

# API REST - Reservas Municipales

Este documento describe los endpoints y contratos de la API, con énfasis en autenticación vía Azure AD y formatos de request/response. Se omiten hosts locales para facilitar su uso en distintos entornos.

## Autenticación

### Flujo de autenticación (Azure AD)

- El inicio de sesión se realiza mediante Azure Active Directory (OAuth 2.0 / OpenID Connect).
- El cliente obtiene un token de Azure (id\_token o access\_token) y lo intercambia por un JWT propio del sistema para consumir los endpoints.
- Ya no se admite el login por correo y contraseña en la documentación de la API.

### Intercambio de token (Azure → JWT del sistema)

- Método: POST
- Endpoint: [/api/auth/login](#)
- Headers: Content-Type: application/json
- Request (contrato):

```
{  
  "azureToken": "<AZURE_ID_TOKEN_O_ACCESS_TOKEN>"  
}
```

- Response (200 OK):

```
{  
  "token": "eyJhbGciOiJIUzI1NiJ9...",  
  "type": "Bearer",  
  "username": "user@tenant.com",  
  "roles": ["ROLE_USER"]  
}
```

## Autorización en endpoints protegidos

- Header requerido:

```
Authorization: Bearer <token>
```

- El [`<token>`](#) es el JWT emitido por este backend tras el intercambio con Azure.

## Convenciones

- Base path: `/api`
  - Formato de fechas: ISO-8601 con zona horaria, p. ej. `2025-10-25T14:00:00-06:00`
  - Códigos de estado: 2xx éxito, 4xx errores de cliente/validación, 5xx errores del servidor
- 

## Cómo usar la API (flujo típico)

1. Autentícate con Azure AD en el cliente (MSAL u otro) y obtén un `azureToken` (`id_token` o `access_token`).
2. Intercámbialo por un JWT del sistema con `POST /api/auth/login`.
3. Usa `Authorization: Bearer <token>` en todas las llamadas protegidas.
4. Descubre espacios: `GET /api/spaces` o filtros en `GET /api/spaces/search`.
5. Crea una reserva: `POST /api/reservations` (valida horarios RF15).
6. Confirma/ajusta la reserva: `PUT /api/reservations/{id}`.
7. Usa QR: obtener (`GET /api/reservations/{id}/qr`), validar (`POST /api/reservations/{id}/validate-qr`).
8. Exporta a Excel: `GET /api/reservations/export/excel` (o por `userId`).

### Formato estándar de error

```
{  
  "error": "<CODIGO_0_TIPO>",  
  "message": "<descripcion>",  
  "timestamp": "2025-10-28T10:30:00-06:00",  
  "path": "/api/endpoint",  
  "status": 400  
}
```

Notas:

- Los mensajes se devuelven en español.
- Algunos endpoints incluyen más campos (por ejemplo, validaciones específicas).

## Modelos de datos (resumen)

Estos son los campos principales que aparecen en las respuestas y solicitudes. Pueden existir campos adicionales según el contexto.

- Space
  - `id` (uuid)
  - `name` (string)
  - `spaceTypeId` (int)
  - `capacity` (int)
  - `location` (string)
  - `outdoor` (boolean)
  - `description` (string)
  - `active` (boolean)

- Reservation
  - `id` (uuid)
  - `spaceId` (uuid)
  - `userId` (uuid)
  - `startsAt` (datetime ISO-8601)
  - `endsAt` (datetime ISO-8601)
  - `status` (enum: PENDING, CONFIRMED, CANCELLED, COMPLETED)
  - `totalAmount` (number)
  - `currency` (string)
  - `createdAt` (datetime)
  - `cancellationReason` (string, opcional)
- Review
  - `reviewId` (number)
  - `spaceId` (uuid)
  - `userId` (uuid)
  - `reservationId` (uuid)
  - `rating` (int 1-5)
  - `comment` (string)
  - `visible` (boolean)
  - `createdAt` (datetime)
- User
  - `userId` (uuid)
  - `email` (string)
  - `fullName` (string)
  - `phone` (string)
  - `active` (boolean)
  - `roleCode` (string: ROLE\_ADMIN | ROLE\_SUPERVISOR | ROLE\_USER)

## Permisos por rol (resumen)

- ROLE\_ADMIN
  - Gestión completa de usuarios y roles (incluye `PATCH /api/users/change-role`)
  - Administración de espacios y horarios
  - Gestión total de reservas (crear, actualizar, cancelar sin restricción de tiempo, eliminar)
  - Acceso a dashboard y métricas
  - Exportación de datos de cualquier usuario
- ROLE\_SUPERVISOR
  - Visualización y gestión de reservas
  - Gestión de horarios de espacios
  - Acceso a dashboard y métricas
  - Exportación de datos de usuarios
  - Validación de códigos QR

