

Crear o Resolver Tercero

[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 registra un **tercero** en nuby —es decir, una persona natural o jurídica con la que tu inmobiliaria se relaciona: un cliente, un propietario, un proveedor, un arrendatario— y le asigna sus tipos o roles dentro del sistema. Si el documento ya existe en nuby, **no se crea un duplicado**: el servicio devuelve el tercero que ya estaba registrado. Si no existe, lo crea en ese mismo momento.

Es el servicio que necesitas cuando tu plataforma maneja sus propios clientes y quiere que nuby tenga la misma información, sin pedirle al usuario que vuelva a digitar los datos en dos sistemas.

¿Para qué sirve este servicio?

Los casos de uso más frecuentes de este servicio son:

- **Sincronizar los clientes de tu plataforma con nuby**, sin pedirle al usuario que vuelva a digitar los datos en dos sistemas.
- **Crear una factura cuando solo conoces el documento** del tercero, no su identificador interno.
- **Garantizar que un tercero exista antes de usarlo en otro proceso**, sin arriesgarte a duplicarlo.

¿Ya tienes el `tercero_id`? Si tu sistema ya guardó el identificador interno de nuby, puedes seguir usándolo directamente en los demás servicios. Si solo tienes el documento, este es el servicio que lo resuelve.

1. El Endpoint (La dirección web)

Método y dirección

Dato	Valor
Método	POST
Recurso	/third-parties
Cuerpo de la petición	Obligatorio: objeto JSON con los datos del tercero (sección 2)

Apunta tu sistema a la siguiente dirección de petición **POST**:

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

¿Qué debes colocar en {{instancia}}? La dirección web principal con la que ingresas a tu plataforma, incluyendo la terminación `.nuby.app` o `.arrendasoft.co`. Por ejemplo, si entras en `inmobiliaria.nuby.app`, esa es tu instancia. No incluyas `https://` ni barras diagonales al final.

Encabezados requeridos

Encabezado	Valor
Content-Type	application/json
Authorization	Bearer <token>, con el token obtenido en el servicio Login .

Permiso requerido (scope)

El cliente OAuth con el que se genera el token debe tener habilitado el permiso `third-parties:create`. Si el token es válido pero al cliente le falta ese 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**. Allí se explica cómo identificarte y cómo se construye el encabezado `Authorization`.

2. La Petición (¿Qué debes enviarnos?)

El cuerpo debe ser un **objeto JSON**. La raíz del JSON lleva la referencia de idempotencia y, al mismo nivel, los datos del tercero.

Campos obligatorios

Campo	Tipo	Requerido	Descripción
<code>referencia_externa</code>	texto	Sí	Identificador único de tu operación. Es la llave de idempotencia: sirve para que un reintento no cree un segundo tercero. No puede venir vacío.
<code>documento</code>	texto	Sí	Número de documento o NIT del tercero. Es la llave con la que nuby decide si el tercero ya existe. Solo admite letras y números (sin puntos, guiones ni espacios).
<code>tipo_doc_id</code>	entero	Sí	Identificador del tipo de documento en la maestra de nuby (por ejemplo, NIT o cédula). Debe ser un número mayor que cero y existir en el sistema.
<code>nombre1</code>	texto	Sí	Primer nombre, o la razón social completa si es una empresa. nuby lo guarda en mayúsculas.
<code>direccion1</code>	texto	Sí	Dirección principal del tercero.
<code>municipio_id</code>	entero	Sí	Identificador del municipio. Debe ser un número mayor que cero y existir en la maestra de municipios.
<code>tipo_tercero_id</code> o <code>tipos[]</code>	entero / arreglo	Sí (al menos uno)	Rol o roles del tercero. Debes enviar al menos uno de los dos campos (ver el apartado "Cómo se envían los tipos de tercero", más abajo en esta sección).

Importante: los datos obligatorios se exigen cuando el tercero **no existe**. Si el documento ya está registrado, nuby lo resuelve y no valida ni actualiza el resto de los campos.

