIPCIÓN DEL SISTEMA Y MODELO DE DATOS

1.1 Alcance del Sistema

El sistema **Gym_DB** ha sido diseñado para operar como el núcleo digital del gimnasio. Su alcance abarca desde el registro inicial del socio en la recepción hasta la facturación mensual y la reserva de clases grupales. El sistema permite la interacción concurrente de múltiples perfiles de usuario (recepcionistas, gerentes, entrenadores) sin comprometer la integridad de la información.

1.2 Diccionario de Datos y Entidades

A continuación, se describen las tablas principales que conforman el esquema dbo, detallando su propósito y estructura.

1.2.1 Módulo de Gestión de Socios

- **Tabla member:** Representa la entidad central del negocio. Almacena los datos demográficos de los clientes. Se ha diseñado utilizando tipos de datos eficientes (VARCHAR para textos variables y DATE para fechas) para optimizar el almacenamiento.
    - *Columna Clave:* member_id (Identity).
    - *Datos Críticos:* identity_document (Documento único) y email (Medio de contacto).
- **Tabla member_membership:** Gestiona la relación muchos a muchos entre socios y tipos de membresía, pero con un enfoque histórico. Permite saber no solo qué plan tiene el socio actualmente, sino cuáles ha tenido en el pasado mediante las fechas de vigencia (start_date, end_date).

1.2.2 Módulo de Operaciones y Reservas

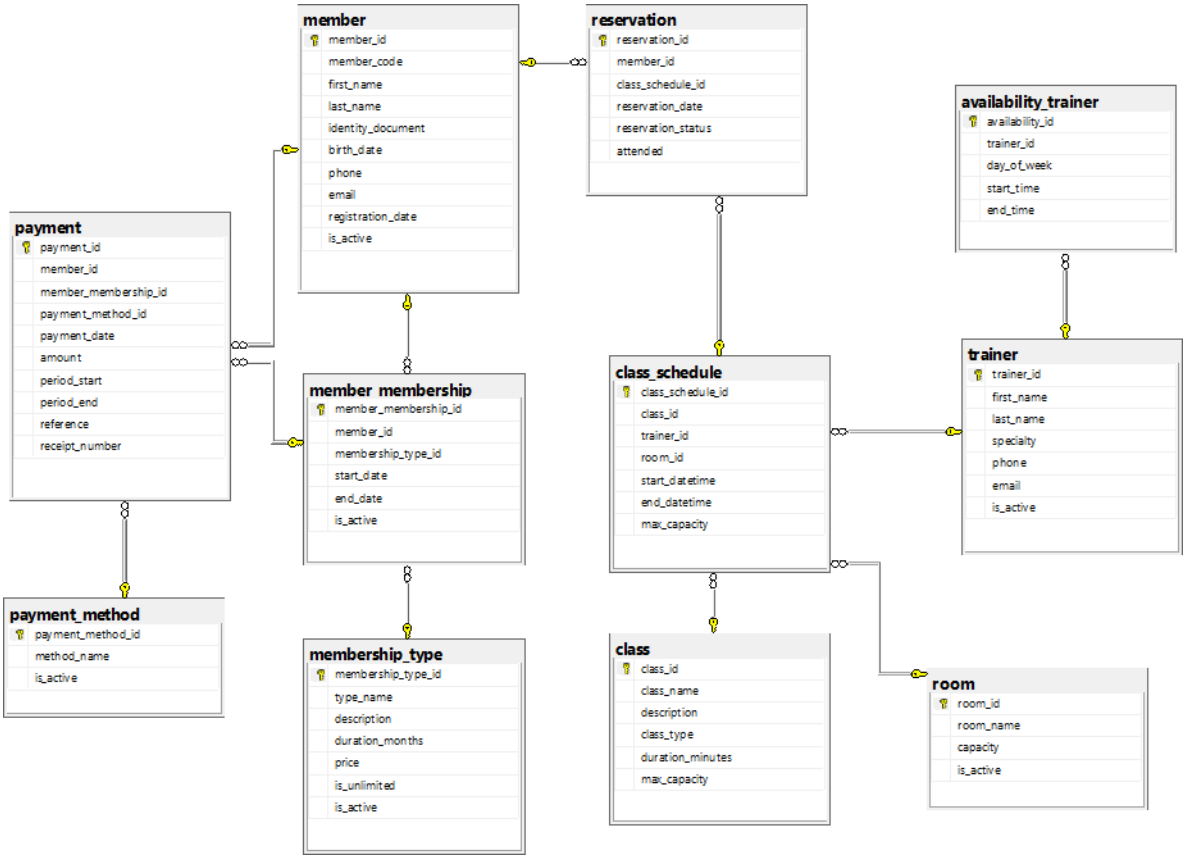
- **Tabla class_schedule:** Es la tabla maestra de la agenda. Relaciona tres recursos finitos: el Entrenador (trainer_id), la Sala (room_id) y la Clase (class_id) en un momento específico del tiempo.
- **Tabla reservation:** Controla la ocupación. Esta tabla es transaccional y de alto crecimiento. Incluye restricciones de unicidad (UNIQUE CONSTRAINT) para evitar que un mismo socio reserve dos veces la misma clase, previniendo errores lógicos en la aplicación.

1.2.3 Módulo Financiero

- **Tabla payment:** Registra el flujo de caja. Cada fila representa un ingreso monetario validado. Se vincula estrictamente al socio para permitir auditorías financieras.

1.3 Modelo Entidad-Relación (Diagrama ER)

El diseño sigue las reglas de normalización hasta la **Tercera Forma Normal (3NF)** para eliminar redundancias. Las relaciones se mantienen mediante llaves foráneas (FOREIGN KEY) que aseguran, por ejemplo, que no se pueda registrar un pago para un socio que no existe.



