

Crear Propiedad

Permite registrar una nueva propiedad en el sistema de forma integral. A través de este servicio, es posible guardar la información básica de la propiedad, asociar propietarios (existentes o creados dinámicamente en la misma petición), registrar características personalizadas, coordenadas geográficas, videos de recorridos, fotos del inmueble y relacionar otras propiedades.

¿Para qué sirve este servicio?

Úsalo cuando necesites automatizar el ingreso de propiedades a nuby desde portales externos, aplicaciones móviles o sistemas de captación propios. Este endpoint está diseñado bajo un modelo transaccional robusto, asegurando que si algún paso crítico falla (por ejemplo, validación de un propietario o falta de características obligatorias del tipo de propiedad), toda la operación se cancele de forma automática para evitar datos inconsistentes en tu base de datos.

1. El Endpoint (La dirección web)

Apunta tu sistema a la siguiente dirección de petición POST. Recuerda reemplazar **{{instancia}}** por la dirección web completa que utilizas para ingresar a tu plataforma.

```
POST https://{{instancia}}/service/v2/public/properties
```

¿Qué debes colocar en {{instancia}}?

Es muy sencillo: corresponde a la **dirección web principal** que utilizas a diario para ingresar a tu plataforma (incluyendo la terminación `.nuby.app` o `.arrendasoft.co`).

Por ejemplo, si para entrar a tu sistema escribes `inmobiliaria.nuby.app` o `inmobiliaria.arrendasoft.co` en tu navegador, esa será exactamente tu instancia. Solo asegúrate de no incluir el "https://" ni barras diagonales ("/") al final.

2. La Petición (¿Qué debes enviarnos?)

Este servicio requiere autenticación mediante un Token JWT. Envía los encabezados requeridos y estructura la petición con un cuerpo (Body) en formato JSON que contenga las secciones detalladas a continuación:

Método	POST
Content-Type	application/json
Authorization	Bearer token , Token obtenido al consumir el servicio Login .

Autenticación requerida

Este servicio requiere un Token de autenticación válido. Debes incluir el encabezado `Authorization: Bearer TU_TOKEN` en cada petición. El token se obtiene consumiendo el servicio de **Login**. Adicionalmente, el cliente OAuth debe contar con el scope `write` para poder consumir este endpoint.

Cuerpo de la petición (JSON Body - Estructura General)

La raíz del JSON enviado debe estructurarse con las siguientes llaves principales:

Elemento Raíz	Tipo	Requerido	Descripción
Propiedad	object	Sí	Contiene la información básica del inmueble (Título, tipo, ubicación, valores, etc.).
Propietarios	array	Sí	Lista de propietarios asociados. Puede ser un arreglo de IDs existentes o de objetos con datos de nuevos propietarios.
Características	object	No	Mapeo de características de la propiedad en formato llave-valor <code>{"id_caracteristica": "valor"}</code> .
Videos	array	No	Listado de enlaces a videos del inmueble.
Fotos	array	No	Listado de imágenes del inmueble en formato base64 o URLs públicas.
PropiedadesRelacionadas	array	No	Arreglo de códigos numéricos de propiedades que se desea relacionar de forma recíproca.

Elemento Raíz	Tipo	Requerido	Descripción
Coordenadas	string object	No	Coordenadas de latitud y longitud. Ejemplo de texto: "6.2089, -75.5678".

Estructura Detallada de las Secciones

2.1 Sección "Propiedad" ?

Contiene la información comercial y técnica base del inmueble. Los siguientes campos son validados estrictamente bajo validación rápida (Fail-Fast):

