

Pasarelas de Pago

Aquí encontrarás la documentación especializada y exclusiva para integrar soluciones de recaudo y procesamiento de pagos con nuestra plataforma.

Si la API general es nuestro menú principal, considera esta sección como la carta dedicada enteramente a gestionar las transacciones financieras de manera fluida y segura.

⚠ **Un requisito indispensable antes de empezar:** Dado que estamos manejando información financiera, la seguridad es nuestra prioridad. Por eso, para poder consumir los *endpoints* de esta categoría, **es obligatorio que operes como una pasarela de pago registrada oficialmente en nuby y que cuentes con un cliente OAuth válido.** Piensa en este cliente OAuth como tu "llave de acceso de alta seguridad"; es lo que nos certifica quién eres y nos garantiza que la comunicación entre tu pasarela y nuestra plataforma está 100% protegida y autorizada.

Si ya tienes tus credenciales de acceso listas, ¡explora esta sección y haz que tu pasarela de pagos se integre a la perfección con el ecosistema de nuby!

- [Endpoints](#)
 - [Registrar Pago](#)
 - [Obtener Cartera](#)

Endpoints

Aquí te detallamos los *endpoints* (puntos de conexión específicos) diseñados para interactuar con la información de cobros de la inmobiliaria.

En este apartado te explicaremos paso a paso cómo usar servicios clave, como por ejemplo:

- **Obtener cartera:** Cómo consultarle a nuestro sistema cuáles son las deudas, facturas o saldos pendientes de los clientes.
- **Registrar pagos:** Cómo notificarle a nuby automáticamente en el instante en que un pago se ha procesado con éxito en tu plataforma.

Registrar Pago

Permite a una pasarela de pagos autenticada (por ejemplo, Palomma) **registrar el pago** de uno o varios ítems pagables en la inmobiliaria: una factura, un movimiento (concepto / interés) o una factura junto con sus intereses adjuntos. El endpoint es idempotente, devuelve un cuerpo en formato de arreglo y aplica automáticamente la configuración de la pasarela (forma de pago, envío DIAN).

¿Para qué sirve este servicio?

Está pensado exclusivamente para integraciones con pasarelas de pago. La pasarela lo invoca cuando el deudor confirma el pago en línea de los ítems devueltos previamente por **GET /gateways/portfolio**. El sistema persiste el pago, genera el recibo de caja y, cuando la configuración lo requiera, dispara el envío del documento electrónico a la DIAN.

Endpoint restringido a pasarelas registradas

Este endpoint **solo** puede ser consumido por clientes OAuth2 cuyo `client_name` esté registrado como una pasarela activa en la inmobiliaria. Cualquier otro cliente recibirá `403 Forbidden`.

1. El Endpoint (La dirección web)

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

```
POST https://{{instancia}}/service/v2/public/gateways/payments
```

¿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?)

Debes enviar una petición **POST** con el detalle del pago en el cuerpo de la solicitud en formato JSON. La pasarela se identifica automáticamente a partir del `client_name` del token; **no debes enviar el nombre de la pasarela ni la forma de pago en el body**: ambos se derivan de la configuración registrada.

Método	POST
Content-Type	application/json
Authorization	Bearer token , Token obtenido al consumir el servicio Login con un cliente OAuth2 registrado como pasarela activa.

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**. El cliente OAuth además debe estar registrado como pasarela activa; en caso contrario el endpoint responderá `403`.

Cuerpo de la petición (Body):

Campo	Tipo	Requerido	Descripción
<code>factura_id</code>	integer	Condicional	Identificador de la factura a pagar. Debe existir en el sistema. Es obligatorio si no se envía <code>movimiento_id</code> .

Campo	Tipo	Requerido	Descripción
movimiento_id	integer array	Condicional	Identificador (o lista de identificadores) de los movimientos a pagar. Es obligatorio si no se envía <code>factura_id</code> . Cuando se envía sin <code>factura_id</code> , por defecto solo se admite un único movimiento; sin embargo, si la pasarela tiene activo el filtro « Agrupar conceptos del mismo periodo », se aceptan varios movimientos del mismo tercero y se procesan en modo GROUP (un solo recibo de caja + cuando aplique una factura agrupada por contrato).
monto	number	Sí	Valor pagado por el deudor. Debe ser un número mayor a cero. Si excede el saldo adeudado, el excedente se registra como anticipo .
fecha_pago	string	Sí	Fecha y hora del pago. Formato estricto: <code>YYYY-MM-DD HH:MM:SS</code> .
n_comprobante	string	No	Número de comprobante de la pasarela. Máximo 100 caracteres. Solo letras, números, guiones, guiones bajos, puntos y espacios. Recomendado: garantiza la idempotencia ante reintentos.

Modos de operación

El endpoint clasifica el pago en uno de cuatro modos según los campos enviados y la configuración de la pasarela:

- **Modo movimiento único:** `factura_id = null` y `movimiento_id = [n]`.
- **Modo GROUP:** `factura_id = null` y `movimiento_id = [m1, m2, ...]` con varios movimientos del mismo tercero. Solo se admite cuando la pasarela tiene activo el filtro avanzado «**Agrupar conceptos del mismo periodo**». Genera un único recibo de caja y, cuando aplique, una factura agrupada por contrato.
- **Modo factura:** `factura_id = X` y `movimiento_id = []` o ausente.
- **Modo factura + intereses adjuntos:** `factura_id = X` y `movimiento_id = [m1, m2, ...]`. Los movimientos que pertenezcan a la factura X se ignoran (ya están

cubiertos); los demás se procesan como intereses sueltos en la misma transacción.