2. POLÍTICAS DE SEGURIDAD IMPLEMENTADAS

La seguridad de la información no es una característica opcional, sino un requisito funcional. En el desarrollo de **Gym_DB**, se ha descartado el uso de un único usuario administrador para todas las conexiones (mala práctica común), optando por un modelo de seguridad granular.

2.1 Principio de Menor Privilegio

Se ha aplicado estrictamente el principio de **Menor Privilegio**, el cual dicta que cada usuario debe tener únicamente los permisos necesarios para desempeñar su función laboral, y ni uno más. Esto minimiza la superficie de ataque ante accesos no autorizados o errores humanos internos.

2.2 Implementación de RBAC (Role-Based Access Control)

En lugar de asignar permisos a individuos, se crearon **Roles de Base de Datos**. Esto facilita la administración: si un empleado cambia de puesto, solo se cambia su rol, sin necesidad de reconfigurar permisos objeto por objeto.

Detalle de Roles Configurados:

A. Rol: general_manager_rol(Administrador general)

Este rol es el dueño de la base de datos como tal.

B. Rol: membership_manager_rol (Área de Ventas)

Este rol está diseñado para los asesores comerciales.

- **Permisos Concedidos:** SELECT, INSERT, UPDATE en las tablas member y member_membership.
- **Justificación:** Los vendedores necesitan registrar nuevos clientes y actualizar sus datos.
- **Restricciones:** Se ha revocado explícitamente el permiso DELETE. Un vendedor no debe poder eliminar un cliente de la base de datos; si un cliente se da de baja, simplemente se cambia su estado a inactivo (is_active = 0), preservando la integridad histórica.

B. Rol: payment_manager_rol (Área de Caja)

Este rol maneja la información más sensible: el dinero.

- **Permisos Concedidos:** Control total sobre la tabla payment.
- **Aislamiento:** Este es el único rol operativo con capacidad de escritura en la tabla financiera. Esto previene fraudes, ya que ni los recepcionistas ni los entrenadores pueden "inventar" pagos en el sistema.

C. Rol: trainer_rol(Manejo de entrenadores)

Este rol nos da acceso a la lectura de informacion de una clase.

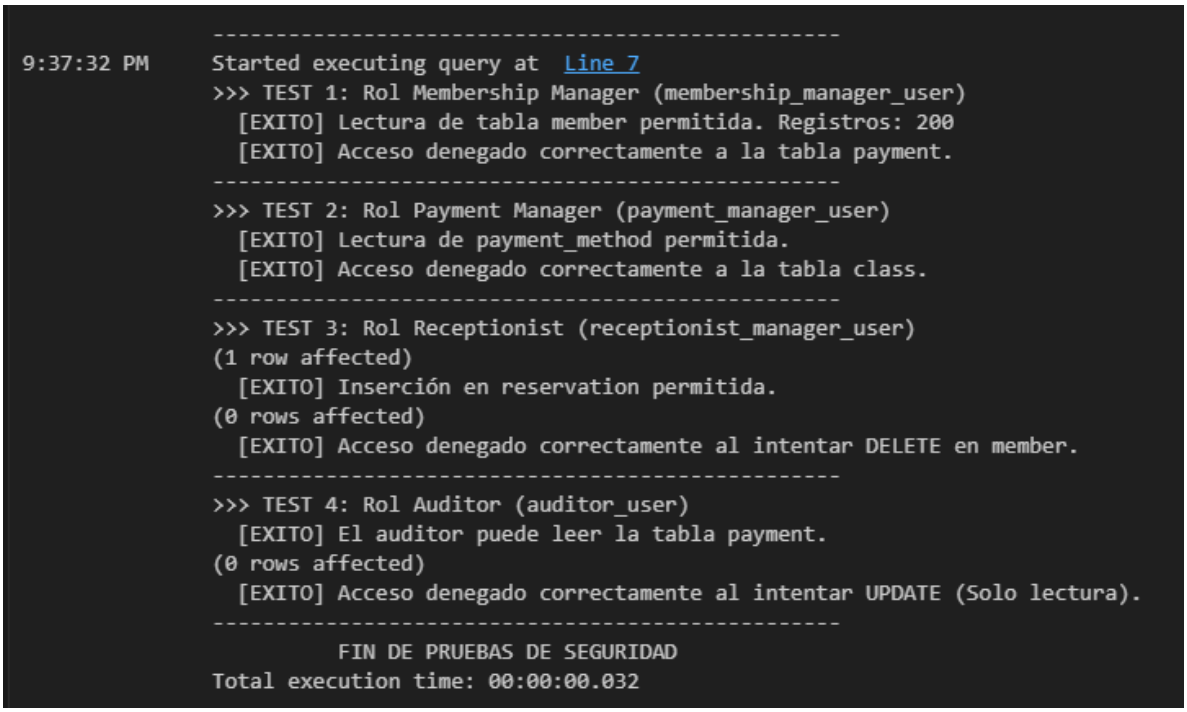
- **Permisos Concedidos:** SELECT en las tablas class, class_schedule, member y SELECT , UPDATE en reservation dado que puede actualizar información limitada de la información de reservación ya sea si el socio ya no esta en la clase se cambia la fecha de entrenamiento u hora.
- **Justificación:** Los entrenadores solo pueden tener acceso a ver la información de las clases y sus miembros dado que necesitan estar al tanto de ello, pero se le puede dar
- **Restricciones:** Se ha revocado explícitamente el permiso DELETE. Un vendedor no debe poder eliminar un cliente de la base de datos; si un cliente se da de baja, simplemente se cambia su estado a inactivo (is_active = 0), preservando la integridad histórica.