- ROLE\_USER
  - Crear y gestionar sus propias reservas
  - Consultar espacios disponibles
  - Crear reseñas de espacios utilizados (post-uso)
  - Exportar sus propias reservas
  - Ver y usar códigos QR de sus reservas

## Reglas de negocio clave (resumen)

- Reservas y horarios (RF15): si un espacio tiene horarios, las reservas deben caer completamente dentro de algún bloque; si no hay horarios, se permite cualquier horario.
- Cancelación: por defecto, usuarios ROLE\_USER deben cancelar con  $\geq 24h$ ; ROLE\_ADMIN puede cancelar sin restricción. No se puede cancelar una reserva ya cancelada.
- Auto-cancelación: reservas PENDING con hora de inicio pasada se cancelan automáticamente cada ~5 minutos.
- Reseñas: solo para reservas CONFIRMED o COMPLETED, posterior al fin del periodo, y una reseña por reserva.

## Buenas prácticas del cliente

- Incluye siempre `Authorization: Bearer <token>` en endpoints protegidos y `Content-Type: application/json` en solicitudes con cuerpo.
- Usa fechas en ISO-8601 con zona horaria explícita. Recomendada UTC o zona local consistente en todo el flujo.
- Maneja correctamente 401 Unauthorized (token inválido/ausente) y 403 Forbidden (falta de permisos) mostrando mensajes orientativos al usuario.
- Ante validaciones de negocio (400/409), muestra el `message` devuelto por la API para guiar la corrección.
- Si el token expira, repite el intercambio con Azure (`POST /api/auth/login`) para obtener un JWT vigente.

## SpaceController

### Obtener todos los espacios

```
GET /api/spaces
```

### Obtener espacio por ID

```
GET /api/spaces/{id}
```

### Crear espacio

```
POST /api/spaces
Content-Type: application/json

{
  "name": "Parque Central",
  "spaceTypeId": 1,
  "capacity": 100,
  "location": "Centro de la ciudad",
  "outdoor": true,
  "description": "Parque con juegos infantiles"
}
```

## Actualizar espacio

```
PUT /api/spaces/{id}
Content-Type: application/json

{
  "name": "Parque Central Actualizado",
  "capacity": 150,
  "description": "Descripción actualizada"
}
```

## Desactivar espacio (Soft Delete - RECOMENDADO)

Solo ADMIN - Marca el espacio como inactivo sin eliminarlo de la base de datos

```
DELETE /api/spaces/{id}
```

Response exitoso (200 OK):

```
{
  "message": "Space deactivated successfully"
}
```

Características:

- NO elimina de la base de datos (cambia `active = false`)
- Reversible, mantiene historia y reservas

## Eliminar espacio permanentemente (Hard Delete - PELIGROSO)

Solo ADMIN - Elimina físicamente el espacio de la base de datos

```
DELETE /api/spaces/{id}/permanent
```

Response exitoso (200 OK):

```
{  
  "message": "Space permanently deleted"  
}
```

Response bloqueado por reservas (409 CONFLICT):

```
{  
  "error": "Cannot delete space",  
  "message": "Cannot delete space: it has 5 associated reservation(s). Please  
deactivate instead."  
}
```

Cuándo usar cada uno:

- Soft Delete: cierre temporal o desactivación lógica
- Hard Delete: limpieza de datos de prueba sin reservas asociadas

## Búsqueda avanzada de espacios

```
GET /api/spaces/search?name=parque  
GET /api/spaces/search?minCapacity=50&maxCapacity=200  
GET /api/spaces/search?outdoor=true  
GET /api/spaces/search?location=centro  
GET /api/spaces/search?spaceTypeId=1  
GET /api/spaces/search?name=parque&outdoor=true&minCapacity=100
```

## Buscar espacios disponibles por fechas

```
GET /api/spaces/available?startDate=2025-10-20T08:00:00-06:00&endDate=2025-10-  
20T18:00:00-06:00  
GET /api/spaces/available?startDate=2025-10-20T14:00:00-06:00&endDate=2025-10-  
20T16:00:00-06:00&minCapacity=50&spaceTypeId=1
```

