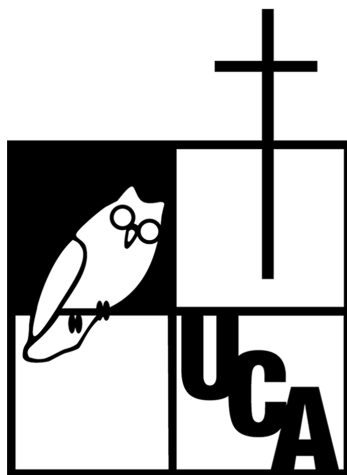


# Universidad Centroamericana José Simeón Cañas



## Proyecto Final - Administración de base de datos

### Integrantes:

**Marcos Alexi Caballero Lopez | 00103824**

**Hector Eduardo Gallardo Garcia | 00110924**

**Guillermo Alejandro Murcia Estrada Secc.02 | 00205324**

### Introducción:

#### 1. Descripción del sistema elegido y modelo de datos (ER).

La base de datos ha sido diseñada para administrar de forma integral las operaciones internas de un gimnasio, permitiendo un control eficiente de clientes, membresías, pagos, reservas, clases, entrenadores, salas, equipos y mantenimientos. Su estructura se fundamenta en un modelo relacional que garantiza la integridad de los datos, la coherencia entre entidades y la trazabilidad completa de las actividades registradas.

El sistema centraliza la información clave del gimnasio, facilitando la automatización de procesos esenciales como la inscripción de clientes, asignación de planes, programación de clases, seguimiento de pagos, administración de recursos y control del mantenimiento del equipo. Gracias a su arquitectura, el modelo garantiza que los datos almacenados se mantengan consistentes y actualizados, a la vez que soportan consultas y reportes que favorecen la toma de decisiones.

En primer lugar, la entidad Cliente registra la información personal y el estado de cada usuario del gimnasio. Esta se relaciona con el módulo de Membresías, donde se administran los planes contratados, sus fechas de vigencia y su estado actual. Cada membresía genera un historial financiero mediante la tabla Pago, que permite llevar un control preciso de los montos cobrados, fechas y estados de pago.

Complementariamente, el sistema contempla la planificación y ejecución de actividades mediante las entidades Entrenador, Sala y Clase, las cuales trabajan de forma conjunta para definir la oferta de clases disponibles, horarios, niveles, capacidad y responsables de impartirlas. Los clientes pueden registrarse en dichas actividades a través del módulo de Reserva, que permite validar disponibilidad, registrar asistencia y gestionar cancelaciones.

Por otra parte, el modelo incorpora un componente fundamental para la administración del entorno físico: el control del Equipo de gimnasio. La entidad correspondiente almacena información sobre cada máquina o aparato, incluyendo su código de inventario, estado y ubicación. A su vez, se integra con la tabla Mantenimiento, la cual registra las actividades preventivas y correctivas realizadas, permitiendo monitorear costos, fechas de intervención y condiciones operativas.

En conjunto, la base de datos ofrece una visión completa del funcionamiento del gimnasio, atendiendo no solo la gestión de los usuarios y servicios, sino también la administración de recursos humanos, infraestructura y equipamiento. El modelo garantiza la integridad referencial mediante llaves primarias y foráneas, aplica restricciones (CHECK, UNIQUE, DEFAULT) para asegurar la calidad de los datos y organiza las relaciones entre entidades bajo un enfoque 1:N coherente con los procesos reales del negocio.

De esta manera, la base de datos constituye una solución robusta, escalable y bien estructurada, capaz de soportar las necesidades operativas de un gimnasio moderno, proporcionando orden, control y confiabilidad en cada aspecto de su gestión diaria.

Nuestro sistema está compuesto por las siguientes entidades:

- **Entidades**

1. **plan:** define los planes de membresía que ofrece el gimnasio (nombre, precio, duración y cupo de clases).
2. **membresía:** registra la suscripción de un cliente a un plan, con fechas de inicio/fin y estado de la membresía.
3. **cliente:** almacena los datos personales y de contacto de las personas que usan el gimnasio y su estado (activo/inactivo).
4. **pago:** guarda los pagos realizados por las membresías, con fecha, monto y estado del pago (pagado, atrasado, etc.).
5. **reserva:** registra las reservas que hacen los clientes para asistir a una clase en una fecha y horario determinados, junto con su estado.
6. **clase:** describe las clases ofrecidas (título, nivel, cupo y estado), que luego se relacionan con reservas y entrenadores.