D. Rol: receptionist_rol (Atención al Cliente)

- **Permisos Concedidos:** Acceso de lectura a horarios y modificación de la tabla reservation.
- **Privacidad:** El recepcionista necesita ver si un socio está activo para dejarlo entrar, pero **no tiene permisos** para ver cuánto pagó el socio ni su historial financiero detallado, protegiendo la privacidad del cliente.

E. Rol: auditor_rol (Control Interno)

- **Permisos Concedidos:** SELECT sobre todo el esquema dbo.
- **Seguridad:** Este usuario es de "Solo Lectura". Garantiza que durante una auditoría no se alteren los datos accidentalmente.



3. EVIDENCIA DE OPTIMIZACIÓN Y RENDIMIENTO

A medida que la base de datos escala de cientos a miles de registros, el rendimiento de las consultas puede degradarse. Para mitigar esto, se han implementado estructuras de optimización físicas (Índices) y lógicas (Funciones de Ventana).

3.1 Estrategia de Indexación

Se han analizado las cargas de trabajo (workloads) esperadas y se han creado los siguientes índices No Agrupados (*Non-Clustered Indexes*):

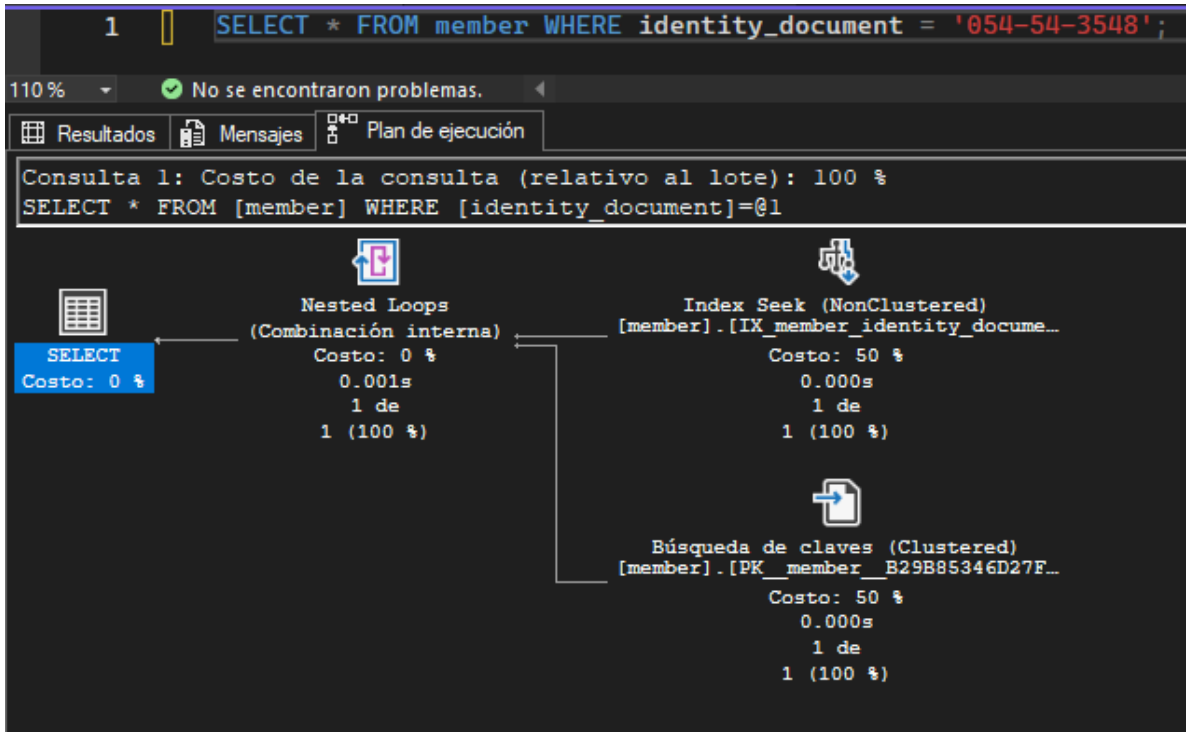
3.1.1 Índice Filtrado para Reservas (IX_reservation_active)