---

## ReservationController

### Obtener todas las reservas

Roles: ADMIN, SUPERVISOR, USER

```
GET /api/reservations
```

Obtener reserva por ID

Roles: ADMIN, SUPERVISOR, USER

```
GET /api/reservations/{id}
```

Obtener reservas por usuario

Roles: ADMIN, SUPERVISOR, USER

```
GET /api/reservations/user/{userId}
```

Obtener reservas por espacio

Roles: ADMIN, SUPERVISOR, USER

```
GET /api/reservations/space/{spaceId}
```

Obtener reservas por estado

Roles: ADMIN

```
GET /api/reservations/status/{status}
```

Obtener reservas en rango de fechas

Roles: ADMIN, SUPERVISOR, USER

```
GET /api/reservations/date-range?startDate=2025-10-20T00:00:00-06:00&endDate=2025-10-30T23:59:59-06:00
```

Crear reserva

Roles: ADMIN, SUPERVISOR, USER

```
POST /api/reservations
Content-Type: application/json

{
  "spaceId": "[id_espacio]",
  "userId": "[id_usuario]",
  "startsAt": "2025-10-25T14:00:00-06:00",
  "endsAt": "2025-10-25T16:00:00-06:00",
  "status": "PENDING",
  "totalAmount": 15000.00,
  "currency": "CRC"
}
```

## Actualizar reserva

Roles: ADMIN, SUPERVISOR, USER

```
PUT /api/reservations/{id}
Content-Type: application/json

{
  "startsAt": "2025-10-25T15:00:00-06:00",
  "endsAt": "2025-10-25T17:00:00-06:00",
  "status": "CONFIRMED",
  "totalAmount": 20000.00
}
```

## Cancelar reserva

Roles: ADMIN, SUPERVISOR, USER Restricciones:

- Con al menos 24 horas de anticipación (configurable en `application-docker.yml`).
- USER: 24+ horas; ADMIN: sin restricción; no cancelar si ya está **CANCELLED**.

```
PATCH /api/reservations/{id}/cancel?reason=Usuario no puede asistir
```

Respuesta exitosa (200 OK):

```
(Sin contenido)
```

Respuesta tardía (403 FORBIDDEN):

```
{
  "error": "Cancelación no permitida",
```

```

    "message": "La cancelación debe realizarse con al menos 24 horas de anticipación. Actualmente faltan 18 horas para la reserva. Solo un ADMIN puede cancelar con menos anticipación.",
    "timestamp": "2025-10-23T10:30:00-06:00"
}

```

Ya cancelada (403 FORBIDDEN):

```

{
  "error": "Cancelación no permitida",
  "message": "Esta reserva ya ha sido cancelada previamente.",
  "timestamp": "2025-10-23T10:30:00-06:00"
}

```

## Eliminar reserva

```
DELETE /api/reservations/{id}
```

### ⌚ Auto-cancelación de Reservas Pendientes

- Tarea programada cada 5 minutos
- Cancela automáticamente reservas PENDING cuyo startsAt ya pasó
- Ajusta estado a CANCELLED y registra motivo

Ejemplo de motivo:

```
"Cancelada automáticamente - No se confirmó antes de la hora de inicio (25/10/2025 14:00)"
```

Estados: PENDING → CONFIRMED → CANCELLED → COMPLETED

### 📊 Exportar reservas a Excel (Usuario autenticado)

```
GET /api/reservations/export/excel
Authorization: Bearer <token>
```

Response:

- Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
- Content-Disposition: attachment; filename="reservas\_[usuario]\_[fecha].xlsx"
- Hojas: "Reservaciones" (detalle) y "Resumen" (estadísticas)

### 📊 Exportar reservas a Excel (Admin/Supervisor)

```
GET /api/reservations/export/excel/{userId}
Authorization: Bearer <token>
```

Columnas: ID, Espacio, Inicio, Fin, Estado, Monto, Moneda, Creación, Observaciones

#### 📱 Obtener código QR (JSON Base64)

```
GET /api/reservations/{id}/qr
```

#### 📷 Obtener QR como imagen PNG

```
GET /api/reservations/{id}/qr/image
```

#### ☑ Validar QR y marcar asistencia

```
POST /api/reservations/{id}/validate-qr
Content-Type: application/json
```

```
{
  "qrContent": "RESERVA:...",
  "validationToken": "cualquier-token"
}
```

#### 🔄 Regenerar código QR (ADMIN/SUPERVISOR)

```
POST /api/reservations/{id}/regenerate-qr
```

---

## ReviewController

### Obtener todas las reseñas

```
GET /api/reviews
```

### Obtener reseña por ID

```
GET /api/reviews/{id}
```

## Obtener reseñas por espacio

```
GET /api/reviews/space/{id}
```

## Obtener reseñas por usuario

```
GET /api/reviews/user/{id}
```

## Obtener estadísticas de un espacio

```
GET /api/reviews/space/{id}/statistics
```

Response:

```
{
  "spaceId": "[id]",
  "averageRating": 4.5,
  "totalReviews": 10,
  "ratingDistribution": {"1":0, "2":1, "3":2, "4":3, "5":4}
}
```

