

Listar Terceros [BETA]

Funcionalidad en desarrollo y pruebas: este servicio está disponible en el ambiente de pruebas y todavía no debe usarse en producción. El funcionamiento, los campos y los mensajes de error pueden cambiar antes del despliegue definitivo.

Este servicio devuelve una **lista paginada de terceros** registrados en nuby, con sus datos principales de identificación y contacto. Puedes filtrar por documento exacto, por coincidencia parcial en el nombre y por tipo de documento.

Úsalo para **buscar un tercero del que no conoces el identificador interno**, para conciliar tus clientes con los de nuby o para construir un buscador en tu propia interfaz.

¿Para qué sirve este servicio?

Los casos de uso más frecuentes de este servicio son:

- **Buscar un tercero del que no conoces el identificador interno**, por documento exacto o por coincidencia parcial en el nombre.
- **Conciliar tus clientes con los que ya están en nuby.**
- **Construir un buscador en tu propia interfaz**, con paginación.
- **Obtener el** `tercero_id` **que necesitan los demás servicios de la API.**

1. El Endpoint (La dirección web)

Método y dirección

Dato	Valor
Método	GET
Recurso	/third-parties/list
Cuerpo de la petición	No se envía cuerpo

```
GET https://{{instancia}}/service/v2/public/third-parties/list
```

¿Qué debes colocar en `{{instancia}}`? La dirección web principal con la que ingresas a tu plataforma, incluyendo la terminación `.nuby.app` o `.arrendasoft.co`, sin `https://` y sin barras diagonales al final.

Encabezados requeridos

Encabezado	Valor
<code>Authorization</code>	<code>Bearer <token></code> , con el token obtenido en el servicio Login .

Permiso requerido (scope)

Este servicio es de **solo lectura** y exige el permiso `catalogs:read` en el cliente OAuth. Si el token es válido pero al cliente le falta el permiso, la respuesta es `403` con el código `RECURSO_NO_AUTORIZADO`.

Si aún no tienes un token, consulta la página [Login](#) del libro **API nuby V2**.

2. La Petición (¿Qué debes enviarnos?)

Todos los parámetros son opcionales y se envían en la dirección web (*query string*). Si no envías ninguno, recibes la primera página con los primeros 50 terceros ordenados por documento.

Parámetro	Tipo	Requerido	Valor por defecto	Descripción
<code>documento</code>	texto	No	—	Busca por documento exacto . Es el filtro más preciso: úsalo cuando ya conoces el documento del tercero.
<code>nombre</code>	texto	No	—	Busca una coincidencia parcial en <code>nombre1</code> , <code>nombre2</code> , <code>apellido1</code> y <code>apellido2</code> . No distingue entre nombres y apellidos.

Parámetro	Tipo	Requerido	Valor por defecto	Descripción
tipo_doc_id	entero	No	—	Filtra por tipo de documento exacto. Debe ser un entero mayor que cero.
page	entero	No	1	Número de página, empezando en 1.
page_size	entero	No	50	Cantidad de registros por página. Si envías un valor mayor a 1000, nuby lo ajusta a 1000.

Importante: `page` y `page_size` deben ser números positivos. Un valor en cero, negativo o con letras produce `400`; no se corrige automáticamente.

Los filtros se combinan. Si envías `documento` y `tipo_doc_id` al mismo tiempo, la búsqueda exige que se cumplan ambos.

Ejemplos de dirección

```
GET /third-parties/list?page=1&page_size=50
GET /third-parties/list?documento=900123456
GET /third-parties/list?nombre=acme&page_size=10
GET /third-parties/list?tipo_doc_id=1&page=2
```

Validaciones de parámetros

La consulta se valida antes de ejecutarse. Si un parámetro no cumple su regla, la respuesta es `400 VALIDATION_ERROR` y no se devuelve ningún tercero:

Parámetro	Regla	Mensaje 400
page	Entero positivo. 0, negativos o texto no son válidos.	El parámetro "page" debe ser un número positivo.
page_size	Entero positivo. Si supera 1000, nuby lo ajusta a 1000.	El parámetro "page_size" debe ser un número positivo.
tipo_doc_id	Entero positivo; se compara exactamente contra el tipo de documento.	El parámetro "tipo_doc_id" debe ser un número entero positivo.