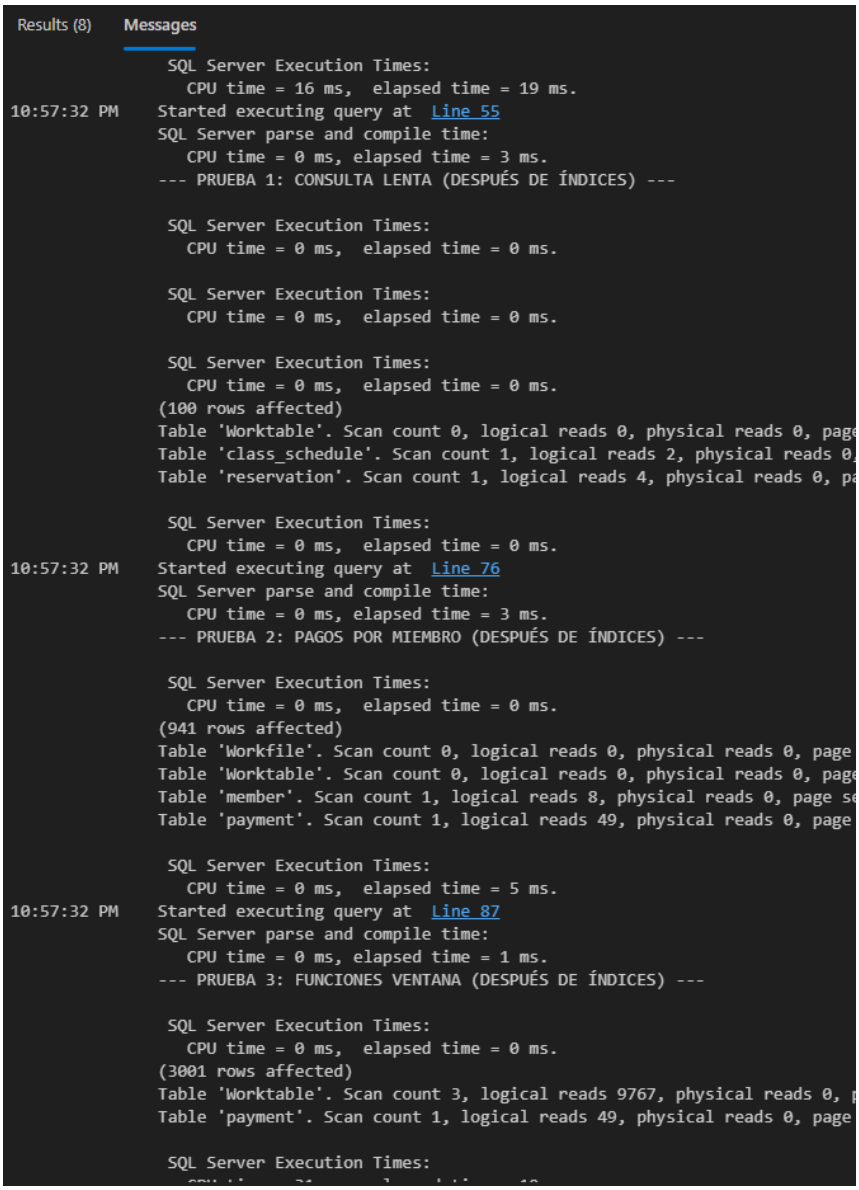
- **El Problema:** Con el tiempo, la tabla reservation tendrá millones de registros históricos. Sin embargo, el sistema consulta el 90% del tiempo solo las reservas "Futuras" o "Activas".

- **La Solución:** CREATE INDEX ... WHERE reservation_status = 'Reserved'.
- **Impacto:** Este índice es mucho más pequeño y rápido que un índice normal, ya que solo incluye las filas vivas. Reduce drásticamente las operaciones de lectura en disco (I/O) para el dashboard de recepción.

3.1.2 Índice de Búsqueda por Nombre (IX_member_name)

- **El Problema:** SQL Server, por defecto, ordena los socios por ID. Buscar a "Juan Pérez" obligaría al motor a escanear toda la tabla (Table Scan).
- **La Solución:** Un índice compuesto por first_name y last_name.
- **Impacto:** Convierte la búsqueda secuencial en una búsqueda directa (Index Seek), reduciendo el tiempo de respuesta de segundos a milisegundos.





3.2 Consultas Avanzadas y Funciones de Ventana

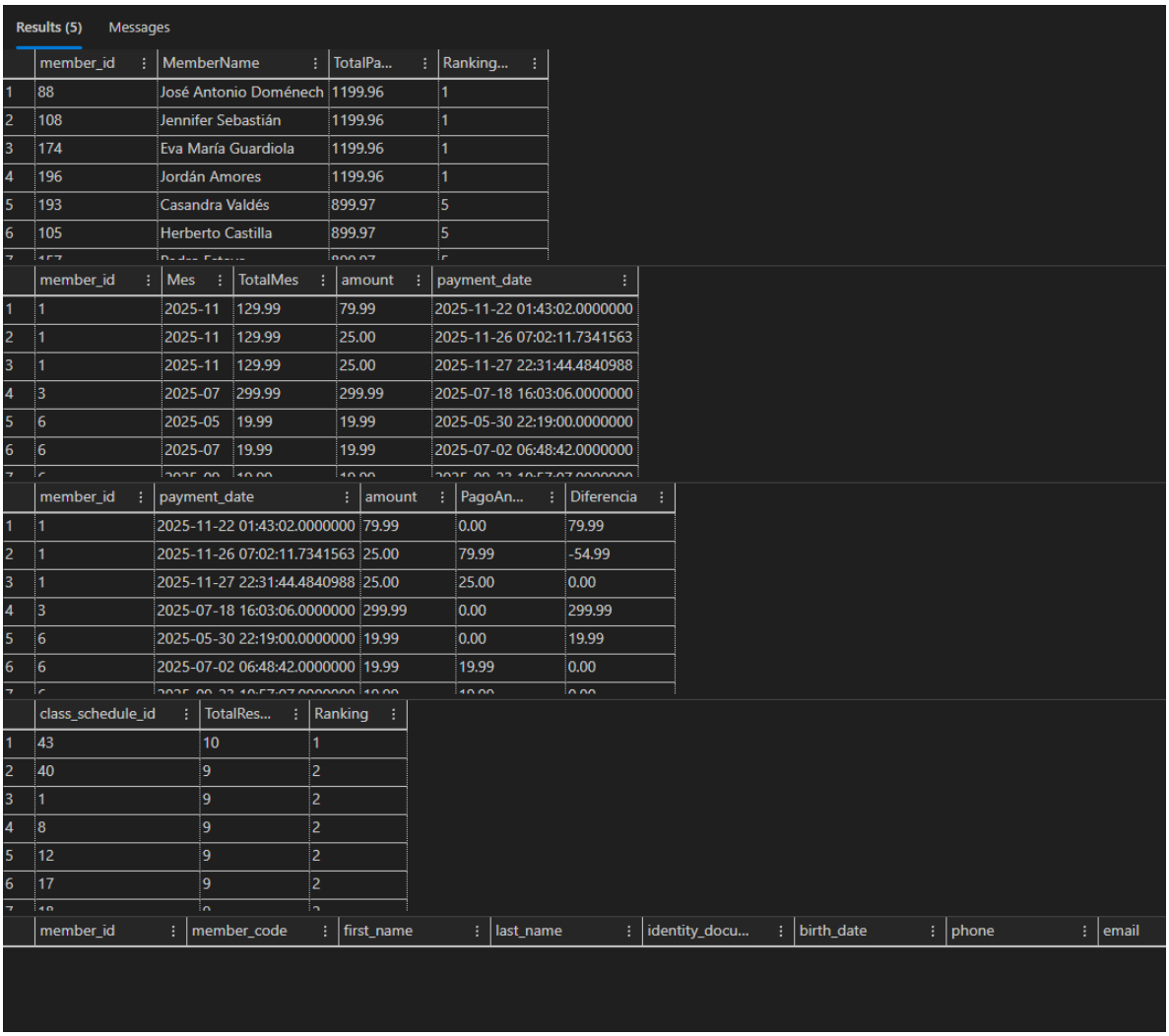
Para evitar procesar datos en la aplicación (lo cual es lento), se ha trasladado la lógica compleja al motor de base de datos utilizando Window Functions.