Ejemplo 1: Pago de una sola factura

```
{
  "factura_id": 4521,
  "monto": 1750000.00,
  "fecha_pago": "2026-04-25 14:30:00",
  "n_comprobante": "PALM-2026-04-25-998877"
}
```

Ejemplo 2: Pago de un movimiento (concepto suelto)

```
{
  "movimiento_id": 98515,
  "monto": 280000.00,
  "fecha_pago": "2026-04-25 14:30:00",
  "n_comprobante": "PALM-2026-04-25-998878"
}
```

Ejemplo 3: Pago de una factura junto con sus intereses adjuntos

```
{
  "factura_id": 4521,
  "movimiento_id": [98410, 98411],
  "monto": 1820000.00,
  "fecha_pago": "2026-04-25 14:30:00",
  "n_comprobante": "PALM-2026-04-25-998879"
}
```

3. La Respuesta (¿Qué te entregaremos?)

El sistema responde con HTTP `200` cuando el pago se registra correctamente. La clave `body` es **siempre un arreglo**, incluso cuando el pago resulta en un solo registro. Cada elemento del arreglo describe un documento generado por la operación: la factura pagada, una factura nueva creada por intereses facturables, o un anticipo si hubo excedente.

```
{
  "success": true,
  "status": 200,
  "message": "El pago fue registrado exitosamente.",
  "body": [
    {
      "factura_id": 4521,
      "movimiento_id": null,
      "pago_id": 87123,
      "recibo_id": 56412,
      "documento_contable_id": 102345,
      "monto_pagado": 1750000.00,
      "fecha_pago": "2026-04-25 14:30:00",
      "forma_pago_id": 7,
      "forma_pago": "Pasarela Palomma",
      "n_comprobante": "PALM-2026-04-25-998877",
      "saldo_anterior": 1750000.00,
      "saldo_actual": 0.00,
      "estado": "pagada",
      "mensaje": "El pago cubrió el total de la factura.",
      "estado_dian": "pendiente",
      "gateway": "palomma",
      "client_name": "palomma_inmobiliaria_xyz"
    }
  ],
  "alertas": [],
  "data": {
    "factura_id": 4521,
    "pago_id": 87123,
    "recibo_id": 56412,
    "documento_contable_id": 102345,
    "confirm_pay_id": 9921,
    "gateway": "palomma",
    "client_name": "palomma_inmobiliaria_xyz",
    "documentos_generados": [
      { "tipo": "recibo", "id": 56412, "numero": "REC-56412" },
      { "tipo": "factura", "id": 4521, "numero": "FAC-4521" }
    ]
  }
}
```

Ejemplo de respuesta con excedente (anticipo):

```
{
  "success": true,
  "status": 200,
  "message": "El pago fue registrado exitosamente.",
  "body": [
    {
      "factura_id": 4521,
      "movimiento_id": null,
      "pago_id": 87124,
      "recibo_id": 56413,
      "documento_contable_id": 102346,
      "monto_pagado": 1750000.00,
      "fecha_pago": "2026-04-25 14:30:00",
      "forma_pago_id": 7,
      "forma_pago": "Pasarela Palomma",
      "n_comprobante": "PALM-2026-04-25-998880",
      "saldo_anterior": 1750000.00,
      "saldo_actual": 0.00,
      "estado": "pagada",
      "mensaje": "El pago cubrió el total de la factura.",
      "estado_dian": "pendiente",
      "gateway": "palomma",
      "client_name": "palomma_inmobiliaria_xyz"
    },
    {
      "factura_id": null,
      "movimiento_id": null,
      "pago_id": null,
      "recibo_id": null,
      "documento_contable_id": 102347,
      "monto_pagado": 50000.00,
      "fecha_pago": "2026-04-25 14:30:00",
      "forma_pago_id": 7,
      "forma_pago": "Pasarela Palomma",
      "n_comprobante": "PALM-2026-04-25-998880",
      "saldo_anterior": null,
      "saldo_actual": null,
      "estado": "anticipo",
    }
  ]
}
```

```

    "mensaje": "Se generó un anticipo por valor de 50000.00 con el excedente del
pago.",
    "estado_dian": null,
    "gateway": "palomma",
    "client_name": "palomma_inmobiliaria_xyz"
  }
],
"alertas": [],
"data": {
  "factura_id": 4521,
  "pago_id": 87124,
  "recibo_id": 56413,
  "documento_contable_id": 102346,
  "anticipo_documento_contable_ids": [102347],
  "confirm_pay_id": 9922,
  "gateway": "palomma",
  "client_name": "palomma_inmobiliaria_xyz"
}
}

```

Campos de la respuesta

Clave	Descripción
success	Indica si la operación fue exitosa (<code>true</code>) o no (<code>false</code>).
status	Código HTTP devuelto.
message	Mensaje descriptivo del resultado.
body	Arreglo con uno o más registros generados por la operación (ver tabla detallada abajo).
alertas	Arreglo de alertas no bloqueantes generadas durante el proceso (por ejemplo, fallas en el envío DIAN).
data	Objeto con la información consolidada del pago (IDs principales, identificador de confirmación, gateway, cliente). Adicionalmente puede incluir <code>documentos_generados</code> : un arreglo de objetos <code>{tipo, id, numero}</code> que lista todos los documentos creados u afectados por el pago (recibo de caja y factura(s)). El campo <code>tipo</code> puede ser <code>"recibo"</code> o <code>"factura"</code> ; <code>numero</code> es el consecutivo legible para el usuario final.
error_code	Solo presente en respuestas de error. Códigos posibles: <code>"DUPLICATE_PAYMENT"</code> , <code>"VALIDATION_ERROR"</code> , <code>"GATEWAY_NOT_FOUND"</code> , <code>"GATEWAY_NOT_ACTIVE"</code> , <code>"GATEWAY_PAYMENT_METHOD_REQUIRED"</code> , <code>"INTERNAL_ERROR"</code> .