## Crear reseña

Restricciones:

- Solo reservas CONFIRMED o COMPLETED
- Después de la fecha de fin de la reserva
- Solo el usuario dueño de la reserva
- Una reseña por reserva

```
POST /api/reviews
Content-Type: application/json
```

```
{
  "spaceId": "[id]",
  "userId": "[id]",
  "reservationId": "[id]",
  "rating": 5,
  "comment": "Excelente espacio, muy limpio y bien equipado",
  "visible": true
}
```

## UserController - Gestión de Usuarios

### Obtener todos los usuarios

```
GET /api/users  
Authorization: Bearer <token>
```

### Obtener usuario por ID

```
GET /api/users/{id}  
Authorization: Bearer <token>
```

### Cambiar rol de usuario (Solo ADMIN)

El sistema envía un correo al usuario notificando el cambio.

```
PATCH /api/users/change-role  
Authorization: Bearer <admin_token>  
Content-Type: application/json  
  
{  
  "userId": "550e8400-e29b-41d4-a716-446655440000",  
  "roleCode": "ROLE_ADMIN"  
}
```

Roles válidos: [ROLE\\_ADMIN](#), [ROLE\\_SUPERVISOR](#), [ROLE\\_USER](#)

---

### Estados válidos de reservas

- PENDING
- CONFIRMED
- CANCELLED
- COMPLETED

### Rangos válidos de calificaciones

- Rating: 1-5 (1 = Muy malo, 5 = Excelente)

### Tipos de espacio

- 1: Parque
- 2: Salón Comunal
- 3: Campo Deportivo

## DashboardController - Métricas del Sistema

Obtener métricas del dashboard (ADMIN/SUPERVISOR)

```
GET /api/admin/dashboard  
Authorization: Bearer <token>
```

Response incluye 5 categorías de métricas (general, por estado, ingresos, top espacios, temporal), con cache de ~10 minutos.

## SpaceScheduleController - Horarios de Espacios (RF15)

Obtener horarios de un espacio

```
GET /api/spaces/{spaceId}/schedules  
Authorization: Bearer <token>
```

Response:

```
[  
  {  
    "scheduleId": 1,  
    "spaceId": "uuid-del-espacio",  
    "weekday": 1,  
    "weekdayName": "Monday",  
    "timeFrom": "08:00:00",  
    "timeTo": "12:00:00"  
  }  
]
```

Crear horario (ADMIN/SUPERVISOR)

```
POST /api/spaces/{spaceId}/schedules  
Authorization: Bearer <token>  
Content-Type: application/json  
  
{  
  "weekday": 1,  
  "timeFrom": "08:00:00",  
  "timeTo": "12:00:00"  
}
```

Horario por defecto:

- Si falta `timeFrom`: 06:00
- Si falta `timeTo`: 20:00

Días de la semana: 0=Dom, 1=Lun, 2=Mar, 3=Mié, 4=Jue, 5=Vie, 6=Sáb

### Eliminar horario específico (ADMIN/SUPERVISOR)

```
DELETE /api/spaces/{spaceId}/schedules/{scheduleId}
Authorization: Bearer <token>
```

### Eliminar todos los horarios (ADMIN)

```
DELETE /api/spaces/{spaceId}/schedules
Authorization: Bearer <token>
```

Validación en reservas:

- Si hay horarios, las reservas deben estar dentro de los bloques definidos
- Si no hay horarios, se permite cualquier horario (compatibilidad hacia atrás)

---

## WeatherController - Información del Clima

### Clima para espacio (solo `outdoor: true`)

```
GET /api/weather/space/{spaceId}
Authorization: Bearer <token>
```

### Clima por ubicación/ciudad

```
GET /api/weather/location?location={ciudad}
Authorization: Bearer <token>
```

Formato `location`:

- "Ciudad,PaísISO" (p. ej., "San Jose,CR", "London,GB")
- "Ciudad" (si es única)

### Health check (ADMIN)

```
GET /api/weather/health
Authorization: Bearer <admin_token>
```

---

## Notas técnicas (clima)

- Cache en memoria ~5 min (máx. ~500 entradas)
- Resilience4j: retry (x3), circuit breaker (50% fail → 30s), timeout (~2s), fallback
- Fuentes: [API](#) (OpenWeather), [CACHE](#), [FALLBACK](#)