- Análisis de Tendencias de Pago (Función LAG):

El script implementado permite comparar el pago actual de un cliente con su pago inmediato anterior en una sola consulta. Esto es vital para detectar "Churn" (clientes que bajan su plan) o "Up-selling" (clientes que mejoran su plan).

- Ranking de Popularidad (Función DENSE_RANK):

Se utiliza para clasificar las clases con mayor asistencia. A diferencia de un COUNT simple, esta función permite manejar empates en el ranking de manera matemática correcta, proporcionando información de calidad para la toma de decisiones gerenciales.



Results (5) Messages

	member_id	MemberName	TotalPa...	Ranking...
1	88	José Antonio Doménech	1199.96	1
2	108	Jennifer Sebastián	1199.96	1
3	174	Eva María Guardiola	1199.96	1
4	196	Jordán Amores	1199.96	1
5	193	Casandra Valdés	899.97	5
6	105	Herberto Castilla	899.97	5

	member_id	Mes	TotalMes	amount	payment_date
1	1	2025-11	129.99	79.99	2025-11-22 01:43:02.0000000
2	1	2025-11	129.99	25.00	2025-11-26 07:02:11.7341563
3	1	2025-11	129.99	25.00	2025-11-27 22:31:44.4840988
4	3	2025-07	299.99	299.99	2025-07-18 16:03:06.0000000
5	6	2025-05	19.99	19.99	2025-05-30 22:19:00.0000000
6	6	2025-07	19.99	19.99	2025-07-02 06:48:42.0000000

	member_id	payment_date	amount	PagoAn...	Diferencia
1	1	2025-11-22 01:43:02.0000000	79.99	0.00	79.99
2	1	2025-11-26 07:02:11.7341563	25.00	79.99	-54.99
3	1	2025-11-27 22:31:44.4840988	25.00	25.00	0.00
4	3	2025-07-18 16:03:06.0000000	299.99	0.00	299.99
5	6	2025-05-30 22:19:00.0000000	19.99	0.00	19.99
6	6	2025-07-02 06:48:42.0000000	19.99	19.99	0.00

	class_schedule_id	TotalRes...	Ranking
1	43	10	1
2	40	9	2
3	1	9	2
4	8	9	2
5	12	9	2
6	17	9	2

	member_id	member_code	first_name	last_name	identity_docu...	birth_date	phone	email

---

4. ESTRATEGIA DE DIMENSIONAMIENTO, RESPALDO Y RECUPERACIÓN

Este capítulo define la infraestructura necesaria para garantizar la sostenibilidad y resiliencia del sistema a largo plazo.

4.1 Estrategia de Dimensionamiento (Capacity Planning)

Se ha realizado una proyección de crecimiento a 3 años para determinar los recursos de almacenamiento necesarios.

Supuestos del Escenario:

- Base de usuarios: 2,000 socios activos.
- Crecimiento anual de usuarios: 20%.
- Retención de datos históricos en línea: 2 años.

Análisis de Volumetría por Tabla Crítica:

Tabla	Tamaño Fila (Bytes)	Filas/Año Estimadas	Crecimiento Anual (MB)
Reservation	60 Bytes	312,000	18.72 MB
Payment	140 Bytes	48,000	6.72 MB
Member	400 Bytes	2,500	1.00 MB
Logs (LDF)	Variable	N/A	~2,000 MB

Justificación: Aunque las tablas de datos crecen moderadamente (~30 MB/año), el archivo de Log de Transacciones (.ldf) tendrá un crecimiento agresivo debido a la alta transaccionalidad.

Configuración de Archivos Físicos:

Para evitar la fragmentación del sistema de archivos NTFS, se establece la siguiente configuración inicial:

- **Data File (.mdf):** Tamaño inicial de 100 MB con *Autogrowth* de 50 MB.
- **Log File (.ldf):** Tamaño inicial de 100 MB con *Autogrowth* de 100 MB.

4.2 Métricas de Continuidad del Negocio

Se han definido los siguientes acuerdos de nivel de servicio (SLA) para el gimnasio:

- RPO (Recovery Point Objective): 30 Minutos.

Debido a la dinámica de las reservas de clases, no se permite perder más de 30 minutos de información. Una pérdida mayor ocasionaría overbooking (sobreventa) de clases y conflictos en la recepción.

