# Estructura de Base de Datos para Sistema de Gestión de Gimnasio

## 1. Descripción de las Tablas

### Tabla `users`

* user\_id (PK, UNIQUEIDENTIFIER): Identificador único del usuario.
* nombre (NVARCHAR): Nombre completo.
* email (NVARCHAR, UNIQUE): Correo electrónico.
* rol (NVARCHAR CHECK): Rol del sistema (admin, trainer, receptionist, client).
* token (NVARCHAR): Token de autenticación.
* foto\_url (NVARCHAR): URL de la foto de perfil.

Esta tabla almacena la información principal de las cuentas de los usuarios, gestionando sus credenciales, rol y datos de autenticación para controlar el acceso al sistema.

### Tabla `user\_profiles`

* profile\_id (PK, UNIQUEIDENTIFIER): Identificador del perfil.
* user\_id (FK, UNIQUEIDENTIFIER): Referencia a users.user\_id (1:1).
* edad (INT), peso (DECIMAL), altura (DECIMAL), imc (DECIMAL).
* historial\_medico (TEXT): Historia médica si aplica.
* fotos\_progreso (JSON): URLs de fotos de antes y después.

Esta tabla guarda datos adicionales del perfil de usuario, como características físicas, historial médico y fotografías de progreso, complementando la información básica de la tabla users.

### Tabla `memberships`

* membership\_id (PK, UNIQUEIDENTIFIER): Identificador de membresía.
* user\_id (FK): Referencia a users.user\_id (1:N).
* tipo (VARCHAR): Tipo de membresía.
* fecha\_inicio, fecha\_vencimiento (DATE).
* estado (ENUM CHECK): active, expired, pending.

Conserva el historial de membresías de cada usuario, incluyendo fechas de inicio y vencimiento, tipos de suscripción y estado actual, facilitando el seguimiento de renovaciones y caducidades.

### Tabla `attendance`

* attendance\_id (PK): Identificador de registro.
* user\_id (FK): users.user\_id (1:N).
* fecha (DATE), hora\_ingreso (TIME), hora\_salida (TIME).

Registra las entradas y salidas de los usuarios al gimnasio, con fecha y hora, lo que permite llevar un control preciso de la asistencia y generar estadísticas de uso.

### Tabla `capacity\_configs`

* config\_id (PK): Identificador de configuración.
* activo (BIT): Activa/inactiva.
* cupo\_max (INT): Límite de aforo.
* dia\_semana (ENUM): día de la semana.
* hora (TIME).

Configura las reglas de aforo máximas por día y hora, activables o desactivables según las necesidades del administrador, para controlar la ocupación del gimnasio.

### Tabla `training\_plans`

* plan\_id (PK): Identificador de plan.
* user\_id (FK): users.user\_id (1:N).
* titulo (VARCHAR), ejercicios (JSON), pdf\_url (VARCHAR).

Almacena los planes de entrenamiento diseñados para cada usuario, incluyendo el título, los ejercicios recomendados y la ubicación del PDF generado para su descarga.

### Tabla `diet\_plans`

* diet\_id (PK), user\_id (FK), objetivo (VARCHAR).
* calorias (INT), macros (JSON).

Contiene los planes nutricionales asociados a los usuarios según sus objetivos, detallando calorías y distribución de macronutrientes para el seguimiento dietético.

### Tabla `meal\_records`

* record\_id (PK), user\_id (FK), fecha (DATE), comidas (JSON).

Registra diariamente las comidas de cada usuario, guardando los detalles de los alimentos consumidos para análisis y seguimiento nutricional.

### Tabla `classes`

* class\_id (PK), nombre (VARCHAR), tipo (VARCHAR).
* horario (DATETIME), cupo\_max (INT).

Define las clases colectivas disponibles en el gimnasio, con información de nombre, tipo, horario y capacidad máxima, utilizada para la gestión de reservas.

### Tabla `reservations`

* reservation\_id (PK), user\_id (FK), class\_id (FK).
* fecha (DATE), estado (ENUM).

Guarda las reservas de los usuarios para las clases grupales, indicando la fecha, el estado de la reserva y enlazando al usuario y a la clase correspondiente.

### Tabla `trainer\_profiles`

* trainer\_id (PK), user\_id (FK, UNIQUE) – rol=trainer.
* evaluaciones (JSON).