Campo	Tipo	Requerido	Descripción
txtTitulo	string	Sí	Título descriptivo o publicitario del inmueble. Máximo 255 caracteres.
tipo_id	integer	Sí	ID de la clase de propiedad (Ej: apartamento, casa, local). Debe ser una clase existente en la maestra de clases de inmueble.
propiedad_tipo	string	Sí	Modalidad de comercialización. Valores válidos: "arriendo", "venta", o "venta y arriendo".
municipio_id	integer	Sí	ID de la ciudad o municipio donde se ubica el inmueble (debe existir en la maestra de municipios).
direccion	string	Sí	Dirección exacta del inmueble. El sistema validará que no exista otra propiedad con la misma dirección en el mismo municipio para evitar duplicados accidentales.
estrato	integer	Sí	Estrato socioeconómico del inmueble. Debe ser un número entero entre 1 y 6.
barrio_id	integer	No	ID del barrio (debe existir en la maestra de barrios de la ciudad).
urbanizacion	string	No	Nombre del edificio, conjunto cerrado o urbanización.

Campo	Tipo	Requerido	Descripción
valor_arriendo	numeric	No	Canon mensual del arriendo (monto mayor o igual a 0).
valor_venta	numeric	No	Precio de venta solicitado (monto mayor o igual a 0).
valor_administracion	numeric	No	Costo de la cuota de administración de la copropiedad.
propiedad_area	numeric	No	Área privada del inmueble en metros cuadrados. Se envía estrictamente con el nombre físico de columna.
observaciones	string	No	Descripción comercial extensa del inmueble.
llaves_en	string	No	Ubicación física de las llaves. Valores válidos: "oficina", "propiedad", o "otro".
paga_cuota_sost	string	No	Establece quién asume el pago de la cuota de administración. Valores válidos: "propietario" o "inquilino".
folio_matricula	string	No	Número de matrícula inmobiliaria del inmueble.
referencia_catastral	string	No	Número de referencia catastral asignado por el municipio.

2.2 Sección "Propietarios" ?

Cada propiedad en nuby requiere obligatoriamente tener asociado al menos un propietario. El arreglo acepta dos formatos de datos, los cuales pueden combinarse:

Formato A: Propietario Existente (Por ID)

Si el propietario ya se encuentra registrado en el ERP nuby, simplemente envía su identificador único (ID de tercero) como un entero dentro del arreglo:

```
"Propietarios": [1528, 4390]
```

Formato B: Creación Dinámica de Propietario (Objeto de Datos)

Si el propietario no existe, puedes enviar sus datos completos estructurados en un objeto. El sistema validará su información, lo creará automáticamente en la base de datos de nuby, y lo asociará al inmueble bajo una misma transacción comercial:

Campo	Tipo	Requerido	Descripción
persona	integer	Sí	Naturaleza jurídica del tercero: <input type="checkbox"/> 1 = Persona Natural, <input type="checkbox"/> 2 = Persona Jurídica.
documento	string	Sí	Número de documento de identidad o NIT.
tipo_doc_id	integer	Sí	ID de la maestra de tipos de documentos (Cédula, NIT, Pasaporte, etc.).
nombre1	string	Sí	Primer nombre del propietario o razón social completa.
apellido1	string	Sí (para Natural)	Primer apellido del propietario (requerido si <input type="checkbox"/> persona = 1).
direccion1	string	Sí	Dirección de domicilio del propietario.
municipio_id	integer	Sí	ID del municipio de residencia (debe existir en la base de datos).
email	string	Sí	Correo electrónico para envío de facturas and notificaciones. Debe tener formato de email válido.
telefono	string	Sí	Número telefónico o celular de contacto.
tipo_persona_id	integer	Sí (para Natural)	ID de tipo de persona tributaria (requerido si <input type="checkbox"/> persona = 1).
regimen_id	integer	Sí (para Natural)	ID del régimen tributario del propietario (requerido si <input type="checkbox"/> persona = 1).
resp_fiscal_id	integer	Sí (para Natural)	ID de la responsabilidad fiscal (requerido si <input type="checkbox"/> persona = 1).