Campos opcionales

Grupo	Campos
-------	--------

Identidad	nombre2, apellido1, apellido2, persona, sexo, fecha_nacimiento, dv, tratamiento_id
Contacto	direccion2, telefono1, telefono2, fax, email, email_cumpleanos, codigo_postal
Información tributaria	tipo_persona_id, regimen_id, resp_fiscal_id, autoretenedor, autore_res, gran_contribuyente, gran_contr_res, aplica_documento_soporte
Información bancaria	banco_id, intTipoCuentaB, no_cuenta
Representación legal	representante, representante_legal, certificado
Tipos adicionales	tipos[]

Campos opcionales: tipo y descripción

Los campos opcionales se guardan si vienen, pero ninguno reemplaza a un obligatorio. Los identificadores se validan contra las maestras y, si no existen, la respuesta es 422.

Campo	Tipo	Efecto
nombre2	texto	Segundo nombre. Se almacena en mayúsculas.
apellido1	texto	Primer apellido. Se almacena en mayúsculas.
apellido2	texto	Segundo apellido. Se almacena en mayúsculas.
persona	entero	1 persona natural; 2 persona jurídica. Activa la validación de coherencia con el tipo de documento.
sexo	texto	Se acepta al crear, pero no se devuelve en el detalle del tercero.
fecha_nacimiento	fecha	Se acepta al crear, pero no se devuelve en el detalle.
dv	texto	Dígito de verificación del documento. Se devuelve en el detalle.
tratamiento_id	entero	Identificador de la maestra de tratamiento. Se valida contra el catálogo.
direccion2	texto	Dirección secundaria. Se devuelve en el detalle.
telefono1, telefono2	texto	Teléfonos del tercero. Se devuelven en el detalle.
fax	texto	Fax del tercero. Se devuelve en el detalle.

Campo	Tipo	Efecto
email	texto	Correo principal. No se valida su formato; se devuelve en el detalle.
email_cumpleanos	texto	Correo alternativo para cumpleaños. Se devuelve en el detalle.
codigo_postal	texto	Código postal. Se devuelve en el detalle.
tipo_persona_id	entero	Identificador de la maestra de tipo de persona. Se valida contra el catálogo.
regimen_id	entero	Régimen tributario. Se valida contra el catálogo.
resp_fiscal_id	entero	Responsabilidad fiscal. Se valida contra el catálogo.
autoretenedor	indicador	Marca de autorretenedor. Se devuelve en el detalle.
autore_res	indicador	Se acepta al crear; el detalle no lo devuelve.
gran_contribuyente	indicador	Marca de gran contribuyente. Se devuelve en el detalle.
gran_contr_res	indicador	Se acepta al crear; el detalle no lo devuelve.
aplica_documento_soporte	indicador	Marca de documento soporte. Se devuelve en el detalle.
banco_id	entero	Banco del tercero. Se valida contra el catálogo y se devuelve en el detalle.
intTipoCuentaB	entero	Tipo de cuenta bancaria. Se devuelve en el detalle.
no_cuenta	texto	Número de cuenta bancaria. Se devuelve en el detalle.
representante	texto	Se acepta al crear; el detalle no lo devuelve.
representante_legal	texto	Se acepta al crear; el detalle no lo devuelve.
certificado	texto / indicador	Se acepta al crear; el detalle no lo devuelve.
tipos[]	arreglo o cadena	Se fusiona con tipo_tercero_id, se deduplica y se ordena. Si no queda ningún tipo válido, la respuesta es 422.

Sobre `persona`: `1` indica persona natural y `2` persona jurídica. Si lo envías, nuby valida que el tipo de documento sea coherente: una persona natural no puede registrarse con NIT, y una persona jurídica debe registrarse con NIT. Si no lo envías, esa coherencia no se valida.

Campos que se ignoran: si envías campos que no están en las listas anteriores (por ejemplo `tercero_id`, `creador` o `password_hash`), el servicio los descarta en silencio y **nunca** los escribe. No se responde error por ese motivo.

Cómo se envían los tipos de tercero

Puedes enviar el rol del tercero de dos formas, y también combinarlas:

- `tipo_tercero_id` → un solo identificador, como número o como texto.
- `tipos[]` → un arreglo con varios identificadores.