Parámetro	Regla	Mensaje 400
<code>documento</code> , <code>nombre</code>	Texto libre. Se comparan como se describe en la tabla de filtros.	No generan 400 por formato.

Cuándo usar cada filtro

Filtro	Caso de uso	Precisión
<code>documento</code>	Buscar un tercero concreto del que ya conoces el documento.	Exacta: devuelve el registro con ese documento.
<code>nombre</code>	Búsqueda exploratoria cuando no tienes el documento.	Parcial sobre <code>nombre1</code> , <code>nombre2</code> , <code>apellido1</code> y <code>apellido2</code> ; puede devolver varios.
<code>tipo_doc_id</code>	Segmentar por tipo de documento (por ejemplo, solo NIT).	Exacta sobre el identificador del tipo.
<code>documento</code> + <code>tipo_doc_id</code>	Confirmar que el documento encontrado es del tipo esperado.	Exacta: los filtros se combinan con <code>AND</code> .
<code>page</code> / <code>page_size</code>	Recorrer un resultado largo sin traer todo de una vez.	No filtra: controla el tamaño y la posición de la página.

3. La Respuesta (¿Qué te entregaremos?)

La respuesta tiene una estructura uniforme para todas las consultas de la API: los datos van en `body` y la información de paginación en `pagination`.

```
{
  "status": 200,
  "message": null,
  "body": [
    {
      "tercero_id": 123,
      "documento": "900123456",
      "tipo_doc_id": 1,
      "tipo_doc": "NIT",
      "nombre_completo": "ACME SAS",
      "email": "cuenta@acme.co",
      "telefono1": "6040000000",
      "direccion1": "CL 1 #2-3",
      "municipio_id": 1,
      "tipo_tercero_id": "1,2"
    }
  ]
}
```

```
],
"pagination": {
  "page": 1,
  "page_size": 50,
  "total": 1,
  "total_pages": 1
},
"filters_applied": {
  "documento": "900123456"
}
}
```

Variantes de respuesta

El listado cambia el contenido de `filters_applied` y de `message` según los filtros y las coincidencias. Maneja estos tres casos:

Con filtros

```
{
  "status": 200,
  "message": null,
  "body": [
    {
      "tercero_id": 123,
      "documento": "900123456",
      "tipo_doc_id": 1,
      "tipo_doc": "NIT",
      "nombre_completo": "ACME SAS",
      "email": "cuenta@acme.co",
      "telefono1": "6040000000",
      "direccion1": "CL 1 #2-3",
      "municipio_id": 1,
      "tipo_tercero_id": "1,2"
    }
  ],
  "pagination": { "page": 1, "page_size": 50, "total": 1, "total_pages": 1 },
  "filters_applied": { "documento": "900123456" }
}
```

Sin filtros

Sin parámetros, `filters_applied` es `null` y recibes la primera página de todos los terceros, ordenados por documento ascendente.

```
{
  "status": 200,
  "message": null,
  "body": [
    {
      "tercero_id": 123,
      "documento": "900123456",
      "tipo_doc_id": 1,
      "tipo_doc": "NIT",
      "nombre_completo": "ACME SAS",
      "email": "cuenta@acme.co",
      "telefono1": "6040000000",
      "direccion1": "CL 1 #2-3",
      "municipio_id": 1,
      "tipo_tercero_id": "1,2"
    },
    {
      "tercero_id": 124,
      "documento": "901234567",
      "tipo_doc_id": 1,
      "tipo_doc": "NIT",
      "nombre_completo": "INVERSIONES ANDINAS SAS",
      "email": null,
      "telefono1": "6044445555",
      "direccion1": "CR 43A #1-50",
      "municipio_id": 1,
      "tipo_tercero_id": "1"
    }
  ],
  "pagination": { "page": 1, "page_size": 50, "total": 2, "total_pages": 1 },
  "filters_applied": null
}
```