- RTO (Recovery Time Objective): 2 Horas.

Es el tiempo máximo tolerado para restaurar el servicio antes de que las pérdidas operativas sean críticas.

4.3 Plan de Respaldos (Backup Strategy)

Para cumplir con el RPO de 30 minutos, se implementa una estrategia de respaldo mixta bajo el **Modelo de Recuperación Completa (Full Recovery Model)**.

4.3.1 Cronograma de Tareas Automatizadas (Jobs)

1. **Respaldo Completo (Full Database Backup):**
    - *Frecuencia:* Semanal (Domingos 02:00 AM).
    - *Descripción:* Copia total de la base de datos. Se realiza en horario de mínima actividad para no afectar el rendimiento.
2. **Respaldo Diferencial:**
    - *Frecuencia:* Diario (Lunes a Sábado 02:00 AM).
    - *Descripción:* Respalda solo los datos modificados desde el último Full. Esto agiliza el proceso de copia diario.
3. **Respaldo de Log de Transacciones (Transaction Log):**
    - *Frecuencia:* Cada 30 minutos (06:00 AM - 10:00 PM).
    - *Descripción:* Captura las transacciones individuales. Es la pieza clave que permite cumplir el RPO y liberar espacio en el archivo .ldf.

4.3.2 Política de Retención y Limpieza

Para garantizar la eficiencia del almacenamiento en disco, se implementa una política de "Ventana Deslizante":

- Los respaldos FULL y DIFF se conservan por **4 semanas**.
- Los respaldos de LOG se conservan por **48 horas**.
- Se utilizan *Maintenance Cleanup Tasks* para eliminar automáticamente los archivos que excedan estos periodos.

4.4 Plan de Recuperación ante Desastres (Disaster Recovery)

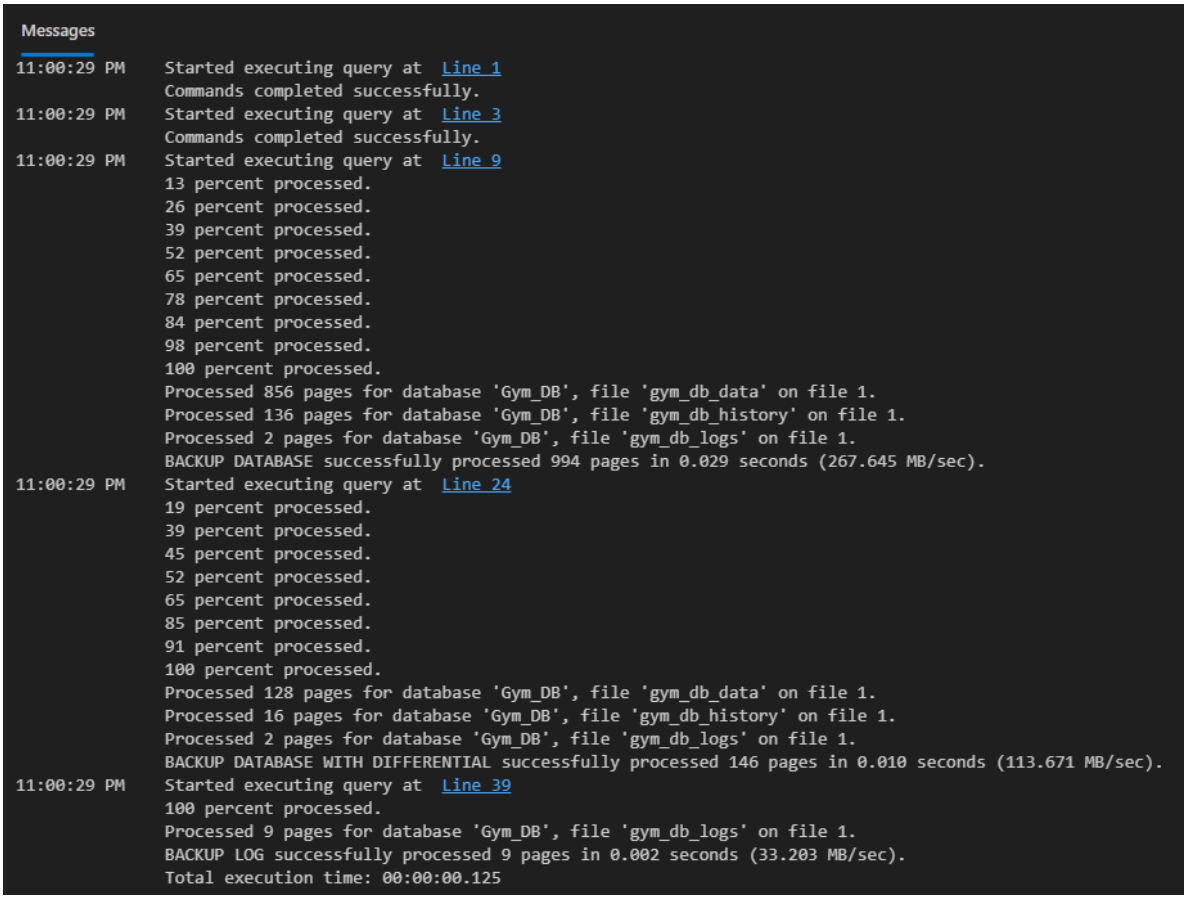
Se han documentado los procedimientos para los escenarios de fallo más probables.

Escenario: Corrupción Lógica o Error Humano

Situación: Un usuario elimina accidentalmente registros de pagos a las 10:15 AM.