Clave	Descripción
duplicate	Solo presente en respuestas <code>409 Conflict</code> . Contiene el snapshot del pago previamente registrado.

Estructura de cada elemento de `body[]`

Clave	Descripción
factura_id	Identificador de la factura pagada o generada. <code>null</code> para registros de anticipo.
movimiento_id	Identificador (o arreglo de IDs) de los movimientos asociados al registro. Puede ser <code>null</code> .
pago_id	Identificador del pago creado en el sistema. <code>null</code> para anticipos.
recibo_id	Identificador del recibo de caja generado.
documento_contable_id	Identificador del documento contable asociado (recibo, anticipo, etc.).
monto_pagado	Valor efectivamente aplicado a la factura/movimiento. <code>null</code> para anticipos sin asignación específica.
fecha_pago	Fecha y hora del pago. Formato: <code>YYYY-MM-DD HH:MM:SS</code> .
forma_pago_id	Identificador interno de la forma de pago contable (derivada de la configuración de la pasarela).
forma_pago	Nombre legible de la forma de pago.
n_comprobante	Número de comprobante enviado por la pasarela.
saldo_anterior	Saldo de la factura/movimiento antes de aplicar el pago.
saldo_actual	Saldo restante después de aplicar el pago.
estado	Estado del registro tras la operación. Valores posibles: <ul style="list-style-type: none"> <code>"pagada"</code>: el monto cubrió el saldo total de la factura (<code>saldo_actual = 0</code>). <code>"pendiente"</code>: el monto fue inferior al saldo y la factura sigue con saldo pendiente. El campo <code>mensaje</code> describe el saldo restante. <code>"anticipo"</code>: la fila representa un excedente registrado como anticipo (sin factura asociada).
mensaje	Mensaje legible que describe el resultado del pago a nivel de cada fila. Valores típicos: <ul style="list-style-type: none"> <code>"El pago cubrió el total de la factura."</code> cuando <code>estado = "pagada"</code>. <code>"El pago fue registrado parcialmente. La factura aún tiene un saldo pendiente de {monto}."</code> cuando <code>estado = "pendiente"</code>. <code>"Se generó un anticipo por valor de {monto} con el excedente del pago."</code> cuando <code>estado = "anticipo"</code>.

Clave	Descripción
<code>estado_dian</code>	Estado del envío DIAN. Valores posibles: <code>"pendiente"</code> (envío en cola o fallido) o <code>null</code> (no aplica). El detalle del fallo se reporta en <code>alertas</code> .
<code>gateway</code>	Slug de la pasarela autenticada (por ejemplo, <code>"palomma"</code>).
<code>client_name</code>	Nombre del cliente OAuth2 autenticado.

Escenarios especiales del cuerpo

Escenario	Comportamiento del <code>body</code>
Pago simple	Un único elemento que describe el recibo generado.
Intereses facturables	Al menos dos elementos: uno por la factura original pagada y uno por la factura nueva creada para los intereses facturables.
Excedente / anticipo	Cuando el monto pagado supera el saldo de los ítems enviados, el sistema genera un anticipo con el sobrante. La respuesta incluirá una fila adicional con <code>estado = "anticipo"</code> , <code>documento_contable_id</code> poblado, <code>monto_pagado</code> con el valor del anticipo, y los campos <code>fecha_pago</code> , <code>forma_pago_id</code> , <code>forma_pago</code> y <code>n_comprobante</code> heredados del pago original. Los campos <code>factura_id</code> , <code>movimiento_id</code> , <code>pago_id</code> , <code>recibo_id</code> , <code>saldo_anterior</code> y <code>saldo_actual</code> serán <code>null</code> . Importante: el sistema garantiza que primero se cubre el saldo de cada ítem enviado (factura y/o movimientos) y solo el sobrante real se transforma en anticipo; nunca se desvía a anticipo el monto destinado a un movimiento explícitamente declarado.
Falla en envío DIAN	El pago queda persistido (<code>estado = "pagada"</code>), <code>estado_dian = "pendiente"</code> , y <code>alertas</code> contiene una entrada <code>{ "level": "warning", "code": "DIAN_SEND_FAILED", "message": "...", "factura_id": ... }</code> .

4. Idempotencia

El endpoint detecta automáticamente reintentos de un mismo pago y responde `409 Conflict` sin volver a registrarlo. Las reglas de idempotencia son:

Disparador	Comportamiento
Mismo <code>n_comprobante</code>	Cuando el <code>n_comprobante</code> ya fue registrado para la misma pasarela, se devuelve <code>409</code> con el registro previo.
Sin <code>n_comprobante</code> , mismos datos esenciales	Si los datos (<code>factura_id</code> , <code>movimiento_ids</code> ordenados, <code>monto</code> , <code>fecha_pago</code> , <code>client_name</code>) coinciden con un pago previo, se devuelve <code>409</code> .
<code>n_comprobante</code> distinto, mismos datos esenciales	Se aplica un hash de respaldo sobre los datos esenciales y se devuelve <code>409</code> si coincide con un pago previo.