Si envías ambos, nuby los **une en una sola lista**, elimina los repetidos y conserva el orden. Los valores deben ser números enteros mayores que cero; los valores inválidos se descartan y, si al final no queda ningún tipo, la respuesta es `422`.

Ejemplo: persona natural

Para una persona natural envía `persona: 1` y un tipo de documento que **no** sea NIT. El identificador `2` del ejemplo es ilustrativo: usa el que corresponda a cédula en la maestra de tu instancia.

```
{
  "referencia_externa": "TP-000124",
  "documento": "1020304050",
  "tipo_doc_id": 2,
  "persona": 1,
  "nombre1": "María",
  "apellido1": "Gómez",
  "direccion1": "CL 10 #5-20",
  "municipio_id": 1,
  "tipo_tercero_id": 2,
  "email": "maria.gomez@example.co",
  "telefono1": "3001234567"
}
```

nuby almacena los nombres y apellidos en mayúsculas: este envío queda como `MARÍA GÓMEZ`.

Ejemplo: persona jurídica

Para una persona jurídica envía `persona: 2` y un documento de tipo NIT. Si además necesitas asignarle varios roles, combina `tipo_tercero_id` con `tipos[]`.

```
{
  "referencia_externa": "TP-000125",
  "documento": "901234567",
  "tipo_doc_id": 1,
  "persona": 2,
  "nombre1": "Inversiones Andinas SAS",
  "direccion1": "CR 43A #1-50",
  "municipio_id": 1,
  "tipo_tercero_id": 1,
  "tipos": [2],
  "email": "contacto@andinas.co",
  "telefono1": "6044445555"
}
```

El ejemplo usa `tipo_doc_id: 1` para NIT, igual que el contrato. La coherencia con `persona: 2` se valida antes de crear el tercero.

Ejemplo completo de la petición

```
{
  "referencia_externa": "TP-000123",
  "documento": "900123456",
  "tipo_doc_id": 1,
  "nombre1": "Acme SAS",
  "tipo_tercero_id": 1,
  "tipos": [2],
  "direccion1": "CL 1 #2-3",
  "municipio_id": 1,
  "email": "cuenta@acme.co",
  "telefono1": "6040000000"
}
```

3. La Respuesta (¿Qué te entregaremos?)

El servicio tiene **dos resultados posibles**, y ambos son exitosos:

Resultado	Código HTTP	Qué significa
Tercero creado	201	El documento no existía en nuby. Se creó el tercero y se le asignaron sus tipos.
Tercero resuelto	200	El documento ya existía. Se devuelve el mismo <code>tercero_id</code> y no se modifica nada .

Tercero creado (201 Created)

```
{
  "success": true,
  "status": 201,
  "message": "Tercero creado.",
  "body": {
    "tercero_id": 123,
    "created": true,
    "documento": "900123456"
  }
}
```

Tercero existente (200 OK)

```
{
  "success": true,
  "status": 200,
  "message": "Tercero existente resuelto.",
  "body": {
    "tercero_id": 123,
    "created": false,
    "documento": "900123456"
  }
}
```

Reintento de una solicitud ya completada (replay)

Si repites la petición con la **misma** `referencia_externa` y el mismo contenido, nuby devuelve la respuesta original —con el mismo código HTTP— y le agrega la marca

```
idempotent_replay: true:
```

```
{
  "success": true,
  "status": 201,
  "message": "Tercero creado.",
  "body": {
    "tercero_id": 123,
    "created": true,
    "documento": "900123456"
  },
  "idempotent_replay": true
}
```

Estructura del envelope de respuesta

Campo	Tipo	Descripción
<code>success</code>	booleano	<code>true</code> cuando la operación se completó; <code>false</code> en un error.
<code>status</code>	entero	Replica el código HTTP de la respuesta: <code>201</code> si el tercero se creó y <code>200</code> si ya existía.
<code>message</code>	texto	Mensaje legible de la operación: <code>Tercero creado.</code> o <code>Tercero existente resuelto.</code>
<code>body</code>	objeto	Resultado de la operación: <code>tercero_id</code> , <code>created</code> y <code>documento</code> .
<code>idempotent_replay</code>	booleano	Solo aparece cuando la respuesta se recuperó de una solicitud anterior.