2.3 Sección "Características" ??

Permite registrar valores para las características personalizadas del inmueble. Se envía en formato de objeto clave-valor, donde la clave es el ID numérico de la característica y el valor es el contenido a asignar:

```
"Caracteristicas": {
  "1": "3",    // ID 1 (Habitaciones) = 3
  "2": "2",    // ID 2 (Baños) = 2
  "31": "si",  // ID 31 (Red de gas) = si (Obligatoria)
  "63": "1"   // ID 63 (Garaje) = 1 (Checkbox de Garaje activo)
}
```

Validación de Características Obligatorias por Tipo

El motor de validación del sistema nuby verificará qué características están configuradas en el ERP como **obligatorias** para el tipo de propiedad seleccionado (`tipo_id`). Si omites alguna de estas características obligatorias en la petición, o el formato del valor es erróneo (por ejemplo, enviar texto en un campo netamente numérico), el endpoint devolverá un código de estado `422 Unprocessable Entity` y se revertirá todo el proceso de guardado.

Nota: En una configuración estándar, el apartamento (`1247`) y la casa (`1249`) exigen de forma obligatoria las características de: N° de Habitaciones (ID 1), N° de Baños (ID 2) y Red de gas (ID 31).

2.4 Sección "Fotos" ?

Permite adjuntar imágenes del inmueble de forma directa. El procesamiento de imágenes está automatizado y realiza optimizaciones de rendimiento y de presentación visual (redimensionamiento, corrección de orientación por metadatos EXIF, e inserción de la marca de agua corporativa configurada en el sistema).

Campo	Tipo	Requerido	Descripción
url	string	Sí	Contenido de la imagen. Puede enviarse como una cadena Base64 válida (<code>data:image/jpeg;base64,...</code>) o como una URL accesible públicamente para su descarga.
nombre	string	No	Nombre del archivo o descripción física para almacenar el documento.

2.5 Sección "Videos" ?

Permite registrar recorridos en video de las propiedades (por ejemplo, cargados en plataformas como YouTube o Vimeo):

Campo	Tipo	Requerido	Descripción
video_url	string	Sí	URL completa o código físico del video. Ejemplo: <code>"https://www.youtube.com/watch?v=..."</code> .
video_tipo	string	No	Tipo o plataforma emisora del video (Ej: <code>"youtube"</code> o <code>"vimeo"</code>).
video_descripcion	string	No	Descripción del video.
video_posicion	integer	No	Posición de visualización de este video.

2.6 Sección "PropiedadesRelacionadas" y "Coordenadas" ?

- **PropiedadesRelacionadas** → Permite enlazar propiedades entre sí (ideal para bodegas subdivididas, oficinas del mismo centro de negocios, etc.). Recibe una lista de códigos de inmuebles: `[137, 1042]`. El sistema nuby creará una asociación **recíproca**: la propiedad creada apuntará a estas asociadas, y estas asociadas apuntarán automáticamente a la nueva propiedad.
- **Coordenadas** → Coordenadas de geolocalización. Se recomienda enviar como una cadena de texto separada por comas `"latitud,longitud"` (Ej: `"6.2089,-75.5678"`). El motor de nuby se encarga de analizar la cadena, validar su rango físico geográfico y guardarlo bajo el formato propietario del sistema.

3. La Respuesta (¿Qué te entregaremos?)

Esta sección describe la respuesta que recibirás del sistema cuando la creación del inmueble sea completamente exitosa.

Respuesta Exitosa (201 Created)

Se genera cuando la propiedad y todos sus componentes asociados han sido validados e insertados con éxito. Se retorna el identificador único físico asignado al inmueble:

```
{
  "error": false,
  "type": "success",
  "msg": "La propiedad ha sido creada exitosamente.",
```

```
"propiedad_id": 7786,  
"warnings": []  
}
```

¿Qué contiene la sección "warnings"?

Si se incluyen fotos o videos y el almacenamiento en la nube o procesamiento de marcas de agua tiene algún inconveniente no letal (por ejemplo, formato no soportado o redimensión fallida de una imagen en particular), el inmueble **se creará con éxito**, pero se devolverán advertencias en esta sección para que puedas corregir los archivos de forma manual. Esto no interrumpe el registro del inmueble.