Cómo interpretar el 409

Una respuesta `409 Conflict` de este endpoint **no es un fallo**: significa que el pago **ya existe** en nuby. La pasarela debe tratarla como confirmación idempotente y **no reintentar**. El cuerpo incluye `error_code = "DUPLICATE_PAYMENT"`, `is_business_error = true` y, en `data`, los identificadores del pago original (`confirm_pay_id`, `confirm_pay_reference_code`, `factura_id`, `pago_id`, `recibo_id`, `documento_contable_id`).

Ejemplo de respuesta 409:

```
{
  "success": false,
  "status": 409,
  "message": "El pago ya fue registrado previamente para la pasarela
palomma_inmobiliaria_xyz.",
  "body": [ /* snapshot del body original */ ],
  "error_code": "DUPLICATE_PAYMENT",
  "data": {
    "factura_id": 4521,
    "pago_id": 87123,
    "recibo_id": 56412,
    "documento_contable_id": 102345,
    "confirm_pay_id": 9921,
    "confirm_pay_reference_code": "PALM-2026-04-25-998877",
    "gateway": "palomma",
    "client_name": "palomma_inmobiliaria_xyz"
  },
  "duplicate": {
    "payload": { /* payload original recibido en el primer intento */ }
  }
}
```

5. Seguridad y Posibles Errores

El sistema realiza validaciones de autenticación, autorización por pasarela, configuración y consistencia del payload. Si alguna falla, devolverá un error con su respectivo código HTTP. La estructura siempre incluye `success: false`, `status`, `message` y, cuando aplica, `error_code`.

Código HTTP	Descripción
-------------	-------------

400	<p>Token faltante o inválido. Posibles causas:</p> <ul style="list-style-type: none"> — No se envió el encabezado <code>Authorization</code>. — El encabezado no tiene el formato <code>Bearer {token}</code>. — El token no fue encontrado en el sistema.
401	El token ha expirado. Debes generar uno nuevo consumiendo el servicio de Login .
403	El cliente OAuth autenticado no está registrado como pasarela activa en la inmobiliaria.
409	<p>Pago duplicado. El pago ya fue registrado previamente. <code>error_code = "DUPLICATE_PAYMENT"</code>. La pasarela debe tratar la respuesta como confirmación idempotente y no reintentar. Ver sección de Idempotencia.</p>
422	<p>Error de validación. <code>error_code = "VALIDATION_ERROR"</code>. Posibles causas:</p> <ul style="list-style-type: none"> — No se envió ni <code>factura_id</code> ni <code>movimiento_id</code>. — <code>factura_id</code> no es un entero positivo o no existe. — Algún <code>movimiento_id</code> no es un entero positivo o no existe. — Sin <code>factura_id</code> se envió más de un <code>movimiento_id</code>. — <code>monto</code> no es numérico o es menor o igual a cero. — <code>fecha_pago</code> no cumple el formato <code>YYYY-MM-DD HH:MM:SS</code>. — <code>n_comprobante</code> supera los 100 caracteres o contiene caracteres no permitidos. — La <code>factura_id</code> y los <code>movimiento_id</code> pertenecen a terceros distintos. — La pasarela no tiene una forma de pago configurada (<code>error_code = "GATEWAY_PAYMENT_METHOD_REQUIRED"</code>). — El cliente no corresponde a una pasarela registrada (<code>error_code = "GATEWAY_NOT_FOUND"</code>) o la pasarela está inactiva (<code>error_code = "GATEWAY_NOT_ACTIVE"</code>). — Errores de negocio retornados por el motor de ingresos (por ejemplo, factura ya pagada).
500	<p>Error interno al registrar el pago. <code>error_code = "INTERNAL_ERROR"</code>. El detalle se registra en el canal de log <code>PASARELAS</code>.</p>

6. Ejemplos de integración

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

cURL

```
# Pago de una factura
curl -X POST "https://{{instancia}}/service/v2/public/gateways/payments" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer TU_TOKEN_AQUI" \
```

```

-d '{
  "factura_id": 4521,
  "monto": 1750000.00,
  "fecha_pago": "2026-04-25 14:30:00",
  "n_comprobante": "PALM-2026-04-25-998877"
}'

# Pago de una factura junto con sus intereses adjuntos
curl -X POST "https://{{instancia}}/service/v2/public/gateways/payments" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer TU_TOKEN_AQUI" \
-d '{
  "factura_id": 4521,
  "movimiento_id": [98410, 98411],
  "monto": 1820000.00,
  "fecha_pago": "2026-04-25 14:30:00",
  "n_comprobante": "PALM-2026-04-25-998879"
}'

```

PHP

```

<?php

$instance = 'tu_instancia';
$token = 'TU_TOKEN_AQUI'; // Token de un cliente OAuth registrado como pasarela

$url = "https://{$instance}/service/v2/public/gateways/payments";

$payload = [
  'factura_id'    => 4521,
  'monto'        => 1750000.00,
  'fecha_pago'   => '2026-04-25 14:30:00',
  'n_comprobante' => 'PALM-2026-04-25-998877',
];

$ch = curl_init($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 HTTP: {$http_code}\n";
$data = json_decode($response, true);

if ($http_code === 200) {
    echo "Pago registrado. Recibo: {$data['data']['recibo_id']}\n";
} elseif ($http_code === 409) {
    echo "Pago ya existía previamente. confirm_pay_id:
{$data['data']['confirm_pay_id']}\n";
    // No reintentar: el pago ya está registrado en nuby.
} else {
    echo "Error registrando el pago:\n{$response}\n";
}

?>