Procedimiento de Recuperación (Point-in-Time Recovery):

1. Realizar un *Tail-Log Backup* de emergencia (con opción NO_TRUNCATE).
2. Restaurar el último Backup FULL con NORECOVERY.
3. Restaurar el último Backup DIFERENCIAL con NORECOVERY.
4. Restaurar la cadena de Logs secuenciales hasta las 10:00 AM.
5. Restaurar el último Log utilizando la cláusula STOPAT = 'Fecha 10:14:59' para recuperar el sistema al estado exacto previo al error.



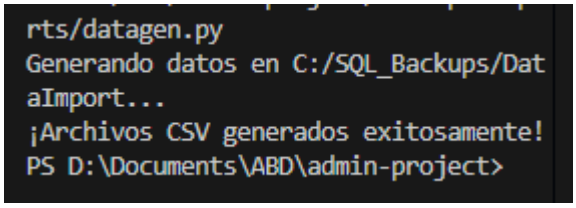
5. ESTRATEGIA DE MIGRACIÓN Y CARGA DE DATOS

La integridad de los datos es crítica al iniciar operaciones. Para cumplir con el requerimiento de carga masiva, se desarrolló una solución híbrida utilizando Python para la generación de datos y T-SQL para la ingesta de estos.

5.1 Generación de Datos Sintéticos (Mock Data)

Se desarrolló un script en Python (datagen.py) utilizando la librería **Faker** y **Pandas**. Este enfoque permitió:

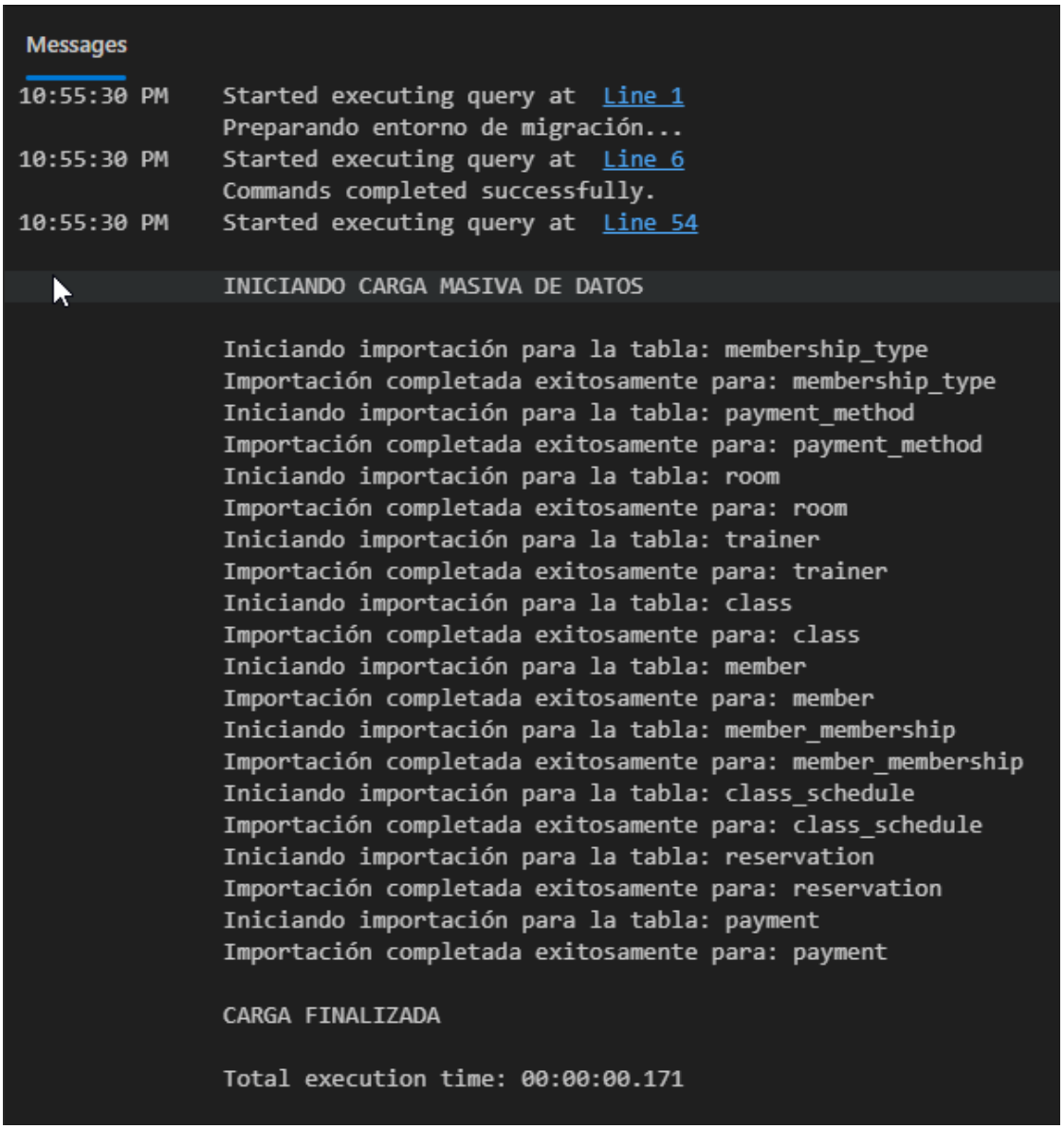
- **Volumen Realista:** Generar miles de registros de prueba coherentes (nombres latinos, DNI válidos, fechas lógicas).
- **Integridad Referencial:** El script garantiza que no se generen pagos para socios inexistentes o clases sin entrenador, respetando las llaves foráneas desde el origen.
- **Lógica de Negocio:** Se implementaron reglas para asignar precios correctos según el tipo de membresía, evitando inconsistencias financieras en la data de prueba.