4. Seguridad y Posibles Errores

El sistema realiza validaciones de autenticación, permisos (scopes) y estructura de datos. Si alguna falla, devolverá un error con su respectivo código HTTP y un mensaje descriptivo:

Código HTTP	Significado y Solución
400 Bad Request	<p>Causa: La estructura de la petición es incorrecta o faltan datos esenciales. Ocurre antes de que se intente procesar la lógica de negocio.</p> <p>Ejemplos comunes:</p> <ul style="list-style-type: none">No se envió el objeto <code>Propiedad</code> o el array <code>Propietarios</code>.El array <code>Propietarios</code> está vacío.Se omitieron campos obligatorios dentro de la sección <code>Propiedad</code> (ej: <code>municipio_id</code>, <code>direccion</code>).Se envió un tipo de dato incorrecto (ej: texto en un campo numérico como <code>estrato</code>). <p>Solución: Revisa el cuerpo (body) de tu petición JSON y compáralo con la estructura definida en la sección 2 de esta guía. Asegúrate de que todos los campos requeridos estén presentes y tengan el tipo de dato correcto.</p>
401 Unauthorized	<p>Causa: El Token JWT de autenticación no es válido o ha expirado.</p> <p>Solución: Vuelve a consumir el servicio de Login para generar un nuevo token de acceso y úsalo en el encabezado <code>Authorization</code>.</p>

Código HTTP	Significado y Solución
403 Forbidden	<p>Causa: El token es válido, pero el cliente OAuth con el que fue generado no tiene los permisos (scopes) necesarios para esta operación.</p> <p>Solución: Verifica la configuración de tu cliente OAuth en nuby y asegúrate de que tenga asignado el scope <code>write</code>.</p>
422 Unprocessable Entity	<p>Causa: La petición es sintácticamente correcta, pero incumple una regla de negocio del sistema.</p> <p>Ejemplos comunes:</p> <ul style="list-style-type: none"> • La <code>direccion</code> enviada ya existe para otra propiedad en el mismo <code>municipio_id</code>. • Se omitió una característica marcada como obligatoria para el <code>tipo_id</code> de la propiedad. • Se intentó crear un propietario persona natural omitiendo los campos fiscales requeridos. • Se usó un ID de un modelo que no existe (ej: un <code>municipio_id</code> o <code>tipo_id</code> inválido). <p>Solución: Lee con atención el <code>msg</code> del JSON de error. Te indicará exactamente qué regla de negocio se infringió para que puedas corregir los datos enviados.</p>

Operación Atómica y de Transacción Segura

nuby implementa transacciones anidadas en sus modelos de negocio. Si el endpoint responde con un código `422` (o un fallo 500 del servidor), **toda la creación física del inmueble es cancelada de forma automática (ROLLBACK)**. No tendrás propiedades huérfanas sin propietario ni registros incompletos en las tablas del sistema.

5. Ejemplos de integración

Aquí tienes ejemplos de código listos para que tus desarrolladores los adapten a tu plataforma:

Ejemplos de JSON Body

Ejemplo 1: Creación de propiedad con propietario existente

```
{
  "Propiedad": {
    "txtTitulo": "Penthouse Duplex El Poblado con Terraza",
    "tipo_id": 1247,
```

```

"propiedad_tipo": "arriendo",
"municipio_id": 1,
"direccion": "Carrera 35 # 10B - 120, Apartamento 1201",
"estrato": 1260,
"barrio_id": 5,
"valor_arriendo": 4800000,
"valor_administracion": 650000,
"urbanizacion": "Torres de San Lucas",
"observaciones": "Espectacular penthouse con vista de 360 grados, tina de hidromasajes
en terraza principal, 3 alcobas cada una con baño.",
"propiedad_area": 185.4,
"llaves_en": "oficina",
"paga_cuota_sost": "propietario"
},
"Propietarios": [
  1247
],
"Caracteristicas": {
  "1": "3", // Alcobas
  "2": "4", // Baños
  "31": "si", // Red de gas (Obligatoria)
  "63": "1" // Garaje (Checkbox de Garaje - ID 63 en nuby)
},
"Coordenadas": "6.205210,-75.561240"
}