7. **entrenador:** contiene la información de los entrenadores (nombre, contacto, especialidad y estado laboral).
8. **sala:** representa las salas físicas del gimnasio donde se imparten las clases, con su capacidad y ubicación.
9. **equipo:** lista el equipo de entrenamiento del gimnasio, con su código de inventario, sala asociada, estado y fecha de compra.
10. **mantenimiento:** registra los mantenimientos realizados al equipo (tipo, estado, costos y fechas), vinculados a un equipo específico.

#### - Modelo entidad-relación

##### **Plan – Membresía (1:N)**

Un **plan** puede estar asociado a múltiples **membresías**, ya que diferentes clientes pueden contratar el mismo tipo de plan. Sin embargo, cada **membresía** pertenece exclusivamente a un solo **plan**, lo que asegura que las condiciones (precio, duración, cupo de clases, etc.) estén claramente definidas para cada suscripción.

##### **Membresía – Cliente (N:1)**

Un **cliente** puede acumular varias **membresías** a lo largo del tiempo, ya sea por renovaciones o por cambios de plan. No obstante, cada **membresía** está asociada a un único **cliente**, garantizando trazabilidad sobre quién es el titular de cada suscripción.

##### **Membresía – Pago (1:N)**

Una **membresía** puede registrar múltiples **pagos** durante su vigencia (por ejemplo, en esquemas fraccionados o cobros periódicos). Cada **pago**, sin embargo, se encuentra vinculado únicamente a una sola **membresía**, lo que permite identificar con precisión a qué suscripción corresponde cada transacción.

##### **Cliente – Reserva (1:N)**

Un **cliente** puede realizar numerosas **reservas** de clases a lo largo del tiempo. Cada **reserva**, por su parte, pertenece a un único **cliente**, lo que facilita el seguimiento del uso real que cada persona hace de los servicios del gimnasio.

##### **Reserva – Clase (N:1)**

Una **clase** puede tener muchas **reservas** asociadas, realizadas por distintos clientes. Sin embargo, cada **reserva** corresponde a una sola **clase** específica (en una fecha y hora determinadas), garantizando el control del cupo y la asistencia.

##### **Entrenador – Clase (1:N)**

Un **entrenador** puede impartir múltiples **clases** dentro de su horario laboral. Cada **clase**, no obstante, es impartida por un solo **entrenador**, lo que permite identificar claramente la responsabilidad de la sesión y analizar el desempeño de cada instructor.

##### **Clase – Sala (N:1)**

Una **sala** puede ser utilizada para muchas **clases** a lo largo del día y de la semana. Cada **clase**, sin embargo, se imparte en una única **sala**, lo que facilita la gestión de la ocupación de espacios y la planificación de horarios.