5.2 Ingesta Masiva (Bulk Insert)

Para la inserción en SQL Server, se descartó el uso de INSERT INTO fila por fila debido a su lentitud. Se implementó el procedimiento almacenado usp_ImportarCSV que utiliza la instrucción **BULK INSERT**.

- **Manejo de Identidad:** El procedimiento detecta automáticamente tablas con columnas IDENTITY y aplica la opción KEEPIDENTITY, permitiendo migrar IDs históricos sin romper las relaciones.
- **Rendimiento:** Esta estrategia redujo el tiempo de carga de minutos a segundos, asegurando una ventana de migración mínima.



6. **INTELIGENCIA DE NEGOCIOS (DASHBOARD)**

Para transformar los datos transaccionales en información estratégica, se implementó una capa de visualización utilizando **Microsoft Power BI** conectado directamente al motor de base de datos.

6.1 Arquitectura de Datos

En lugar de cargar tablas crudas, se crearon **Vistas Materializadas** en SQL Server (vw_Dashboard_Ingresos, vw_Dashboard_Clases) que pre-procesan los cálculos. Esto reduce la carga de procesamiento en el reporte y centraliza la lógica de negocio en el servidor.

6.2 Visualizaciones Clave

El Tablero de Control Operativo responde a tres preguntas de negocio fundamentales:

A. Evolución Financiera (Ingresos Mensuales)

- **Visualización:** Gráfico de columnas agrupadas.
- **Razón de ser:** Permite a la gerencia identificar estacionalidad en los pagos y comparar el rendimiento de diferentes métodos de pago (Efectivo vs. Tarjeta por ejemplo).

B. Eficiencia Operativa (Top Clases)

- **Visualización:** Gráfico de barras horizontales.
- **Razón de ser:** Destaca las clases con mayor demanda (reservas), facilitando la decisión de abrir nuevos horarios o cancelar sesiones de baja asistencia.

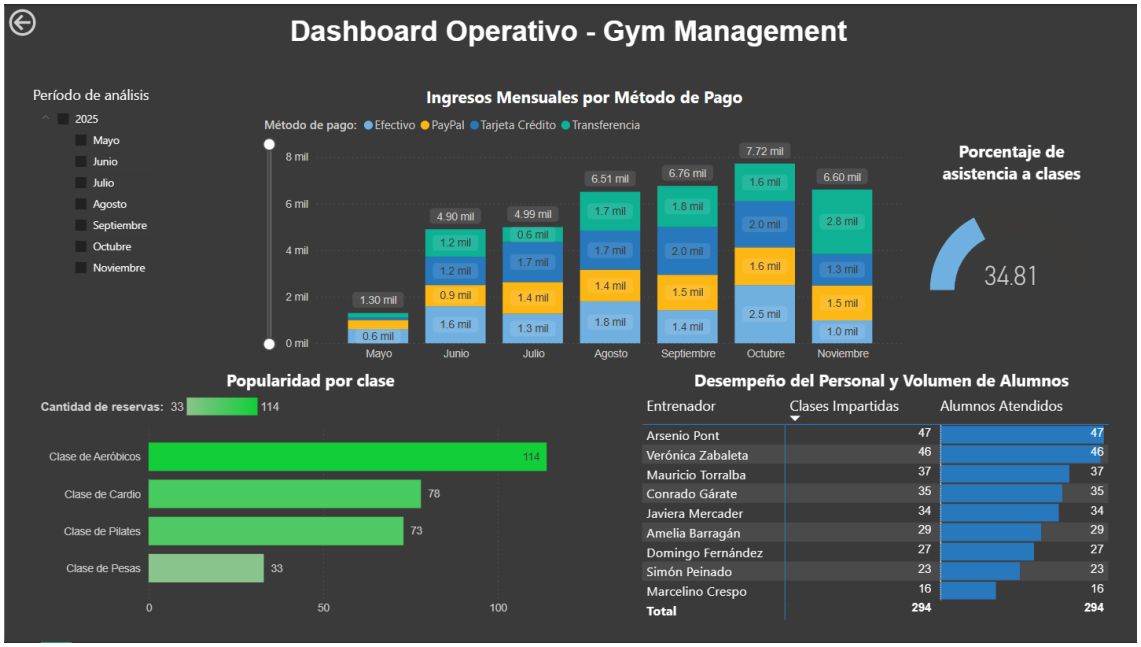
C. KPI de Compromiso (Tasa de Asistencia)

- **Visualización:** Medidor (Gauge).
- **Razón de ser:** Mide el porcentaje de socios que reservan y efectivamente asisten. Un KPI bajo alerta sobre la necesidad de implementar políticas de cancelación más estrictas.

D. Desempeño del personal (Datos por entrenador)

- **Visualización:** Gráfico de barras horizontales.
- **Razón de ser:** Mide las clases impartidas por entrenador y las junta con los alumnos atendidos por clase, facilitando la adquisición de datos del personal y

formación de métricas internas para mejorar la asignación de recursos.



CONCLUSIONES

- **Seguridad por Diseño:** La implementación de roles granulares demostró ser superior al uso de un superusuario único, reduciendo el riesgo de corrupción de datos accidental por parte del personal operativo.
- **Optimización de Almacenamiento:** La separación de índices en Filegroups secundarios (Gym_History_FG) y el uso de tipos de datos adecuados optimizan el I/O del servidor, preparando la base de datos para un crecimiento sostenido.
- **Valor del Dato:** La integración con Power BI valida que un modelo de datos normalizado (3FN) facilita la extracción de inteligencia de negocios sin requerir modelado de datos o transformaciones subsecuentes dentro de otro software.