Una escritura no devuelve `pagination` ni `filters_applied`: la respuesta es el resultado de la operación.

Qué contiene el cuerpo de la respuesta

Campo	Tipo	Descripción
<code>tercero_id</code>	entero	Identificador interno del tercero en nuby. Guárdalo: es la llave para consultarlo después.
<code>created</code>	booleano	<code>true</code> si el tercero se creó en esta petición; <code>false</code> si ya existía.

Campo	Tipo	Descripción
<code>documento</code>	texto	El documento que enviaste, ya normalizado (sin espacios).
<code>idempotent_replay</code>	booleano	Solo aparece en un replay. Indica que la respuesta se recuperó de una solicitud anterior y que no se ejecutó nada nuevo.

La respuesta de este servicio es corta a propósito. Si necesitas el detalle completo del tercero (nombres, apellidos, ubicación, tipos, información tributaria), consulta la página [Buscar Tercero por Código](#).

4. Cómo decide nuby si crea o resuelve

El servicio sigue siempre el mismo orden. Conocerlo te ayuda a interpretar las respuestas:

1. Verifica que venga `referencia_externa`. Si falta, responde `422` y no registra nada.
2. Consulta si tu `referencia_externa` ya fue usada (idempotencia).
3. Busca un tercero con ese `documento`.
4. **Si ya existe** → responde `200` con `created: false` y el mismo `tercero_id`. No valida ni modifica el resto de los datos.
5. **Si no existe** → valida los obligatorios y las reglas de las maestras (tipo de documento, municipio, régimen, responsabilidad fiscal).
6. Si todo es válido, crea el tercero, guarda sus tipos y responde `201` con `created: true`.

Ojo con las mayúsculas: los nombres y apellidos (`nombre1`, `nombre2`, `apellido1`, `apellido2`) se almacenan en mayúsculas. Si envías "Acme SAS", en nuby quedará "ACME SAS".

Ojo con el documento: solo se aceptan letras y números. Un documento con puntos, guiones o espacios (por ejemplo `900.123.456`) no pasa la validación. Envíalo normalizado: `900123456`.

Un tercero, un registro. El documento es único en nuby. Por eso dos envíos con el mismo documento —aunque traigan otro nombre— apuntan siempre al mismo tercero.

5. Idempotencia y reintentos

Toda escritura en esta API exige `referencia_externa`. nuby guarda la combinación **(consumidor, recurso, referencia_externa)** y recuerda qué respondió, para que un reintento no duplique el registro.

Situación	Qué responde nuby
Primera vez que se usa la referencia	Procesa la solicitud normal (<code>201</code> o <code>200</code>).
La referencia ya se completó con el mismo contenido	La respuesta original, con <code>idempotent_replay: true</code> .
La referencia ya se completó con otro contenido	<code>409</code> con <code>IDEMPOTENCY_PAYLOAD_MISMATCH</code> .
La referencia está siendo procesada en este momento (envío simultáneo)	<code>409</code> con <code>IDEMPOTENCY_IN_PROGRESS</code> .
La referencia quedó en una solicitud fallida	Se permite reintentar, incluso con datos corregidos.

Importante: una respuesta `422` por datos inválidos **no consume** la referencia. Puedes corregir el contenido y reintentar con la misma `referencia_externa` sin conflicto.

Atención: no reutilices una `referencia_externa` para operaciones distintas. Si usas la misma referencia con un contenido diferente, nuby responde `409` `IDEMPOTENCY_PAYLOAD_MISMATCH` y no procesa nada. La referencia debe identificar **una operación concreta**, no un tercero.

Recomendación: guarda en tu plataforma la relación `referencia_externa → tercero_id`. Así puedes reintentar con seguridad ante timeouts o caídas de red, y sabes a qué tercero corresponde cada operación.

6. Seguridad y Posibles Errores

Las respuestas de error traen siempre `success: false`, el código HTTP, un `message` legible y un `error_code` estable que puedes usar en tu lógica.

HTTP	<code>error_code</code>	Cuándo ocurre y cómo resolverlo
<code>400</code>	<code>VALIDATION_ERROR</code>	El cuerpo no es un objeto JSON. Envía un objeto JSON válido y define <code>Content-Type: application/json</code> .