##### **Sala – Equipo (1:N)**

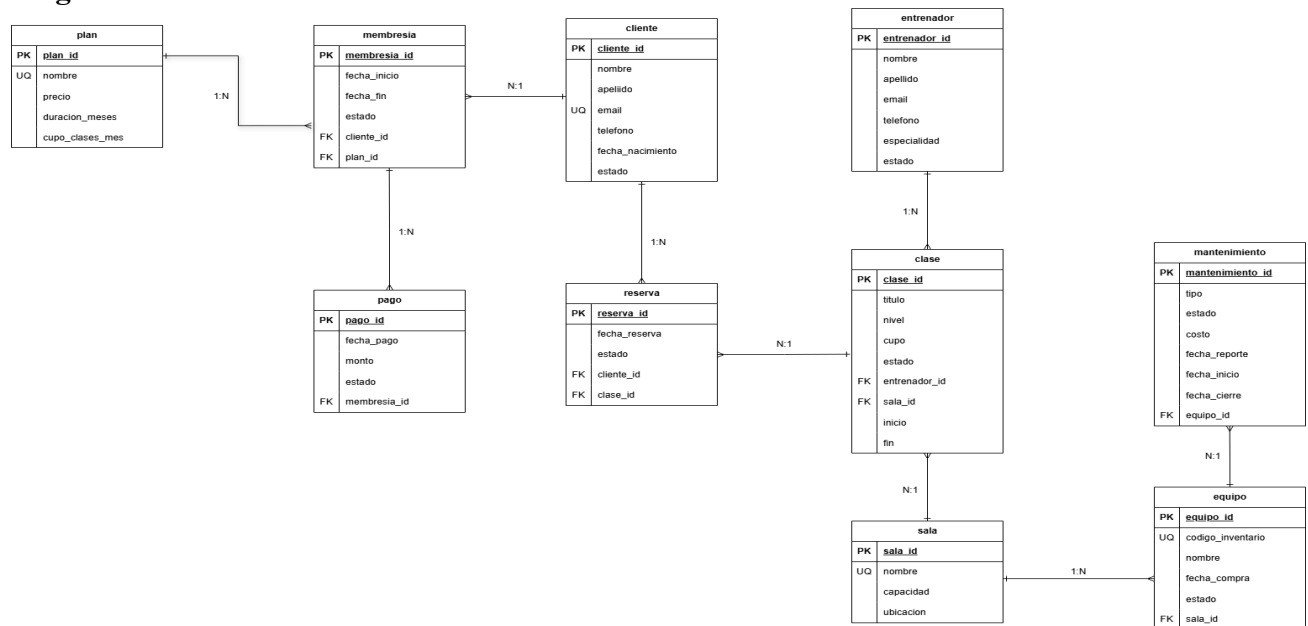
Una **sala** puede contener varios **equipos** (máquinas, accesorios, etc.). Cada **equipo** está asignado a una sola **sala**, lo que permite organizar el inventario físico y saber en qué ubicación se encuentra cada

recurso.

### Mantenimiento – Equipo (N:1)

Un **equipo** puede tener múltiples registros de **mantenimiento** a lo largo de su vida útil (correctivos, preventivos, revisiones periódicas). Cada registro de **mantenimiento** está asociado a un único **equipo**, lo que posibilita llevar un historial detallado de intervenciones, costos y fechas de servicio.

### Diagrama normalizado



## 2. Políticas de seguridad empleadas.

Se crean logins con el fin de asociarlos a usuarios, cada login cumple con políticas de seguridad del servidor (complejidad mínima y expiración periódica)

```
-----  
-- 1. CREACIÓN DE LOGINS  
-----  
USE master;  
GO  
  
-- Admin general del sistema  
CREATE LOGIN admin_gym  
WITH PASSWORD = 'EjemploAdmin245!',  
CHECK_POLICY = ON,  
CHECK_EXPIRATION = ON;  
GO
```

Así mismo, para cada login se le asignó un usuario:

```

-- 2. USUARIOS EN LA BASE DE DATOS

USE ROCAH_GYM_DB;
GO

CREATE USER admin_u FOR LOGIN admin_gym;
CREATE USER operacion_u FOR LOGIN recepcion_gym;
CREATE USER reportes_u FOR LOGIN reportes_gym;
GO

```

Se crea un esquema de roles para controlar el acceso de los datos:

Rol	Permisos	Descripción
rol_admin	CONTROL, en toda la base de datos	Permite el control total de toda la base de datos
rol_recepcion	SELECT, UPDATE y INSERT en las tablas cliente, reserva, membresía y pago	Rol dirigido a la operación del gimnasio, maneja la información de los clientes, reservas, membresías y pagos
rol_lectura	SELECT en todas las tablas de la base	Permite la lectura de la información de las tablas

Ejemplo de la creación de roles y la asignación de roles:

```

-- 3. ROLES EN LA BASE DE DATOS
--
CREATE ROLE rol_admin;
CREATE ROLE rol_recepcion;
CREATE ROLE rol_lectura;
GO

-- 4. PERMISOS POR ROL
--
-- 4.1 Rol administrador: control completo sobre la BD
GRANT CONTROL ON DATABASE::rocah_gym_db TO rol_admin;
GO

```

### 3. Evidencia de consultas optimizadas e índices aplicados.

Se crean índices estratégicos para mejorar el rendimiento de las consultas:

Índice	Tabla	Propósito
IX_pago_estado_membresia_fecha	pago	Optimizar las consultas de ingresos que filtran pagos por estado, membresía y fecha de pago.
IX_reserva_clase_estado	reserva	Agilizar el conteo de reservas por clase y estado (ocupación de clases).
IX_entrenador_estado	entrenador	Mejorar las búsquedas de entrenadores por estado (por ejemplo, solo activos) para calcular su desempeño.
IX_mantenimiento_equipo_estado_fecha	mantenimiento	Acelerar las consultas de mantenimientos por equipo, filtrando por estado y fecha de cierre para obtener costos y últimos mantenimientos.

```
CREATE NONCLUSTERED INDEX IX_pago_estado_membresia_fecha
ON dbo.pago (estado, membresia_id, fecha_pago)
INCLUDE (monto);
gd
```

```
SELECT * FROM pago;
```

MAS 48

SALIDA

CONSOLA DE DEPURACIÓN

TERMINAL

PUERTOS

RESU

dos (1)

Mensajes

42

La consulta comenzó a ejecutarse a las Línea 299

SQL Server parse and compile time:

CPU time = 0 ms, elapsed time = 2 ms.

(2000 filas afectadas)

Table 'pago'. Scan count 1, logical reads 21, physi

SQL Server Execution Times:

CPU time = 0 ms, elapsed time = 19 ms.

Tiempo total de ejecución: 00:00:00.043

```
SELECT * FROM pago;
```

MAS 48 SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS RESU

los (1) Mensajes

46 La consulta comenzó a ejecutarse a las Línea 300  
SQL Server parse and compile time:  
CPU time = 0 ms, elapsed time = 0 ms.  
(2000 filas afectadas)  
Table 'pago'. Scan count 1, logical reads 12, physi

SQL Server Execution Times:  
CPU time = 0 ms, elapsed time = 18 ms.  
Tiempo total de ejecución: 00:00:00.035

#### 4. Estrategia de dimensionamiento, respaldo y recuperación.

##### Estrategia de respaldos (BACKUP) de la base de datos

Para proteger la información de la base de datos ROCAH\_GYM\_DB se definió una estrategia de respaldos basada en el modelo de recuperación FULL. Esto permite combinar respaldos completos, diferenciales y de log de transacciones para reducir la pérdida de datos ante fallos y facilitar la recuperación en distintos puntos en el tiempo.

Los respaldos se almacenan en la ruta local: C:\backups\proyecto\_abd\

La estrategia implementada es la siguiente:

Backup FULL: se genera un respaldo completo de la base de datos, almacenado como ROCAH\_GYM\_DB\_full.bak. Este backup sirve como base para los diferenciales y como punto de recuperación principal ante desastres.

Backup DIFERENCIAL: a partir del backup FULL, se realizan respaldos diferenciales en ROCAH\_GYM\_DB\_diff.bak, que contienen solo los cambios desde el último FULL. Esto reduce el tamaño de los archivos intermedios y acelera el proceso de respaldo.

Backup del LOG de transacciones: se respalda el archivo de log en ROCAH\_GYM\_DB\_log.trn, permitiendo recuperar la base de datos con un nivel de granularidad más fino (por ejemplo, hasta los últimos minutos de operación) y evitando que el log crezca de forma indefinida.

Nombre	Fecha de modificación	Tipo	Tamaño
ROCAH_GYM_DB_diff.bak	27/11/2025 17:47	Archivo BAK	2,168 KB
ROCAH_GYM_DB_full.bak	27/11/2025 17:46	Archivo BAK	10,360 KB
ROCAH_GYM_DB_full_compa_20251127....	27/11/2025 19:01	Archivo BAK	10,360 KB
ROCAH_GYM_DB_log.trn	27/11/2025 21:30	Archivo TRN	368 KB

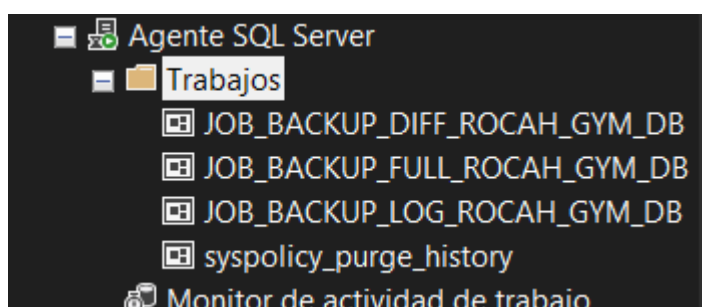
### Jobs de backup en SQL Server Agent

Para automatizar la estrategia de respaldo de la base de datos ROCAH\_GYM\_DB se configuraron tres jobs en SQL Server Agent:

