# Procedimientos Almacenados por Módulo

## Módulo de Autenticación y Roles

### SP\_CreateUser

Inserta un nuevo usuario en la tabla `users` y crea su perfil básico en `user\_profiles`.  
- \*\*Tablas involucradas\*\*: `users`, `user\_profiles`.  
- \*\*Parámetros\*\*: @nombre, @email, @rol, @passwordHash.  
- \*\*Salida\*\*: @newUserId.  
- \*\*Validaciones\*\*: verifica existencia previa de email, aplica hash de contraseña, asigna rol por defecto si no se proporciona.

### SP\_UpdateUserRole

Actualiza el rol de un usuario existente.  
- \*\*Tablas involucradas\*\*: `users`.  
- \*\*Parámetros\*\*: @userId, @newRole.  
- \*\*Validaciones\*\*: comprueba que el rol sea válido (ENUM), registra fecha de cambio en auditoría si corresponde.

### SP\_PasswordRecoveryService

Gestiona la recuperación de contraseña cuando el usuario la olvida.  
- \*\*Tablas involucradas\*\*: `users`, `notifications`.  
- \*\*Parámetros\*\*: @email.  
- \*\*Flujo\*\*: genera token temporal, lo guarda en `users.token`, envía notificación vía correo (insert en `notifications`).  
- \*\*Seguridad\*\*: expira token tras 24 horas, valida unicidad del token generado.

### SP\_ResetPassword

Permite al usuario establecer una nueva contraseña usando un token de recuperación.  
- \*\*Tablas involucradas\*\*: `users`.  
- \*\*Parámetros\*\*: @token, @newPasswordHash.  
- \*\*Flujo\*\*: valida token activo y no expirado, actualiza hash de contraseña, limpia campo `token`.  
- \*\*Seguridad\*\*: registra fecha de reinicio de contraseña en auditoría si aplica.

## Módulo de Gestión de Membresías

### SP\_CreateMembership

Crea una nueva membresía para un usuario.  
- \*\*Tablas involucradas\*\*: `memberships`.  
- \*\*Parámetros\*\*: @userId, @tipo, @fechaInicio, @fechaVencimiento.  
- \*\*Validaciones\*\*: asegura que no exista membresía activa solapada, establece estado inicial.

### SP\_RenewMembership

Renueva la membresía de un usuario expirado.  
- \*\*Tablas involucradas\*\*: `memberships`.  
- \*\*Parámetros\*\*: @membershipId, @nuevaFechaVencimiento.  
- \*\*Flujo\*\*: actualiza `fecha\_vencimiento`, cambia `estado` a active, registra historial de cambios.

### SP\_GetMembershipStatus

Obtiene el estado actual de la membresía de un usuario.  
- \*\*Tablas involucradas\*\*: `memberships`.  
- \*\*Parámetros\*\*: @userId.  
- \*\*Salida\*\*: estado, días restantes, fecha de vencimiento.

## Módulo de Control de Asistencia

### SP\_CheckInUser

Registra la hora de ingreso de un usuario al gimnasio.  
- \*\*Tablas involucradas\*\*: `attendance`, `capacity\_configs`.  
- \*\*Parámetros\*\*: @userId, @fecha, @hora.  
- \*\*Validaciones\*\*: comprueba que el aforo configurado lo permita, lanza error si cupo excedido.

### SP\_CheckOutUser

Registra la hora de salida de un usuario.  
- \*\*Tablas involucradas\*\*: `attendance`.  
- \*\*Parámetros\*\*: @attendanceId, @horaSalida.  
- \*\*Flujo\*\*: actualiza registro de `attendance`, calcula duración de sesión.

### SP\_GetAttendanceStats

Genera estadísticas de asistencia por usuario y rango de fechas.  
- \*\*Tablas involucradas\*\*: `attendance`.  
- \*\*Parámetros\*\*: @userId, @fromDate, @toDate.  
- \*\*Salida\*\*: número de entradas, promedio de tiempo de sesión, lista de fechas.

## Módulo de Rutinas y Planes de Entrenamiento

### SP\_CreateTrainingPlan

Inserta un nuevo plan de entrenamiento para un usuario.  
- \*\*Tablas involucradas\*\*: `training\_plans`.  
- \*\*Parámetros\*\*: @userId, @titulo, @ejercicios (JSON).  
- \*\*Salida\*\*: @newPlanId.  
- \*\*Validaciones\*\*: nombre único por usuario, formato JSON válido.

### SP\_UpdateTrainingPlan

Actualiza los detalles de un plan existente.  
- \*\*Tablas involucradas\*\*: `training\_plans`.  
- \*\*Parámetros\*\*: @planId, @titulo, @ejercicios.  
- \*\*Flujo\*\*: sobrescribe campos modificados, registra fecha de actualización.

### SP\_GenerateTrainingPlanPdf

Invoca la generación de PDF del plan de entrenamiento y actualiza su URL.  
- \*\*Tablas involucradas\*\*: `training\_plans`.  
- \*\*Parámetros\*\*: @planId.  
- \*\*Flujo\*\*: llama a servicio externo, guarda `pdf\_url` resultante.

## Módulo de Seguimiento Nutricional

### SP\_CreateDietPlan

Crea un plan nutricional para un usuario.  
- \*\*Tablas involucradas\*\*: `diet\_plans`.  
- \*\*Parámetros\*\*: @userId, @objetivo, @calorias, @macros (JSON).  
- \*\*Salida\*\*: @newDietId.  
- \*\*Validaciones\*\*: macros sumen valor aceptado, fechas de vigencia.

