

El RA que trabajaremos y CE son:

RA3.- Diseña, desarrolla y consume API RESTful aplicando buenas prácticas y estrategias de seguridad.

API REST de Gestión Inmobiliario de Alquiler

Una inmobiliaria necesita un sistema para gestionar el alquiler de sus propiedades. La base de datos consta de tres tablas relacionadas: **Zonas, Inmuebles y Reservas**.

Tareas a realizar:

2. Endpoints Requeridos

ZONAS ([/api/zonas](#))

| Método | Endpoint | Descripción |
|--------|----------|----------------------------------|
| GET | / | Obtener todas las zonas |
| GET | /:idzona | Obtener una zona por ID |
| POST | / | Crear una nueva zona |
| PUT | /:idzona | Actualizar una zona completa |
| PATCH | /:idzona | Actualizar parcialmente una zona |
| DELETE | /:idzona | Eliminar una zona |

INMUEBLES ([/api/inmuebles](#))

| Método | Endpoint | Descripción |
|--------|---------------|---|
| GET | / | Obtener todos los inmuebles |
| GET | /:idinmueble | Obtener un inmueble por ID |
| GET | /zona/:idzona | Obtener inmuebles de una zona específica |
| GET | /buscar | Buscar inmuebles por zona y rango de precio |
| POST | / | Crear un nuevo inmueble |
| PUT | /:idinmueble | Actualizar un inmueble completo |
| PATCH | /:idinmueble | Actualizar parcialmente un inmueble |
| DELETE | /:idinmueble | Eliminar un inmueble |

Endpoint Especial: Búsqueda Avanzada

GET `/api/inmuebles/buscar?zona=:idzona&precioMin=:min&precioMax=:max`

Parámetros query obligatorios:

- `zona`: ID de la zona
- `precioMin`: Precio mínimo
- `precioMax`: Precio máximo

Ejemplo: `/api/inmuebles/buscar?zona=1&precioMin=100000&precioMax=250000`

RESERVAS (`/api/reservas`)

| Método | Endpoint | Descripción |
|--------|------------------------------------|--|
| GET | <code>/</code> | Obtener todas las reservas |
| GET | <code>/:idreserva</code> | Obtener una reserva por ID |
| GET | <code>/inmueble/:idinmueble</code> | Obtener reservas de un inmueble |
| GET | <code>/dni/:dni</code> | Obtener reservas de un cliente (por DNI) |
| POST | <code>/</code> | Crear una nueva reserva |
| PUT | <code>/:idreserva</code> | Actualizar una reserva completa |
| DELETE | <code>/:idreserva</code> | Eliminar una reserva |

2. Validaciones Requeridas: Se realizarán con **express-validator**. Debes crear validadores diferentes según la operación HTTP.

Importante:

- **POST y PUT:** Todos los campos son obligatorios.
- **PATCH:** Todos los campos son opcionales, pero al menos uno debe ser enviado.
- Usa **body()** para validar datos del body.
- Usa **param()** para validar parámetros de ruta.
- Usa **query()** para validar query strings.
- Se mostrarán los mensajes adecuados según el error.

- **Zonas:**

- **descripcion:** Validaciones a aplicar:
 - No puede estar vacío.
 - Debe tener una longitud entre 3 y 50 caracteres.
 - Eliminar espacios al inicio y final (sanitización).

- Solo puede contener letras y espacios
 - Normalizar espacios múltiples a un solo espacio.
- **Inmuebles:**
- **zona:** Validaciones a aplicar:
 - No puede estar vacío.
 - Debe ser un número entero positivo (mayor que 0).
 - Convertir a entero después de validar.
 - **tipo_inmueble:** Validaciones a aplicar:
 - No puede estar vacío.
 - Debe ser uno de "Piso, Chalet, Casa, Adosado, Duplex, Ático, Estudio, Loft"
 - Eliminar espacios al inicio y final.
 - Normalizar la capitalización (primera letra en mayúsculas, resto minúsculas).
 - **domicilio:** Validaciones a aplicar:
 - No puede estar vacío.
 - Debe tener una longitud entre 10 y 100 caracteres.
 - Eliminar espacios al inicio y al final.
 - Solo puede contener letras, números, espacios, comas, puntos y guiones,
 - Normalizar espacios múltiples a un solo espacio.,
 - **habitaciones:** Validaciones a aplicar:
 - No puede estar vacío.
 - Debe ser un número entre 1 y 20.
 - Convertir a entero.
 - **banos:** Validaciones a aplicar:
 - No puede estar vacío.
 - Debe ser un número entre 1 y 10.
 - Convertir a entero.
 - **metros_cuadrados:** Validaciones a aplicar:
 - No puede estar vacío.
 - Debe ser un número decimal positivo (mayor que 0)
 - Valor máximo: 9999.99
 - Debe tener máximo 2 decimales
 - Convertir a decimal.
 - **precio:** Validaciones a aplicar:
 - No puede estar vacío.
 - Debe ser un número decimal positivo (mayor que 0)
 - Valor máximo: 99,999,999.99
 - Debe tener máximo 2 decimales
 - Convertir a decimal.
 - **reservado:** Validaciones a aplicar.
 - Puede ser opcional en POST
 - Debe ser 0 o 1
- **Reservas**
- **dni:** Validaciones a aplicar:

- No puede estar vacío.
- Eliminar espacios al inicio y final.
- Convertir a mayúsculas.
- Debe cumplir el formato: 8 dígitos seguidos de 1 letra
-
- **idInmueble:** Validaciones a aplicar:
 - No puede estar vacío.
 - Debe ser un número entero positivo
 - Convertir a entero.
- **Validaciones para el endpoint de Búsqueda**
 - Todos los campos son obligatorios y se validan con query().
 - **zona:**
 - No puede estar vacío.
 - Debe ser un número positivo.
 - Convertir a cero.
 - **precioMin y precioMax**
 - No puede estar vacío.
 - Debe ser un número decimal positivo o cero.
 - Convertir a decimal (float)
 - Debe tener máximo 2 decimal.
- **Validaciones de IDs en parámetros de ruta.**
 - **IdZona, idInmueble, idReserva:**
 - No puede estar vacío
 - Debe ser un número entero positivo.
 - Convertir a entero.
 - **dni:**
 - Eliminar espacios
 - Convertir a mayúsculas.
 - Debe cumplir el formato de DNI español