```

Ejemplo 2: Creación de propiedad con propietario nuevo, videos y fotos

```

{
  "Propiedad": {
    "txtTitulo": "Casa de Campo en Llanogrande",
    "tipo_id": 1249,
    "propiedad_tipo": "venta",
    "municipio_id": 2,
    "direccion": "Vía Llanogrande Kilómetro 4, Parcelación La Sofía",
    "estrato": 1259,
    "valor_venta": 1250000000,
    "valor_administracion": 300000,
    "observaciones": "Hermosa casa de un solo nivel, amplias zonas verdes, deck con zona

```

```
BBQ, acabados campestres modernos.",
  "propiedad_area": 320.0,
  "paga_cuota_sost": "propietario"
},
"Propietarios": [
  {
    "persona": 1,
    "documento": "1024567890",
    "tipo_doc_id": 1,
    "nombre1": "Alejandro",
    "apellido1": "Restrepo",
    "direccion1": "Transversal 39B # 4G - 85",
    "municipio_id": 1,
    "email": "alejandro.restrepo@email.com",
    "telefono": "3104567890",
    "tipo_persona_id": 1,
    "regimen_id": 2,
    "resp_fiscal_id": 11
  }
],
"Caracteristicas": {
  "1": "4",
  "2": "5",
  "31": "si", // Red de gas (Obligatoria)
  "63": "1", // Garaje (Checkbox de Garaje - ID 63 en nuby)
  "8": "Sí"
},
"Videos": [
  {
    "video_url": "https://www.youtube.com/watch?v=abc123xyz",
    "video_tipo": "youtube",
    "video_descripcion": "Video aéreo con Dron de la parcelación"
  }
],
"Fotos": [
  {
    "url":
"
DhgGAWjR9awAAAABJRu5ErkJggg=="
  }
]
```

```
    "nombre": "foto_fachada_casa_campo.jpg"
  }
]
}
```

cURL

```
# Define tu token y tu instancia
TOKEN="TU_TOKEN_AQUI"
INSTANCIA="tu-inmobiliaria.nuby.app"

# Prepara el cuerpo de la petición utilizando estrictamente los nombres de campos físicos
PAYLOAD='{
  "Propiedad": {
    "txtTitulo": "Apartamento para prueba cURL",
    "tipo_id": 1247,
    "propiedad_tipo": "arriendo",
    "municipio_id": 127,
    "direccion": "Calle Falsa 123 via cURL",
    "estrato": 1258,
    "propiedad_area": 97
  },
  "Propietarios": [1],
  "Caracteristicas": {
    "1": "3",
    "2": "3",
    "31": "si",
    "63": "1"
  }
}'

curl -X POST "https://${INSTANCIA}/service/v2/public/properties" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${TOKEN}" \
-d "${PAYLOAD}"
```

PHP

```
<?php

$instancia = 'tu-inmobiliaria.nuby.app';
$token = 'TU_TOKEN_AQUI';

$url = "https://{ $instancia }/service/v2/public/properties";

$payload = [
    'Propiedad' => [
        'txtTitulo' => 'Propiedad creada desde PHP',
        'tipo_id' => 1247,
        'propiedad_tipo' => 'venta',
        'municipio_id' => 127,
        'direccion' => 'Avenida Siempreviva 742, PHP',
        'estrato' => 1258,
        'propiedad_area' => 97 // Nombre exacto de la columna física
    ],
    'Propietarios' => [1], // ID de un propietario existente
    'Caracteristicas' => [
        '1' => '3', // Habitaciones
        '2' => '3', // Baños
        '31' => 'si', // Red de gas (Obligatoria)
        '63' => '1' // Garaje (Checkbox de Garaje - ID 63 en nuby)
    ]
];

$ch = curl_init($url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($payload));
curl_setopt($ch, CURLOPT_HTTPHEADER, [
    'Content-Type: application/json',
    "Authorization: Bearer {$token}"
]);

$response = curl_exec($ch);
```

```
$http_code = curl_getinfo($ch, CURLINFO_HTTP_CODE);

curl_close($ch);

echo "Código de estado HTTP: {$http_code}\n";
echo "Respuesta del servidor:\n";
print_r($response);

?>
```