Lista vacía

No es un error: se responde `200` con `body: []` y un mensaje informativo. Si enviaste filtros, `filters_applied` sigue reflejándolos.

```
{
  "status": 200,
  "message": "No se encontraron terceros con los filtros aplicados.",
  "body": [],
  "pagination": { "page": 1, "page_size": 50, "total": 0, "total_pages": 0 },
  "filters_applied": { "nombre": "zzz" }
}
```

Campos de cada tercero en la lista

Importante: la columna “Puede ser `null`” describe el contrato del endpoint para terceros creados o actualizados por la API. En terceros históricos de nuby algunos campos que hoy son obligatorios pueden estar vacíos o en `null`; no asumas que un campo marcado “No” siempre trae valor.

Campo	Tipo	Puede ser <code>null</code>	Descripción
<code>tercero_id</code>	entero	No	Identificador interno del tercero. Úsalo para consultar el detalle.
<code>documento</code>	texto	No	Número de documento o NIT.
<code>tipo_doc_id</code>	entero	No	Identificador del tipo de documento.
<code>tipo_doc</code>	texto	No	Nombre del tipo de documento (por ejemplo, NIT).
<code>nombre_completo</code>	texto	No	Nombre unificado: nombres y apellidos separados por espacios, sin duplicar espacios.
<code>email</code>	texto	Sí	Correo electrónico del tercero, si está registrado.
<code>telefono1</code>	texto	Sí	Teléfono principal, si está registrado.
<code>direccion1</code>	texto	Sí	Dirección principal, si está registrada.
<code>municipio_id</code>	entero	No	Identificador del municipio. El nombre del municipio se obtiene en el detalle.

Campo	Tipo	Puede ser <code>null</code>	Descripción
<code>tipo_tercero_id</code>	texto	Sí	Tipos del tercero en una sola cadena separada por comas, por ejemplo <code>"1,2"</code> . Es <code>null</code> si el tercero no tiene tipos.

Lista vs. detalle

Dato	En el listado	En el detalle
Nombres y apellidos separados	No: solo <code>nombre_completo</code>	Sí: <code>nombre1</code> , <code>nombre2</code> , <code>apellido1</code> y <code>apellido2</code>
Ubicación	Solo <code>municipio_id</code>	<code>municipio</code> , <code>departamento</code> y <code>pais</code>
Información tributaria	No	Sí
Tipos del tercero	Cadena <code>"1,2"</code>	Cadena y arreglo <code>[1, 2]</code>
Información bancaria	No	Sí

Paginación

Campo	Descripción
<code>page</code>	Página devuelta.
<code>page_size</code>	Cantidad de registros por página efectivamente aplicada.
<code>total</code>	Total de terceros que cumplen los filtros.
<code>total_pages</code>	Total de páginas disponibles. Cuando no hay resultados, este valor es <code>0</code> .

`filters_applied` y `message`

- Si **no enviaste filtros**, `filters_applied` es `null`.
- Si **enviaste filtros**, `filters_applied` es un objeto con los filtros usados.
- Si hay resultados, `message` es `null`.
- Si no hay resultados, `message` es `No se encontraron terceros con los filtros aplicados.` y `body` es una lista vacía `[]`.

Comportamiento de `filters_applied` y `message`

Situación	<code>filters_applied</code>	<code>message</code>
Sin filtros y con resultados	<code>null</code>	<code>null</code>

Situación	filters_applied	message
Con filtros y con resultados	Objeto con los filtros aplicados	null
Con filtros y sin resultados	Objeto con los filtros aplicados	No se encontraron terceros con los filtros aplicados.
Sin filtros y sin resultados	null	No se encontraron terceros con los filtros aplicados.