```

Python

```

import requests

# 1. Configura tus datos de acceso
instancia = 'mi-inmobiliaria.nuby.app'
token = 'TU_TOKEN_AQUI' # Token de un cliente OAuth registrado como pasarela

# 2. Prepara el body del pago
url = f"https://{instancia}/service/v2/public/gateways/payments"

```

```

payload = {
  "factura_id": 4521,
  "monto": 1750000.00,
  "fecha_pago": "2026-04-25 14:30:00",
  "n_comprobante": "PALM-2026-04-25-998877",
}

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

# 3. Envía la petición POST y procesa la respuesta
try:
    response = requests.post(url, json=payload, headers=headers)
    print(f"Código HTTP: {response.status_code}")
    data = response.json()

    if response.status_code == 200:
        print(f"Pago registrado. Recibo: {data['data']['recibo_id']}")
    elif response.status_code == 409:
        print(f"Pago ya existía previamente. confirm_pay_id:
{data['data']['confirm_pay_id']}")
        # No reintentar: el pago ya está registrado en nuby.
    else:
        print("Error registrando el pago.")
        print(response.text)
except Exception as e:
    print(f"Error de conexión: {e}")

```

JavaScript

```

// 1. Configura tus datos de acceso
const instancia = 'mi-inmobiliaria.nuby.app';
const token = 'TU_TOKEN_AQUI'; // Token de un cliente OAuth registrado como pasarela

// 2. Prepara el body del pago

```

```
const url = `https://${instancia}/service/v2/public/gateways/payments`;

const payload = {
  factura_id: 4521,
  monto: 1750000.00,
  fecha_pago: '2026-04-25 14:30:00',
  n_comprobante: 'PALM-2026-04-25-998877',
};

// 3. Función para registrar el pago
async function registrarPago() {
  try {
    const response = await fetch(url, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
        'Authorization': `Bearer ${token}`,
      },
      body: JSON.stringify(payload),
    });

    console.log(`Código HTTP: ${response.status}`);
    const data = await response.json();

    if (response.status === 200) {
      console.log(`Pago registrado. Recibo: ${data.data.recibo_id}`);
    } else if (response.status === 409) {
      console.log(`Pago ya existía previamente. confirm_pay_id:
${data.data.confirm_pay_id}`);
      // No reintentar: el pago ya está registrado en nuby.
    } else {
      console.log('Error registrando el pago.');
```

```
registrarPago();
```

Obtener Cartera

Permite a una pasarela de pagos autenticada (por ejemplo, Palomma) consultar la **cartera pagable** de un tercero: facturas pendientes, conceptos facturables, conceptos no facturables e intereses de mora adjuntos a facturas de canon. El sistema aplica automáticamente la configuración de filtros avanzados registrada para la pasarela autenticada.

¿Para qué sirve este servicio?

Está pensado exclusivamente para integraciones con pasarelas de pago. La pasarela lo consume para mostrarle al deudor los ítems que puede pagar en línea (facturas, conceptos e intereses), respetando la configuración comercial de la inmobiliaria (qué incluir, qué excluir, qué resoluciones se admiten, etc.).

Endpoint restringido a pasarelas registradas

Este endpoint **solo** puede ser consumido por clientes OAuth2 cuyo `client_name` esté registrado como una pasarela activa en la inmobiliaria. Cualquier otro cliente recibirá `403 Forbidden`.

1. El Endpoint (La dirección web)

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

```
GET https://{{instancia}}/service/v2/public/gateways/portfolio
```

¿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 obtenido por OAuth2. La pasarela se identifica automáticamente a partir del `client_name` del token; **no debes enviarlo en el body ni como query string**.

Método	GET
Content-Type	application/json
Authorization	Bearer token , Token obtenido al consumir el servicio Login con un cliente OAuth2 registrado como pasarela activa.

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**. El cliente OAuth además debe estar registrado como pasarela activa; en caso contrario el endpoint responderá `403`.

Parámetros de consulta (query string):

Parámetro	Tipo	Requerido	Por defecto	Descripción
documento	string	No	—	Número de documento del tercero (deudor) cuya cartera se desea consultar. Si se omite, se devolverá la cartera de todos los terceros que cumplan los demás filtros.
page	integer	No	1	Número de página de resultados que se desea recuperar.
page_size	integer	No	10	Número máximo de ítems por página. Valor máximo permitido: 1000 .
pendientes	boolean	No	true	Si es <code>true</code> , se devuelven únicamente ítems con <code>estado = "pendiente"</code> y <code>saldo > 0</code> .

Parámetro	Tipo	Requerido	Por defecto	Descripción
incluir_borrador	boolean	No	false	Si es <code>true</code> , incluye facturas en estado Borrador en el resultado.
incluir_conceptos_n o_factura	boolean	No	false	Si es <code>true</code> , incluye conceptos pagables que aún no tienen factura emitida.
incluir_conceptos_f uturos	boolean	No	false	Si es <code>true</code> , incluye conceptos cuyo periodo de aplicación es posterior a la fecha actual (cobros anticipados).
rango_conceptos_f uturos_meses	integer	No	null	Cantidad de meses hacia adelante en los que se admite incluir conceptos futuros. Solo aplica si <code>incluir_conceptos_futuros = true</code> . Use <code>null</code> para no acotar.
incluir_intereses_m ora	boolean	No	false	Si es <code>true</code> , los intereses de mora de una factura se devuelven adjuntos como ítems internos de su factura padre.
resolucion_ids	array string	No	—	Lista de IDs de resoluciones de facturación admitidas. Acepta un arreglo o una cadena separada por comas (por ejemplo: <code>"3,7,12"</code>). Solo se devuelven facturas emitidas con esas resoluciones.

