

Gestión de Gimnasio/centro deportivo

1. Información de Contacto

- **Responsable:** Darío Suárez Domenech
- **Email:** dsd17@alu.ua.es
- **Teléfono:** +34 650725112

2. Explicación de la API

En este documento se detalla el uso e implementación de las APIs del gimnasio.

Se ofrecen servicios para la consulta pública de nuestra oferta comercial (planes y actividades) y servicios securizados para la verificación de socios en sistemas de terceros (interoperabilidad).

La finalidad de la API es permitir que sistemas externos (como comparadores de gimnasios, tiendas de suplementación deportiva o aplicaciones de salud) puedan obtener información actualizada sobre nuestros horarios, tarifas y validar el estado de la membresía de nuestros clientes.

3. URL Base y Autenticación

URL Base

La API se encuentra actualmente en entorno de desarrollo local, no la tenemos alojada en ningún servidor.

(Nota: Se actualizará cuando se realice el despliegue en el servidor de producción).

Autenticación

Para acceder a los *end-points* marcados como **(Securizada)**, es necesario identificarse mediante una **API Key**.

Esta clave debe enviarse en la cabecera (*Header*) de cada petición HTTP:

X-API-KEY: {tu_api_key_asignada}

Obtención de Credenciales

Actualmente, la generación de API Keys es manual. Los grupos de prácticas que deseen interoperar con el gimnasio deben solicitar su clave enviando un correo o un mensaje al responsable.

Para facilitar las pruebas durante esta fase, se ha habilitado la siguiente clave global:

gym_dev_key_2025

4. Listado de End-Points

A continuación se detallan los recursos disponibles.

4.1. Obtener Planes y Tarifas (No Securizada)

Devuelve el catálogo de suscripciones disponibles en el gimnasio, detallando precios y ventajas. Útil para comparadores de precios.

- Ruta: /planes
- Método: GET

Estructura de la Petición: No requiere parámetros.

Estructura de la Respuesta (JSON): Devuelve un array de objetos con la información de los planes.

```
[
  {
    "id": 1,
    "nombre": "Plan Básico",
    "precio_mensual": 30.00,
    "clases_gratis": 2,
    "descripcion": "Acceso a instalaciones + 2 clases extra"
  },
  {
    "id": 2,
    "nombre": "Plan Premium",
    "precio_mensual": 45.00,
    "clases_gratis": 6,
    "descripcion": "Acceso total + 6 clases extra"
  }
]
```

4.2. Consultar Agenda de Actividades (No Securizada)

Permite consultar las clases programadas (sesiones) filtrando por fecha o tipo de actividad.

- Ruta: /sesiones
- Método: GET

Parámetros (Query Params):

Nombre	Tipo	Rango/Formato	Descripción
fecha	String	YYYY-MM-DD	(Opcional) Filtra las clases de un día concreto.
actividad_id	Integer	ID válido	(Opcional) Filtra por tipo (ej: solo Yoga).

Ejemplo de Petición:

BASE_URL/api/sesiones?fecha=2025-10-12

Estructura de la Respuesta (JSON):

```
[
  {
    "id": 105,
    "actividad": "Spinning Intense",
    "sala": "Sala de Spinning",
    "monitor": "Ana García",
    "horario": {
      "inicio": "2025-10-12 10:00:00",
      "fin": "2025-10-12 11:00:00"
    },
    "plazas": {
      "totales": 20,
      "disponibles": 5
    },
    "coste_extra": 5.00
  }
]
```

4.3. Verificación de Socio (Securizada)

Este servicio permite a terceros comprobar si un usuario es socio del gimnasio. Es ideal para aplicar descuentos entre grupos.

- Ruta: /socios/validar
- Método: POST

Estructura de la Petición (Body JSON): Se debe enviar el email o el DNI del usuario a consultar.

```
{
  "email": "usuario@ejemplo.com"
}
```

Gestión de Errores:

- **401 Unauthorized:** Si la API Key es incorrecta o falta.
- **400 Bad Request:** Si el JSON está mal formado o falta el email.
- **404 Not Found:** Si el email no existe en la base de datos (Usuario no registrado).

Estructura de la Respuesta (JSON) - Caso Éxito (200 OK): Devuelve el estado de la membresía.

```
{
  "es_socio": true,
  "estado": "Activo", // Valores posibles: "Activo", "Bloqueado", "Pendiente"
  "plan_actual": "Plan Premium",
  "antiguedad": "2023-01-15"
}
```

Ejemplo de uso:

Si la Tienda Online recibe una compra de usuario@ejemplo.com, consulta este endpoint. Si es_socio es true y estado es Activo, le aplica un 10% de descuento, por ejemplo.