Campos del envelope

Campo	Contenido	Cuándo cambia
status	200 en una lectura exitosa.	Los errores usan su propio código HTTP y agregan error_code.
message	Texto informativo o null.	Deja de ser null cuando no hay coincidencias.
body	Arreglo de terceros con los campos del listado.	Es [] cuando no hay coincidencias.
pagination	Objeto con page, page_size, total y total_pages.	total y total_pages pasan a 0 sin resultados.
filters_applied	Objeto con los filtros usados, o null.	Es null solo cuando no enviaste ningún filtro.

Una lista vacía no es un error. Cuando ningún tercero coincide con los filtros, el servicio responde 200 con body: []. Revisa pagination.total para confirmar que no hay coincidencias.

Orden de los resultados: la lista se ordena por documento ascendente y, cuando hay documentos repetidos, por tercero_id ascendente. El orden es estable entre páginas.

4. Seguridad y Posibles Errores

HTTP	error_code	Cuándo ocurre y cómo resolverlo
400	VALIDATION_ERROR	page o page_size no son números positivos, o tipo_doc_id no es un entero positivo. Corrige el valor y repite la consulta.

HTTP	error_code	Cuándo ocurre y cómo resolverlo
401	—	No enviaste el encabezado <code>Authorization</code> , o el token no existe o expiró. Vuelve a consumir Login .
403	CONSUMIDOR_NO_AUTENTICADO	El token es válido, pero no corresponde a un cliente OAuth habilitado para consumir la API.
403	RECURSO_NO_AUTORIZADO	Al cliente OAuth le falta el permiso <code>catalogs:read</code> .
500	INTERNAL_ERROR	Fallo interno al consultar. Reintenta; si persiste, contacta a soporte.

Ejemplo de parámetro inválido (400)

```
{
  "success": false,
  "status": 400,
  "message": "El parámetro \"page\" debe ser un número positivo.",
  "error_code": "VALIDATION_ERROR"
}
```

5. Ejemplos de integración

cURL

```
TOKEN="TU_TOKEN_AQUI"
INSTANCIA="tu-inmobiliaria.nuby.app"

# Buscar por documento exacto
curl -G "https://${INSTANCIA}/service/v2/public/third-parties/list" \
  --data-urlencode "documento=900123456" \
  -H "Authorization: Bearer ${TOKEN}"

# Buscar por nombre, página 1 de 10 registros
curl -G "https://${INSTANCIA}/service/v2/public/third-parties/list" \
  --data-urlencode "nombre=acme" \
```

```
--data-urlencode "page=1" \  
--data-urlencode "page_size=10" \  
-H "Authorization: Bearer ${TOKEN}"
```

PHP

```
<?php  
  
$instancia = 'tu-inmobiliaria.nuby.app';  
$token = 'TU_TOKEN_AQUI';  
  
$query = http_build_query([  
    'documento' => '900123456',  
    'page' => 1,  
    'page_size' => 50,  
]);  
  
$url = "https://{ $instancia }/service/v2/public/third-parties/list?{$query}";  
  
$ch = curl_init($url);  
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);  
curl_setopt($ch, CURLOPT_HTTPHEADER, [  
    "Authorization: Bearer {$token}",  
]);  
  
$respuesta = curl_exec($ch);  
$codigo = curl_getinfo($ch, CURLINFO_HTTP_CODE);  
curl_close($ch);  
  
$datos = json_decode($respuesta, true);  
  
echo "HTTP: {$codigo}\n";  
echo "Total: " . ($datos['pagination']['total'] ?? 0) . "\n";  
  
foreach ($datos['body'] as $tercero) {  
    echo $tercero['documento'] . ' - ' . $tercero['nombre_completo'] . "\n";  
}
```

```
}
```