Python

```
import requests
import json

instancia = 'tu-inmobiliaria.nuby.app'
token = 'TU_TOKEN_AQUI'
url = f"https://{instancia}/service/v2/public/properties"

payload = {
    "Propiedad": {
        "txtTitulo": "Propiedad Creada desde Python",
        "tipo_id": 1247,
        "propiedad_tipo": "venta",
        "municipio_id": 127,
        "direccion": "Calle de Python, 101",
        "estrato": 1258,
        "propiedad_area": 97
    },
    "Propietarios": [1], # ID de un propietario existente
    "Caracteristicas": {
        "1": "3",
        "2": "3",
        "31": "si",
        "63": "1"
    }
}
```

```
headers = {
  "Content-Type": "application/json",
  "Authorization": f"Bearer {token}"
}

try:
  response = requests.post(url, headers=headers, data=json.dumps(payload))

  print(f"Código de estado HTTP: {response.status_code}")
  print("Respuesta del servidor:")
  print(response.json())

except requests.exceptions.RequestException as e:
  print(f"Ocurrió un error en la petición: {e}")
```

JavaScript (Fetch API)

```
const instancia = 'tu-inmobiliaria.nuby.app';
const token = 'TU_TOKEN_AQUI';
const url = `https://${instancia}/service/v2/public/properties`;

const payload = {
  "Propiedad": {
    "txtTitulo": "Propiedad Creada desde JavaScript",
    "tipo_id": 1247,
    "propiedad_tipo": "arriendo",
    "municipio_id": 127,
    "direccion": "Avenida JavaScript, Lote 5",
    "estrato": 1258,
    "propiedad_area": 97
  },
  "Propietarios": [1], // ID de un propietario existente
  "Caracteristicas": {
    "1": "3",
    "2": "3",
```

```

        "31": "si",
        "63": "1"
    }
};

async function crearPropiedad() {
    try {
        const response = await fetch(url, {
            method: 'POST',
            headers: {
                'Content-Type': 'application/json',
                'Authorization': `Bearer ${token}`
            },
            body: JSON.stringify(payload)
        });

        const data = await response.json();

        console.log(`Código de estado HTTP: ${response.status}`);
        console.log('Respuesta del servidor:');
        console.log(data);

    } catch (error) {
        console.error('Error en la petición:', error);
    }
}

crearPropiedad();

```

Power Query M (Excel / Power BI)

```

let
    instancia = "tu-inmobiliaria.nuby.app",
    token = "TU_TOKEN_AQUI",
    url = "https://" & instancia & "/service/v2/public/properties",

    payload = [
        Propiedad = [

```

```
        txtTitulo = "Propiedad desde Power Query",
        tipo_id = 1247,
        propiedad_tipo = "venta",
        municipio_id = 127,
        direccion = "Calle Power BI, 4.0",
        estrato = 1258,
        propiedad_area = 97
    ],
    Propietarios = {1}, // ID de un propietario existente
    Caracteristicas = [
        #"1" = "3",
        #"2" = "3",
        #"31" = "si",
        #"63" = "1"
    ]
],

jsonPayload = Json.FromValue(payload),

response = Web.Contents(url, [
    Headers = [
        #"Content-Type" = "application/json",
        #"Authorization" = "Bearer " & token
    ],
    Content = jsonPayload
]),

jsonResponse = Json.Document(response)
in
jsonResponse
```

Revisión #3

Creado el 6 julio 2026 16:19:24 por Isabel Higuita

Actualizado el 8 julio 2026 04:40:51 por Isabel Higuita