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 y repite la petición.
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>third-parties:create</code> . Solicita el ajuste a tu administrador en nuby.
409	IDEMPOTENCY_IN_PROGRESS	Hay una solicitud en curso con esa referencia. Espera unos segundos y reintenta con la misma referencia.
409	IDEMPOTENCY_PAYLOAD_MISMATCH	La referencia ya se usó con otro contenido. Usa una <code>referencia_externa</code> nueva.
422	VALIDATION_ERROR	Los datos no cumplen las reglas. Revisa el objeto <code>errors</code> para saber qué campo falló.
500	INTERNAL_ERROR / IDEMPOTENCY_STORAGE_ERROR	Fallo interno de nuby. La solicitud queda marcada como fallida y puede reintentarse. Si persiste, contacta a soporte.

Ejemplo de error de validación (422)

```
{
  "success": false,
  "status": 422,
  "message": "La solicitud contiene errores de validación.",
  "error_code": "VALIDATION_ERROR",
  "errors": {
    "direccion1": [
      "El campo \"Dirección Principal\" es obligatorio."
    ]
  }
}
```

El objeto `errors` está indexado por el **nombre técnico del campo**, y cada entrada es una lista de mensajes. Las validaciones más frecuentes son:

Validaciones y mensajes

Regla	Mensaje	Cómo corregir
Falta un campo obligatorio y el tercero no existe	El campo "<Etiqueta>" es obligatorio.	Incluye el campo. Recuerda que los obligatorios solo se exigen cuando el documento no existe.
tipo_doc_id o municipio_id vacíos, en cero o no numéricos	El campo "<Etiqueta>" debe ser un identificador numérico mayor que cero.	Envía un entero mayor que cero tomado de la maestra correspondiente.
No se envió ningún tipo de tercero	Debe enviar al menos un tipo de tercero en "Tipo de Tercero" o en tipos[].	Envía tipo_tercero_id o tipos[] con al menos un entero mayor que cero.
Un identificador no existe en las maestras	El valor ingresado en el campo "<Etiqueta>" no es válido para el sistema.	Verifica el catálogo de tipo de documento, municipio, régimen, responsabilidad fiscal o banco.
Persona natural con NIT	Una Persona Natural no puede registrarse con un tipo de documento de tipo NIT.	Cambia el tipo_doc_id a uno que no sea NIT o corrige persona.
Persona jurídica sin NIT	Una Persona Jurídica debe registrarse de forma obligatoria con un documento de tipo NIT.	Usa un tipo_doc_id de tipo NIT o corrige persona.

Atención: cuando la respuesta es 422, nuby **no crea ni modifica** ningún tercero. Es seguro corregir el contenido y volver a enviar.

Casos especiales

Estos escenarios explican la mayoría de las dudas de integración. La regla que manda siempre es la misma: primero se resuelve el documento y, si ya existe, el resto del payload no se valida.

Tercero existente con payload incompleto

Escenario	Resultado	Qué hace nuby
El documento ya existe y solo envías referencia_externa y documento	200 con created: false	Resuelve el tercero y no exige los demás obligatorios.
El documento ya existe y omite direccion1 o municipio_id	200 con created: false	No valida los obligatorios ni actualiza el registro.
El documento ya existe y envías un correo con formato inválido	200 con created: false	Ignora el correo y el resto del envío.
El documento no existe y falta direccion1	422 VALIDATION_ERROR	Exige los obligatorios antes de crear.

Persona natural con NIT

Combinación enviada	Resultado	Cómo corregir
<code>persona: 1</code> con un <code>tipo_doc_id</code> de tipo NIT	<code>422</code> : <i>Una Persona Natural no puede registrarse con un tipo de documento de tipo NIT.</i>	Usa un tipo de documento que no sea NIT, o cambia <code>persona</code> a <code>2</code> .
<code>persona: 2</code> con un <code>tipo_doc_id</code> que no es NIT	<code>422</code> : <i>Una Persona Jurídica debe registrarse de forma obligatoria con un documento de tipo NIT.</i>	Usa un tipo de documento NIT, o cambia <code>persona</code> a <code>1</code> .
Omites <code>persona</code> y envías un NIT	Se crea si el resto es válido	No hay validación de coherencia documento↔persona; envíala siempre que puedas.