**JOB\_BACKUP\_FULL\_ROCAH\_GYM\_DB:** realiza un backup FULL semanal de la base de datos. Está asociado al schedule SCH\_BACKUP\_FULL\_ROCAH\_GYM\_DB, que se ejecuta todos los domingos a las 02:00 a.m., garantizando un respaldo completo periódico.

**JOB\_BACKUP\_DIFF\_ROCAH\_GYM\_DB:** ejecuta un backup DIFERENCIAL diario, utilizando el archivo de backup FULL como base. Este job reduce el tamaño de los respaldos intermedios y permite recuperar información a nivel de día sin rehacer todo el backup completo.

**JOB\_BACKUP\_LOG\_ROCAH\_GYM\_DB:** realiza el backup del log de transacciones cada 30 minutos, entre las 06:00 y las 22:00 horas, mediante el schedule SCH\_BACKUP\_LOG\_ROCAH\_GYM\_DB. De esta forma se minimiza la pérdida de datos ante fallos y se controla el crecimiento del log.



### 5. Evidencia del dashboard de power bi.

Para explotar la información de la base de datos ROCAH\_GYM\_DB se diseñó un dashboard en Power BI compuesto por tres páginas temáticas: Ingresos y membresías, Clases y entrenadores y Mantenimiento y equipos. Cada página integra KPIs, filtros interactivos y visualizaciones orientadas a la toma de decisiones.

En cuanto a KPIs, se definieron medidas DAX conectadas a las vistas SQL, entre ellas: Total Ingresos Pagados (ingreso efectivo del gimnasio), Cantidad de Pagos, Clientes Totales y Total

Costo de Mantenimiento. Estos indicadores se muestran en tarjetas y se actualizan automáticamente al aplicar filtros.

El dashboard incorpora filtros como un slicer de año (sobre la vista vw\_ingresos\_mensuales\_por\_plan) y un slicer de sala (en la página de mantenimiento), además de interacciones cruzadas entre gráficos, lo que permite un análisis exploratorio según periodo, plan, sala o entrenador.

La narrativa del dashboard sigue un flujo lógico: primero los ingresos (Ingresos y membresías), luego el uso operativo de esos servicios (Clases y entrenadores) y finalmente los costos asociados a los activos (Mantenimiento y equipos).

Se utilizaron visualizaciones simples pero útiles: tarjetas KPI, gráficos de columnas agrupadas para ingresos por plan y mes, barras horizontales para ocupación de clases y costo por equipo, y tablas con ranking de entrenadores y equipos. Todas se basan en agregaciones definidas en SQL (vistas con funciones ventana y agrupaciones), asegurando coherencia entre la capa de datos y la capa de reporte. Este dashboard de Power BI esta en la sección de anexos.

## **6. Anexos:**

[1] Repositorio: <https://github.com/TFMurciaG/ProyectoABD>

[2] Dashboard Power BI:  
[https://ucaedusv-my.sharepoint.com/:u:/g/personal/00103824\\_uca\\_edu\\_sv/IQBe2ha2oH1FR5gsNXujXs1tAbcVDpHZRtV-ZOcvlQwcNHY?e=uiByZy](https://ucaedusv-my.sharepoint.com/:u:/g/personal/00103824_uca_edu_sv/IQBe2ha2oH1FR5gsNXujXs1tAbcVDpHZRtV-ZOcvlQwcNHY?e=uiByZy)

[3] Diccionario de datos:  
[https://ucaedusv-my.sharepoint.com/:w:/g/personal/00205324\\_uca\\_edu\\_sv/IQBnlbYSCIFRS6N3KGojVrSzAQ1E0YxEpK4a6tTDMswj0AE?e=o2S8XI](https://ucaedusv-my.sharepoint.com/:w:/g/personal/00205324_uca_edu_sv/IQBnlbYSCIFRS6N3KGojVrSzAQ1E0YxEpK4a6tTDMswj0AE?e=o2S8XI)