4.4. Consultar Aforo de Sala (Securizada)

Permite a sistemas de gestión o auditoría consultar la ocupación actual de una sala específica en tiempo real.

- Ruta: /salas/{id}/aforo
- Método: GET

Parámetros:

- id: Identificador numérico de la sala.

Estructura de la Respuesta (JSON):

```
{  
  "sala_id": 1,  
  "nombre": "Sala Musculación",  
  "capacidad_maxima": 50,  
  "ocupacion_actual": 32,  
  "porcentaje_ocupacion": "64%"  
}
```

4.5. Actividades disponibles (Securizada)

Este servicio permite al socio consultar el listado de actividades activas del gimnasio, con información sobre horarios, monitor y plazas disponibles.

- Ruta: /api/socio/actividades
- Método: GET

Estructura de la Petición:

No requiere parámetros ni body.

Estructura de la Respuesta (JSON) – Caso Éxito (200 OK):

Devuelve un listado de actividades disponibles.

```
{  
  "idActividad": 1,  
  "nombre": "Yoga",  
  "monitor": "Ana Pérez",  
  "fecha": "2025-03-20",  
  "hora": "18:00",  
  "plazasDisponibles": 5  
}
```

- **Gestión de Errores:**
 - 401 Unauthorized: El socio no está autenticado.
- 500 Internal Server Error: Error interno del servidor.

Ejemplo de uso:

La aplicación móvil del socio consulta este endpoint para mostrar el calendario de actividades disponibles y permitir su reserva.

4.6. Reserva de actividades (Securizada)

Permite al socio realizar la reserva de una actividad concreta siempre que existan plazas disponibles.

- Ruta: /api/socio/reservas
- Método: POST

Estructura de la Petición (Body JSON):

Se debe indicar el identificador de la actividad que se desea reservar.

```
{  
  "idActividad": 1  
}
```

Parámetros:

idActividad: integer – Identificador válido de una actividad existente.

Estructura de la Respuesta (JSON) – Caso Éxito (200 OK):

```
{  
  "mensaje": "Reserva realizada correctamente"  
}
```

- **Gestión de Errores:**

- 400 Bad Request: La actividad indicada no es válida.

- 401 Unauthorized: El socio no está autenticado.

- 409 Conflict: No hay plazas disponibles para la actividad seleccionada.

Ejemplo de uso:

Cuando el socio pulsa “Reservar” sobre una actividad concreta, la aplicación envía el idActividad a este endpoint.

4.7. Mis reservas (Securizada)

Permite al socio consultar todas sus reservas activas y cancelar una reserva existente.

- Ruta: /api/socio/reservas

- Métodos:

- GET – Lista todas las reservas del socio.

- DELETE – Cancela una reserva concreta.

Estructura de la Respuesta (GET – JSON):

```
{  
  "idReserva": 12,  
  "actividad": "Spinning",  
  "fecha": "2025-03-22",  
  "hora": "19:00"  
}
```

Parámetros (DELETE):

idReserva: integer – Identificador de una reserva perteneciente al socio.

- **Gestión de Errores:**

- 403 Forbidden: La reserva no pertenece al socio autenticado.

- 404 Not Found: La reserva no existe.

Ejemplo de uso:

El socio accede a su panel personal para revisar sus actividades reservadas o cancelar alguna que ya no pueda asistir.

4.8. Gestión de saldo (Securizada)

Permite al socio consultar su saldo disponible y realizar recargas económicas.

- Ruta: /api/socio/saldo
- Métodos:
 - GET – Consulta el saldo actual.
 - POST – Recarga el saldo del socio.

Estructura de la Respuesta (GET – JSON):

```
{  
  "saldo": 25.50  
}
```

Estructura de la Petición (POST – Body JSON):

```
{  
  "importe": 20  
}
```

Parámetros:

importe: number – Valor numérico mayor que 0.

- **Gestión de Errores:**
 - 400 Bad Request: El importe indicado no es válido.
- 401 Unauthorized: El socio no está autenticado.

Ejemplo de uso:

El socio recarga su saldo desde la aplicación para poder reservar actividades de pago sin usar efectivo.

4.9. Mi perfil (Securizada)

Permite al socio consultar y actualizar sus datos personales registrados en el sistema.

- Ruta: /api/socio/perfil
- Métodos:
 - GET – Consulta de datos del perfil.
 - PUT – Actualización de los datos personales.

Estructura de la Respuesta (JSON):

```
{  
  "nombre": "Marcos",  
  "email": "marcos@email.com",  
  "telefono": "600123123"  
}
```

- **Gestión de Errores:**

- 400 Bad Request: Los datos enviados no son válidos.

- 401 Unauthorized: El socio no está autenticado.

Ejemplo de uso:

El socio actualiza su teléfono o email desde su área personal para mantener sus datos siempre actualizados.