Registra datos específicos de los entrenadores, vinculados a su cuenta de usuario, y almacena resúmenes de sus evaluaciones.

### Tabla `trainer\_clients`

* id (PK), trainer\_id (FK), client\_id (FK).
* Relacion N:M entre entrenadores y clientes.

Tabla intermedia que relaciona entrenadores con clientes, representando la asignación personalizada de usuarios a cada entrenador.

### Tabla `physical\_evaluations`

* eval\_id (PK), user\_id (FK).
* fecha (DATE), medidas (JSON), notas (TEXT).

Contiene los registros de evaluaciones físicas realizadas a los usuarios, con fecha, medidas corporales y notas de los entrenadores.

### Tabla `products`

* product\_id (PK), nombre, tipo, stock, precio.

Almacena el catálogo de productos disponibles para la venta en el gimnasio, como suplementos y merchandising, con sus características y precios.

### Tabla `inventory`

* inventory\_id (PK), product\_id (FK, UNIQUE), cantidad.

Gestiona el stock de cada producto, registrando la cantidad disponible y vinculándose al producto correspondiente para el control de inventario.

### Tabla `notifications`

* notification\_id (PK), user\_id (FK).
* tipo (VARCHAR), contenido (TEXT), fecha (DATETIME), leido (BIT).

Registra las notificaciones enviadas a los usuarios, tanto push como correo electrónico, con detalles de contenido, fecha y estado de lectura.

### Tabla `kpis`

* kpi\_id (PK), ingresos, asistencia\_diaria, membresias\_activas.
* estadisticas\_demograficas (JSON).

Guarda métricas clave del negocio, como ingresos, asistencia diaria y número de membresías activas, para alimentar dashboards de rendimiento y reportes ejecutivos.

## 2. Relaciones y Justificación

1. \*\*users ↔ user\_profiles\*\*: 1:1 – Cada usuario tiene un perfil único con datos extendidos.
2. \*\*users ↔ memberships\*\*: 1:N – Historial de membresías y renovaciones.
3. \*\*users ↔ attendance\*\*: 1:N – Registro de entradas y salidas para control de asistencia.
4. \*\*users ↔ training\_plans\*\*: 1:N – Un usuario puede tener múltiples planes a lo largo del tiempo.
5. \*\*users ↔ diet\_plans\*\*: 1:N – Planes nutricionales cambiantes según objetivos.
6. \*\*users ↔ meal\_records\*\*: 1:N – Seguimiento diario de comidas.
7. \*\*classes ↔ reservations\*\*: 1:N – Varias reservas por clase sujeta a cupo.
8. \*\*users ↔ reservations\*\*: 1:N – Un usuario puede reservar varias clases.
9. \*\*trainer\_profiles ↔ users\*\*: 1:1 – Perfil exclusivo para entrenadores.
10. \*\*trainer\_profiles ↔ trainer\_clients\*\*: 1:N – Un entrenador asigna varios clientes.
11. \*\*users ↔ trainer\_clients\*\*: N:1 – Tablas intermedia N:M entrenadores–clientes.
12. \*\*products ↔ inventory\*\*: 1:1 – Control de stock preciso por producto.
13. \*\*kpis\*\*: Tabla independiente que almacena métricas globales sin FK.

## 3. Posibles Usos en Procedimientos Almacenados

* sp\_InsertUser: Inserta un usuario y su perfil asociado.
* sp\_RenewMembership: Renueva o crea una nueva membresía.
* sp\_RecordAttendance: Registra check-in y check-out.
* sp\_CreateReservation: Reserva clase y valida cupo.
* sp\_GenerateTrainingPlanPdf: Invoca generación de PDF y actualiza URL.
* sp\_LogMealRecord: Añade registro de comidas diarias.
* sp\_UpdateStock: Ajusta inventario tras venta de productos.
* sp\_InsertPhysicalEvaluation: Registra evaluación física.

## 4. Posibles Usos para Reportería

* Reporte Asistencia Diaria: Conteo y detalle por usuario y fecha.
* Reporte Membresías Vencidas/Pendientes: Alertas periódicas.
* Informe de Ingresos: Suma de ventas de membresías y productos.
* Reporte Utilización de Clases: Ocupación vs cupo máximo.
* Dashboard KPIs: Visualización de métricas globales.
* Reporte Progreso Nutricional: Seguimiento de macros vs objetivos.