Documento con puntos o guiones

Documento enviado	Resultado	Cómo corregir
<code>900.123.456</code>	<code>422 VALIDATION_ERROR</code>	Envía <code>900123456</code> , sin puntos.
<code>900-123-456</code>	<code>422 VALIDATION_ERROR</code>	Envía <code>900123456</code> , sin guiones.
<code>900123456</code>	<code>422 VALIDATION_ERROR</code>	Quita los espacios antes y después del documento.
<code>900123456</code>	Aceptado	Solo letras y números; no requiere transformación adicional.

Campos ignorados

Campo enviado	Efecto	¿Genera error?
<code>tercero_id</code>	Se descarta; nuby asigna el identificador real.	No
<code>creadopor</code>	Se descarta; nuby registra el consumidor autenticado.	No
<code>password_hash</code>	Se descarta y nunca se escribe.	No
<code>impuestos</code>	Se descarta; no forma parte de la whitelist.	No
Cualquier otro campo fuera de la whitelist	Se ignora en silencio.	No

`tipos[]` + `tipo_tercero_id`

Entrada	Tipos resultantes	Explicación
<code>tipo_tercero_id: 1</code> , <code>tipos: [2]</code>	<code>[1, 2]</code>	Se fusionan ambos campos y se conserva el orden.

Entrada	Tipos resultantes	Explicación
tipo_tercero_id: "1,2", tipos: [2,3]	[1, 2, 3]	Se aceptan cadenas separadas por comas y se eliminan los duplicados.
tipo_tercero_id: 0, tipos: [1]	[1]	Los valores <code><= 0</code> o no numéricos se descartan.
tipos: ["abc"] sin tipo_tercero_id	Ninguno	422: no queda ningún tipo válido.
Omites ambos campos	Ninguno	422: debes enviar al menos un tipo de tercero.

7. Ejemplos de integración

cURL

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

curl -X POST "https://${INSTANCIA}/service/v2/public/third-parties" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer ${TOKEN}" \
-d '{
  "referencia_externa": "TP-000123",
  "documento": "900123456",
  "tipo_doc_id": 1,
  "nombre1": "ACME SAS",
  "tipo_tercero_id": 1,
  "tipos": [2],
  "direccion1": "CL 1 #2-3",
  "municipio_id": 1,
  "email": "cuenta@acme.co",
  "telefono1": "6040000000"
}'
```

PHP

```
<?php

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

$payload = [
    'referencia_externa' => 'TP-000123',
    'documento' => '900123456',
    'tipo_doc_id' => 1,
    'nombre1' => 'ACME SAS',
    'tipo_tercero_id' => 1,
    'tipos' => [2],
    'direccion1' => 'CL 1 #2-3',
    'municipio_id' => 1,
    'email' => 'cuenta@acme.co',
    'telefono1' => '6040000000',
];

$ch = curl_init($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}",
]);

$respuesta = curl_exec($ch);
$codigo = curl_getinfo($ch, CURLINFO_HTTP_CODE);
curl_close($ch);

echo "HTTP: {$codigo}\n";
print_r(json_decode($respuesta, true));
```

Python

```
import requests

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

payload = {
    "referencia_externa": "TP-000123",
    "documento": "900123456",
    "tipo_doc_id": 1,
    "nombre1": "ACME SAS",
    "tipo_tercero_id": 1,
    "tipos": [2],
    "direccion1": "CL 1 #2-3",
    "municipio_id": 1,
    "email": "cuenta@acme.co",
    "telefono1": "6040000000",
}

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

respuesta = requests.post(url, json=payload, headers=headers, timeout=30)

if respuesta.ok:
    datos = respuesta.json()
    print(f"Tercero {datos['body']['tercero_id']} - creado: {datos['body']['created']}")
else:
    print(respuesta.status_code, respuesta.json())
```

JavaScript (Fetch API)