### SP\_UpdateDietPlan

Actualiza un plan nutricional existente.  
- \*\*Tablas involucradas\*\*: `diet\_plans`.  
- \*\*Parámetros\*\*: @dietId, @objetivo, @calorias, @macros.  
- \*\*Flujo\*\*: actualiza campos y fecha de modificación.

### SP\_LogMealRecord

Registra el consumo diario de comidas.  
- \*\*Tablas involucradas\*\*: `meal\_records`.  
- \*\*Parámetros\*\*: @userId, @fecha, @comidas (JSON).  
- \*\*Salida\*\*: @newRecordId.  
- \*\*Validaciones\*\*: evita duplicar registro para misma fecha.

## Módulo de Perfil de Usuario

### SP\_GetUserProfile

Obtiene la información completa de perfil de un usuario.  
- \*\*Tablas involucradas\*\*: `users`, `user\_profiles`.  
- \*\*Parámetros\*\*: @userId.  
- \*\*Salida\*\*: conjunto de datos combinado (JOIN).

### SP\_UpdateUserProfile

Actualiza datos adicionales del perfil de usuario.  
- \*\*Tablas involucradas\*\*: `user\_profiles`.  
- \*\*Parámetros\*\*: @profileId, @edad, @peso, @altura, @historialMedico.  
- \*\*Flujo\*\*: escribe cambios y retorna status de actualización.

## Módulo de Reservas de Clases

### SP\_CreateReservation

Reserva un cupo en una clase para un usuario.  
- \*\*Tablas involucradas\*\*: `reservations`, `classes`.  
- \*\*Parámetros\*\*: @userId, @classId, @fecha.  
- \*\*Validaciones\*\*: comprueba cupo disponible, estado de clase activa.

### SP\_CancelReservation

Cancela una reserva existente.  
- \*\*Tablas involucradas\*\*: `reservations`.  
- \*\*Parámetros\*\*: @reservationId.  
- \*\*Flujo\*\*: actualiza estado a `cancelled`, registra fecha de cancelación.

### SP\_RescheduleReservation

Reprograma una reserva a otra clase o fecha.  
- \*\*Tablas involucradas\*\*: `reservations`.  
- \*\*Parámetros\*\*: @reservationId, @newClassId, @newFecha.  
- \*\*Validaciones\*\*: evita solapamiento con otras reservas del usuario.

## Módulo para Entrenadores

### SP\_AssignClientToTrainer

Asigna un cliente a un entrenador.  
- \*\*Tablas involucradas\*\*: `trainer\_clients`.  
- \*\*Parámetros\*\*: @trainerId, @clientId.  
- \*\*Validaciones\*\*: previene duplicados, comprueba rol de usuario.

### SP\_RemoveClientFromTrainer

Desasigna un cliente de un entrenador.  
- \*\*Tablas involucradas\*\*: `trainer\_clients`.  
- \*\*Parámetros\*\*: @trainerId, @clientId.  
- \*\*Flujo\*\*: elimina fila correspondiente, controla integridad referencial.

### SP\_CreatePhysicalEvaluation

Registra una evaluación física para un usuario.  
- \*\*Tablas involucradas\*\*: `physical\_evaluations`.  
- \*\*Parámetros\*\*: @userId, @fecha, @medidas (JSON), @notas.  
- \*\*Salida\*\*: @newEvalId.

## Panel Administrativo

### SP\_GetKpis

Retorna las métricas globales del negocio.  
- \*\*Tablas involucradas\*\*: `kpis`.  
- \*\*Parámetros\*\*: @periodoInicio, @periodoFin.  
- \*\*Salida\*\*: conjunto de KPIs filtrado por rango.

### SP\_GenerateDailyAttendanceReport

Genera un reporte diario de asistencia.  
- \*\*Tablas involucradas\*\*: `attendance`, `users`.  
- \*\*Parámetros\*\*: @fecha.  
- \*\*Salida\*\*: lista de usuarios con hora de ingreso y salida.

## Gestión de Inventario y Productos

### SP\_AddProduct

Inserta un nuevo producto al catálogo.  
- \*\*Tablas involucradas\*\*: `products`.  
- \*\*Parámetros\*\*: @nombre, @tipo, @stock, @precio.  
- \*\*Salida\*\*: @newProductId.

### SP\_UpdateProductStock

Actualiza el stock de un producto tras venta o recepción.  
- \*\*Tablas involucradas\*\*: `inventory`.  
- \*\*Parámetros\*\*: @productId, @cantidadDelta.  
- \*\*Flujo\*\*: suma o resta cantidad, registra fecha de modificación.

## Notificaciones

### SP\_CreateNotification

Crea una notificación para un usuario.  
- \*\*Tablas involucradas\*\*: `notifications`.  
- \*\*Parámetros\*\*: @userId, @tipo, @contenido.  
- \*\*Salida\*\*: @newNotificationId.

### SP\_MarkNotificationAsRead

Marca una notificación como leída.  
- \*\*Tablas involucradas\*\*: `notifications`.  
- \*\*Parámetros\*\*: @notificationId.  
- \*\*Flujo\*\*: actualiza campo `leido` a 1.