Parámetro	Tipo	Requerido	Por defecto	Descripción
<code>agrupar_conceptos_periodo</code>	boolean	No	false	Si es <code>true</code> , consolida los conceptos del mismo contrato, tercero y periodo (mismas fechas de inicio y fin) en un único ítem pagable. Los conceptos negativos del periodo solo se descuentan si existe un concepto positivo con saldo suficiente. Requiere que al menos uno de <code>incluir_conceptos_no_factura</code> o <code>incluir_conceptos_futuros</code> esté activo; en caso contrario el filtro se ignora.

Filtros avanzados de la pasarela

Los filtros enviados por el caller se combinan con la **configuración de filtros avanzados** registrada para la pasarela autenticada en la inmobiliaria. Si la pasarela no tiene configuración propia, se aplica el comportamiento por defecto (solo facturas con estado pendiente, sin borradores, sin conceptos sueltos ni intereses adjuntos). Los filtros realmente aplicados se devuelven en la respuesta dentro de `filters_applied`.

Exclusión de facturas a propietario

El endpoint **excluye automáticamente** las facturas cuyos detalles estén asociados a movimientos de tipo `FACTURA_PROPIETARIO` (por ejemplo, comisiones del contrato facturadas al propietario). Estas facturas siguen una lógica contable distinta —generan `EGRESO_PROPIETARIO` en lugar de `INGRESO_INQUILINO`— y no son pagables vía pasarela en esta versión. No se incluyen en la respuesta ni en el conteo total.

Ejemplos de peticiones:

```
GET https://{{instancia}}/service/v2/public/gateways/portfolio
```

```
GET https://{{instancia}}/service/v2/public/gateways/portfolio?documento=1020304050
```

GET

```
https://{{instancia}}/service/v2/public/gateways/portfolio?documento=1020304050&pendientes=true&page=1&page_size=20
```

GET

```
https://{{instancia}}/service/v2/public/gateways/portfolio?documento=1020304050&incluir_intereses_mora=true&incluir_conceptos_futuros=true&rango_conceptos_futuros_meses=3
```

GET

```
https://{{instancia}}/service/v2/public/gateways/portfolio?documento=1020304050&resolucion_ids=3,7
```

3. La Respuesta (¿Qué te entregaremos?)

El sistema devolverá una respuesta JSON con los ítems pagables del tercero, los filtros realmente aplicados y la información de paginación. Cada ítem se entrega con la **misma estructura canónica**, sin importar si representa una factura de contrato, una factura de venta, un concepto facturable, un concepto no facturable o una factura con intereses adjuntos.

```
{
  "success": true,
  "status": 200,
  "message": "La cartera de la pasarela fue consultada correctamente.",
  "body": [
    {
      "factura_id": 4521,
      "factura_numero": "291",
      "contrato_id": 187,
      "contrato_numero": "C-000187",
      "tercero_id": 272,
      "tercero_documento": "1020304050",
      "tercero_nombre": "ANA MARÍA PÉREZ RODRÍGUEZ",
      "fecha_vencimiento": "2026-04-30",
      "observaciones": "Factura del Contrato C-000187",
      "valor_total": 1750000.00,
      "saldo": 1750000.00,
      "estado": "pendiente",
      "items": [
        {
```

```
    "movimiento_id": 98231,
    "descripcion": "Canon de arrendamiento - Canon abril 2026",
    "valor_unitario": 1600000.00,
    "cantidad": 1,
    "valor_iva": 0.00,
    "valor_retencion": 0.00,
    "valor_reteiva": 0.00,
    "valor_reteica": 0.00,
    "valor_descuento": 0.00
  },
  {
    "movimiento_id": 98232,
    "descripcion": "Administración - Cuota administración abril 2026",
    "valor_unitario": 150000.00,
    "cantidad": 1,
    "valor_iva": 0.00,
    "valor_retencion": 0.00,
    "valor_reteiva": 0.00,
    "valor_reteica": 0.00,
    "valor_descuento": 0.00
  },
  {
    "movimiento_id": 98410,
    "descripcion": "Intereses de mora - Mora canon marzo 2026",
    "valor_unitario": 35000.00,
    "cantidad": 1,
    "valor_iva": 0.00,
    "valor_retencion": 0.00,
    "valor_reteiva": 0.00,
    "valor_reteica": 0.00,
    "valor_descuento": 0.00
  }
]
},
{
  "factura_id": null,
  "factura_numero": null,
  "contrato_id": 187,
  "contrato_numero": "C-000187",
  "tercero_id": 272,
```

```
    "tercero_documento": "1020304050",
    "tercero_nombre": "ANA MARÍA PÉREZ RODRÍGUEZ",
    "fecha_vencimiento": "2026-05-15",
    "observaciones": "Costo del Contrato C-000187",
    "valor_total": 280000.00,
    "saldo": 280000.00,
    "estado": "pendiente",
    "items": [
      {
        "movimiento_id": 98515,
        "descripcion": "Reparación locativa - Reparación locativa baño principal",
        "valor_unitario": 280000.00,
        "cantidad": 1,
        "valor_iva": 0.00,
        "valor_retencion": 0.00,
        "valor_reteiva": 0.00,
        "valor_reteica": 0.00,
        "valor_descuento": 0.00
      }
    ]
  },
  "filters_applied": {
    "documento": "1020304050",
    "pendientes": true,
    "incluir_borrador": false,
    "incluir_conceptos_no_factura": true,
    "incluir_conceptos_futuros": false,
    "incluir_intereses_mora": true
  },
  "pagination": {
    "total_records": 2,
    "total_pages": 1,
    "current_page": 1,
    "page_size": 10,
    "current_page_records": 2,
    "has_next_page": false,
    "has_previous_page": false
  }
}
```