Python

```
import requests

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

headers = {
    "Authorization": f"Bearer {token}",
}

pagina = 1
total_paginas = 1

while pagina <= total_paginas:
    respuesta = requests.get(
        url,
        headers=headers,
        params={"nombre": "acme", "page": pagina, "page_size": 50},
        timeout=30,
    )

    if respuesta.status_code != 200:
        print(respuesta.status_code, respuesta.json())
        break

    datos = respuesta.json()
    total_paginas = datos["pagination"]["total_pages"]

    for tercero in datos["body"]:
        print(tercero["documento"], tercero["nombre_completo"])

    pagina += 1
```

JavaScript (Fetch API)

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

const params = new URLSearchParams({
  nombre: 'acme',
  page: '1',
  page_size: '10'
});

const url = `https://${instancia}/service/v2/public/third-parties/list?${params}`;

const respuesta = await fetch(url, {
  headers: { 'Authorization': `Bearer ${token}` }
});

const data = await respuesta.json();

if (respuesta.ok) {
  console.log(`Página ${data.pagination.page} de ${data.pagination.total_pages}`);
  data.body.forEach(t => console.log(t.documento, t.nombre_completo));
} else {
  console.error(`${respuesta.status} ${data.error_code}: ${data.message}`);
}
```

Power Query M (Excel / Power BI)

Power Query consulta una página por vez con `Web.Contents`. Puedes cambiar `nombre` por `documento` o `tipo_doc_id` en la cadena de la dirección.

```
let
  // 1. Configura tus datos de acceso
  instancia = "tu-inmobiliaria.nuby.app",
  token = "TU_TOKEN_AQUI",
```

```

// 2. Prepara la dirección con el filtro y la página
url = "https://" & instancia & "/service/v2/public/third-
parties/list?nombre=acme&page=1&page_size=1000",

// 3. Envía la petición GET
response = Web.Contents(url, [
    Headers = [
        #"Authorization" = "Bearer " & token
    ]
]),

// 4. Decodifica el envelope
jsonResponse = Json.Document(response),
body = jsonResponse[body],
pagination = jsonResponse[pagination],

// 5. Convierte el listado en tabla
tabla = Table.FromList(body, Splitter.SplitByNothing(), null, null,
ExtraValues.Error),
expandido = Table.ExpandRecordColumn(tabla, "Column1",
    {"tercero_id", "documento", "tipo_doc_id", "tipo_doc", "nombre_completo",
    "email", "telefono1", "direccion1", "municipio_id", "tipo_tercero_id"},
    {"TerceroID", "Documento", "TipoDocID", "TipoDoc", "NombreCompleto",
    "Email", "Telefono1", "Direccion1", "MunicipioID", "TipoTerceroID"}),
tipos = Table.TransformColumnTypes(expandido, {
    {"TerceroID", Int64.Type},
    {"TipoDocID", Int64.Type},
    {"MunicipioID", Int64.Type}
}),
totalPaginas = pagination[total_pages]
in
tipos

```

6. Buenas prácticas y límites conocidos

- **Para buscar un tercero concreto, usa `documento`.** Es coincidencia exacta y devuelve el registro correcto sin ambigüedad.
- **Usa `nombre` solo para búsquedas exploratorias.** Es coincidencia parcial y puede devolver varios terceros.
- **No pidas `page_size=1000` "por si acaso".** Pagina con un tamaño razonable y recorre con `page`.
- **Recorre hasta `total_pages`** y no hasta que una página venga vacía: si el filtro no tiene resultados, `total_pages` es `0`.
- **No uses la lista como detalle.** Para nombres y apellidos separados, ubicación completa, información tributaria y arreglo de tipos, consulta la página [Buscar Tercero por Código](#).
- **Guarda el `tercero_id`** que encuentres: evita volver a buscar por documento en cada operación.

Si el tercero que buscas **no existe** en nuby, puedes crearlo con el servicio [Crear o Resolver Tercero](#).

Revisión #1

Creado el 28 octubre 2026 14:21:50 por Jorge Iván González

Actualizado el 28 octubre 2026 14:23:01 por Jorge Iván González