```

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

const payload = {
  referencia_externa: 'TP-000123',
  documento: '900123456',
  tipo_doc_id: 1,
  nombre1: 'ACME SAS',
  tipo_tercero_id: 1,
  tipos: [2],
  direccion1: 'CL 1 #2-3',
  municipio_id: 1,
  email: 'cuenta@acme.co',
  telefonol: '6040000000'
};

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

const data = await respuesta.json();

if (respuesta.ok) {
  console.log(`Tercero ${data.body.tercero_id} – creado: ${data.body.created}`);
} else {
  console.error(`${respuesta.status} ${data.error_code}: ${data.message}`, data.errors ?? {});
}

```

Power Query M (Excel / Power BI)

Power Query construye el cuerpo con `Json.FromValue` y lo envía con `Web.Contents`. El resultado se convierte en una tabla de una fila con `tercero_id`, `created` y `documento`.

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

    // 2. Prepara la dirección de la petición
    url = "https://" & instancia & "/service/v2/public/third-parties",

    // 3. Construye el cuerpo JSON
    cuerpo = Json.FromValue([
        referencia_externa = "TP-000123",
        documento = "900123456",
        tipo_doc_id = 1,
        nombre1 = "ACME SAS",
        tipo_tercero_id = 1,
        tipos = {2},
        direccion1 = "CL 1 #2-3",
        municipio_id = 1,
        email = "cuenta@acme.co",
        telefono1 = "6040000000"
    ]),

    // 4. Envía la petición POST
    response = Web.Contents(url, [
        Headers = [
            #"Content-Type" = "application/json",
            #"Authorization" = "Bearer " & token
        ],
        Content = cuerpo
    ]),

    // 5. Decodifica la respuesta JSON
    jsonResponse = Json.Document(response),
    body = jsonResponse[body],
```

```
// 6. Convierte el resultado en tabla
tabla = Table.FromList({body}, Splitter.SplitByNothing(), null, null,
ExtraValues.Error),
    expandido = Table.ExpandRecordColumn(tabla, "Column1",
        {"tercero_id", "created", "documento"},
        {"TerceroID", "Creado", "Documento"}),
    tipos = Table.TransformColumnTypes(expandido, {
        {"TerceroID", Int64.Type},
        {"Creado", type logical}
    })
in
    tipos
```

8. Buenas prácticas y límites conocidos

Recomendaciones

- **Normaliza el documento antes de enviarlo:** sin puntos, sin guiones y sin espacios.
- **Usa una referencia externa por operación** y no la reutilices para otro tercero.
- **Aprovecha el resultado** `created` para saber si debes registrar el alta en tu plataforma o si el tercero ya estaba en nuby.
- **Envía el** `tercero_id` **que devuelve nuby** en los procesos siguientes: evita volver a resolver por documento.
- **Trata el** `422` **como reintentable tras corregir:** la referencia no se consume.
- **Guarda el token fuera del código** y renuévalo antes de que expire.

Límites de este servicio

- No existe un servicio de **actualización** de terceros en esta API: si necesitas corregir los datos de un tercero existente, se gestiona desde nuby.
- Un tercero existente se devuelve **sin validar ni comparar** el resto del envío; no esperes que el servicio detecte diferencias.
- El servicio **no valida el formato del correo electrónico**. Envía direcciones válidas.
- Los terceros creados por este servicio quedan disponibles para los demás procesos de nuby en el mismo momento de la respuesta.

Lo que este servicio NO hace

- **No actualiza** los datos de un tercero que ya existe. Si el tercero está registrado con otro nombre o dirección, el servicio lo resuelve tal como está y descarta el resto del envío.
- **No completa datos por su cuenta.** No rellena direcciones, correos ni teléfonos con valores por defecto: si falta un dato obligatorio y el tercero no existe, responde un error de validación.
- **No gestiona** contactos, archivos, bitácoras ni documentos del tercero: eso se administra en el módulo de terceros de nuby.

Para consultar terceros ya registrados, revisa las páginas [*Listar Terceros*](#) y [*Buscar Tercero por Código*](#).

Revisión #1

Creado el 28 octubre 2026 14:24:41 por Jorge Iván González

Actualizado el 28 octubre 2026 14:25:55 por Jorge Iván González