Campos de la respuesta

Clave	Descripción
success	Indica si la operación fue exitosa (<code>true</code>) o no (<code>false</code>).
status	Código HTTP devuelto (debe coincidir con el código real de la respuesta).
message	Mensaje descriptivo del resultado de la operación.
body	Arreglo de ítems pagables (ver tabla detallada abajo). Puede ser vacío si no hay cartera.
filters_applied	Objeto con los filtros realmente aplicados (combinación de los filtros del caller + la configuración de la pasarela).
pagination	Objeto con la información de paginación (ver tabla detallada abajo).

Estructura de cada ítem (`body[]`)

Clave	Descripción
factura_id	Identificador interno de la factura (PK en base de datos). <code>null</code> cuando el ítem corresponde a un concepto sin factura emitida (concepto facturable o no facturable). No es el número visible al usuario ; ver <code>factura_numero</code> .
factura_numero	Número consecutivo visible de la factura (mismo valor que se muestra como "Factura N°" en la interfaz de nuby). <code>null</code> cuando el ítem es un concepto sin factura emitida.
contrato_id	Identificador del contrato asociado. <code>null</code> para facturas de venta sin contrato.
contrato_numero	Número/código visible del contrato. <code>null</code> si no aplica.
tercero_id	Identificador del tercero (deudor) al que pertenece el ítem.
tercero_documento	Número de documento de identificación del tercero.
tercero_nombre	Nombre completo (o razón social) del tercero.
fecha_vencimiento	Fecha de vencimiento del ítem. Formato: <code>YYYY-MM-DD</code> .

Clave	Descripción
observaciones	<p>Mensaje genérico que identifica la naturaleza del ítem. Valores posibles:</p> <ul style="list-style-type: none"> "Factura del Contrato {numero}" — cuando el ítem es una factura asociada a un contrato (reemplaza {numero} por el consecutivo del contrato). "Factura de Venta" — cuando el ítem es una factura de venta sin contrato asociado. "Costo del Contrato {numero}" — cuando el ítem corresponde a un concepto/movimiento de un contrato sin factura emitida. <p>El detalle por concepto/producto se expone en el campo <code>descripcion</code> de cada renglón de <code>items[]</code>.</p>
valor_total	Valor total original del ítem (suma de los renglones internos).
saldo	Saldo pendiente por pagar a la fecha de la consulta.
estado	Estado lógico del ítem. Valores posibles: "pendiente", "pagada", "anulada". Las facturas con estados internos fuera de alcance se omiten del resultado.
items	Arreglo de renglones internos del ítem (ver tabla detallada abajo). Para facturas, contiene los movimientos que la componen y, si aplica, los intereses de mora adjuntos.

Mapeo de estados de factura

Estado interno	Estado expuesto	Descripción
1 (Facturada)	"pendiente"	Factura emitida y aún sin pagar.
5 (Borrador)	"pendiente"	Factura en estado borrador. Solo aparece si <code>incluir_borrador = true</code> .
2 (Anulada)	"anulada"	Factura anulada manualmente.
6 (Anulada por NC)	"anulada"	Factura anulada por nota crédito.
3 (Pagada)	"pagada"	Factura ya cancelada en su totalidad.
4, 7	—	Estados fuera de alcance. La factura se excluye del resultado.

Estructura de cada renglón interno (`body[].items[]`)

Clave	Descripción
movimiento_id	Identificador del movimiento de contrato asociado al renglón. Puede ser <code>null</code> o <code>0</code> para renglones de facturas de venta sin movimiento.

Clave	Descripción
descripcion	Texto descriptivo del renglón con el formato "{producto} - {descripcion}", donde {producto} es el nombre del producto/concepto y {descripcion} es la descripción puntual de ese ítem (mes facturado, observación específica, etc.). Si alguno de los componentes no existe se incluye vacío respetando el separador.
valor_unitario	Valor unitario del renglón (base, sin IVA ni retenciones).
cantidad	Cantidad facturada.
valor_iva	Valor del IVA aplicado al renglón.
valor_retencion	Valor de la retención en la fuente aplicada.
valor_reteiva	Valor de la retención de IVA aplicada.
valor_reteica	Valor de la retención de ICA aplicada.
valor_descuento	Valor del descuento aplicado al renglón.

Información de paginación (`pagination`)

Clave	Descripción
total_records	Cantidad total de ítems pagables que cumplen los filtros (sin paginar).
total_pages	Cantidad total de páginas calculadas con el <code>page_size</code> solicitado.
current_page	Página actualmente devuelta.
page_size	Tamaño de página efectivamente aplicado.
current_page_records	Cantidad de ítems devueltos en la página actual.
has_next_page	<code>true</code> si existe una página siguiente.
has_previous_page	<code>true</code> si existe una página anterior.

4. Seguridad y Posibles Errores

El sistema realiza validaciones de autenticación, autorización por pasarela y parámetros. Si alguna falla, devolverá un error con su respectivo código HTTP. La estructura del cuerpo de error siempre incluye `success: false`, `status`, `message` y `body: []`.

Código HTTP	Descripción
-------------	-------------

400	<p>Token faltante o inválido. Posibles causas:</p> <ul style="list-style-type: none"> — No se envió el encabezado <code>Authorization</code>. — El encabezado no tiene el formato <code>Bearer {token}</code>. — El token no fue encontrado en el sistema. <p>Parámetros inválidos. Posibles causas:</p> <ul style="list-style-type: none"> — <code>page</code> no es un entero positivo. — <code>page_size</code> no es un entero positivo.
401	<p>El token ha expirado. Debes generar uno nuevo consumiendo el servicio de Login.</p>
403	<p>El cliente OAuth autenticado no está registrado como pasarela activa en la inmobiliaria. Mensaje: "El cliente autenticado no esta autorizado para consumir este endpoint."</p>
500	<p>Error interno al resolver la cartera. El detalle se registra en el canal de log <code>PASARELAS</code> y al caller solo se le devuelve un mensaje genérico.</p>

5. Ejemplos de integración

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

cURL

```
# Consultar la cartera del tercero con documento 1020304050
curl -X GET
"https://{{instancia}}/service/v2/public/gateways/portfolio?documento=1020304050&pendiente
s=true&page=1&page_size=20" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer TU_TOKEN_AQUI"

# Cartera incluyendo intereses de mora y conceptos futuros (3 meses)
curl -X GET
"https://{{instancia}}/service/v2/public/gateways/portfolio?documento=1020304050&incluir_i
ntereses_mora=true&incluir_conceptos_futuros=true&rango_conceptos_futuros_meses=3" \
-H "Content-Type: application/json" \
-H "Authorization: Bearer TU_TOKEN_AQUI"
```

PHP

```

<?php

$instance = 'tu_instancia'; // Reemplaza con tu instancia real
$token = 'TU_TOKEN_AQUI'; // Token obtenido del servicio Login (cliente OAuth registrado
como pasarela)

$queryParams = http_build_query([
    'documento' => '1020304050',
    'pendientes' => 'true',
    'incluir_intereses_mora' => 'true',
    'page' => 1,
    'page_size' => 20,
]);

$url = "https://{instance}/service/v2/public/gateways/portfolio?{$queryParams}";

$ch = curl_init($url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($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 HTTP: {$http_code}\n";

if ($http_code === 200) {
    $data = json_decode($response, true);
    echo "Total ítems: " . ($data['pagination']['total_records'] ?? 0) . "\n";
    foreach ($data['body'] as $item) {
        $tipo = $item['factura_id'] ? "Factura #{$item['factura_id']}" : "Concepto";
        echo "  [{$tipo}] {$item['tercero_nombre']} - Saldo: {$item['saldo']} - Vence:
{$item['fecha_vencimiento']}\n";
    }
} else {
    echo "Error consultando la cartera:\n{$response}\n";
}

```

```
}  
  
?>
```

Python

```
import requests  
  
# 1. Configura tus datos de acceso  
instancia = 'mi-inmobiliaria.nuby.app'  
token = 'TU_TOKEN_AQUI' # Token de un cliente OAuth registrado como pasarela  
  
# 2. Prepara la dirección y los parámetros  
url = f"https://{instancia}/service/v2/public/gateways/portfolio"  
  
params = {  
    "documento": "1020304050",  
    "pendientes": "true",  
    "incluir_intereses_mora": "true",  
    "page": 1,  
    "page_size": 20,  
}  
  
headers = {  
    "Content-Type": "application/json",  
    "Authorization": f"Bearer {token}",  
}  
  
# 3. Envía la petición GET y procesa la respuesta  
try:  
    response = requests.get(url, params=params, headers=headers)  
    print(f"Código HTTP: {response.status_code}")  
  
    if response.status_code == 200:  
        data = response.json()  
        print(f"Total ítems: {data['pagination']['total_records']}")  
        for item in data['body']:
```

```
        tipo = f"Factura #{item['factura_id']}" if item['factura_id'] else "Concepto"
        print(f"  [{tipo}] {item['tercero_nombre']} - Saldo: {item['saldo']} - Vence:
{item['fecha_vencimiento']}")
    else:
        print("Error consultando la cartera.")
        print(response.text)
except Exception as e:
    print(f"Error de conexión: {e}")
```

JavaScript

```
// 1. Configura tus datos de acceso
const instancia = 'mi-inmobiliaria.nuby.app';
const token = 'TU_TOKEN_AQUI'; // Token de un cliente OAuth registrado como pasarela

// 2. Prepara la dirección con parámetros
const params = new URLSearchParams({
  documento: '1020304050',
  pendientes: 'true',
  incluir_intereses_mora: 'true',
  page: 1,
  page_size: 20,
});

const url = `https://${instancia}/service/v2/public/gateways/portfolio?${params}`;

// 3. Función para consultar la cartera
async function consultarCartera() {
  try {
    const response = await fetch(url, {
      method: 'GET',
      headers: {
        'Content-Type': 'application/json',
        'Authorization': `Bearer ${token}`,
      },
    });
  }

  console.log(`Código HTTP: ${response.status}`);
```

```
if (response.ok) {
  const data = await response.json();
  console.log(`Total ítems: ${data.pagination.total_records}`);
  data.body.forEach(item => {
    const tipo = item.factura_id ? `Factura #${item.factura_id}` : 'Concepto';
    console.log(` [${tipo}] ${item.tercero_nombre} - Saldo: ${item.saldo} -
Vence: ${item.fecha_vencimiento}`);
  });
} else {
  const errorData = await response.text();
  console.log('Error consultando la cartera.');
```

```
  console.log(errorData);
}
} catch (error) {
  console.error(`Error de conexión: ${error}`);
}
}

consultarCartera();
```