

Listar Notas [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 notas de crédito y débito** registradas en nuby, con sus datos principales. Puedes filtrar por factura, por tipo de nota y por rango de fechas.

Úsalo para conciliar las notas de una factura con tu plataforma, para construir un historial de ajustes o para verificar qué notas se crearon en un período.

¿Para qué sirve este servicio?
Úsalo cuando necesites **recorrer las notas registradas en nuby**. Los casos de uso más frecuentes son:

- **Conciliar las notas de una factura** con tu plataforma, filtrando por `factura_id`.
- **Construir un historial de ajustes** separando las notas de crédito de las de débito con `tipo`.
- **Cerrar un período contable** revisando las notas creadas entre dos fechas con `desde` y `hasta`.
- **Recorrer resultados largos por páginas** con `page` y `page_size`.
- **Obtener el `nota_id`** para consultar el detalle solo de las notas que vas a mostrar.

1. El Endpoint (La dirección web)

Método y dirección

Dato	Valor
Método	<code>GET</code>
Recurso	<code>/notes/list</code>
Cuerpo de la petición	No se envía cuerpo

GET https://{{instancia}}/service/v2/public/notes/list

¿Qué debo colocar en {{instancia}}?

La dirección web principal con la que ingresas a tu plataforma, incluyendo la terminación `.nuby.app` o `.arrendasoft.co`.

Importante: no incluyas `https://` ni barras diagonales (`/`) al final. La dirección se compone así: `https://` + tu instancia + `/service/v2/public/notes/list`.

Encabezados requeridos

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

Permiso requerido (scope): `catalogs:read`

Este servicio es de **solo lectura** y exige el permiso `catalogs:read`. Si falta, la respuesta es `403` con `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 son opcionales y se envían en la dirección web. Sin filtros, recibes la primera página con las últimas 50 notas.

Parámetro	Tipo	Requerido	Default	Descripción
<code>factura_id</code>	entero	No	—	Devuelve solo las notas de esa factura.
<code>tipo</code>	texto	No	—	<code>credito</code> o <code>debito</code> .
<code>desde</code>	fecha	No	—	Fecha inicial en formato <code>AAAA-MM-DD</code> (inclusive).
<code>hasta</code>	fecha	No	—	Fecha final en formato <code>AAAA-MM-DD</code> (inclusive).

Parámetro	Tipo	Requerido	Default	Descripción
page	entero	No	1	Número de página, empezando en 1.
page_size	entero	No	50	Registros por página. Máximo 1000.

Importante: `page`, `page_size` y `factura_id` deben ser enteros positivos; `desde` no puede ser mayor que `hasta`. Un valor inválido produce `400`.

Los filtros se combinan. Si envías `factura_id` y `tipo`, deben cumplirse ambos.

Ejemplos de dirección

```
GET /notes/list?factura_id=456
GET /notes/list?tipo=credito&desde=2026-10-01&hasta=2026-10-31
GET /notes/list?page=2&page_size=20
```

Validaciones de parámetros

Los parámetros se validan antes de ejecutar la consulta. Si uno no cumple su regla, la respuesta es `400 VALIDATION_ERROR` y no se devuelve ningún dato:

Parámetro	Regla	Mensaje <code>400</code>
page	Entero positivo. <code>0</code> , valores negativos o texto no son válidos.	El parámetro "page" debe ser un número positivo.
page_size	Entero positivo, con <code>1000</code> como máximo.	El parámetro "page_size" debe ser un número positivo.
factura_id	Entero positivo. Si apunta a una factura de contrato, la respuesta es <code>409 FACTURA_CONTRATO_NO_SOPORTADA</code> .	<code>VALIDATION_ERROR</code> con el mensaje del parámetro.
tipo	Solo admite <code>credito</code> o <code>debito</code> .	<code>VALIDATION_ERROR</code> : el valor no está permitido.
desde, hasta	Fecha en formato <code>AAAA-MM-DD</code> .	<code>VALIDATION_ERROR</code> con el mensaje del parámetro.
Rango de fechas	<code>desde</code> no puede ser mayor que <code>hasta</code> .	<code>VALIDATION_ERROR</code> por rango inválido.

Cuándo usar cada filtro

Filtro	Caso de uso	Precisión
<code>factura_id</code>	Conciliar todas las notas de una factura concreta.	Exacta: devuelve solo las notas de esa factura.
<code>tipo</code>	Separar los ajustes que reducen el saldo de los que lo aumentan.	Exacta: <code>credito</code> o <code>debito</code> .
<code>desde</code> / <code>hasta</code>	Cerrar un período contable o revisar los ajustes de un mes.	Por rango de fechas, con ambos extremos incluidos.
<code>factura_id</code> + <code>tipo</code>	Ver solo las notas de crédito o solo las de débito de una factura.	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: solo controla el tamaño y la posición de la página.

3. La Respuesta (¿Qué te entregaremos?)

```
{
  "status": 200,
  "message": null,
  "body": [
    {
      "nota_id": 789,
      "factura_id": 456,
      "tipo": "credito",
      "consecutivo": 1024,
      "fecha": "2026-10-21",
      "valor_total": 59500,
      "estado_id": 5,
      "clasificacion": 1,
      "concepto_id": 6,
      "observaciones": "Ajuste acordado"
    }
  ],
  "pagination": {
    "page": 1,
    "page_size": 50,
    "total": 1,
    "total_pages": 1
  },
  "filters_applied": {
    "factura_id": 456,
    "tipo": "credito"
  }
}
```

```
}  
}
```

Variantes de respuesta

El envelope cambia según los filtros enviados y según haya o no coincidencias. Estos son los tres casos que debes manejar en tu integración:

Con filtros

```
{  
  "status": 200,  
  "message": null,  
  "body": [  
    {  
      "nota_id": 789,  
      "factura_id": 456,  
      "tipo": "credito",  
      "consecutivo": 1024,  
      "fecha": "2026-10-21",  
      "valor_total": 59500,  
      "estado_id": 5,  
      "clasificacion": 1,  
      "concepto_id": 6,  
      "observaciones": "Ajuste acordado"  
    }  
  ],  
  "pagination": { "page": 1, "page_size": 50, "total": 1, "total_pages": 1 },  
  "filters_applied": { "factura_id": 456, "tipo": "credito" }  
}
```

Sin filtros

Cuando no envías ningún filtro, `filters_applied` es `null` y recibes la primera página de todas las notas de venta, ordenadas por fecha descendente.

```
{  
  "status": 200,  
  "message": null,  
  "body": [  
    {  

```

```

    "nota_id": 789,
    "factura_id": 456,
    "tipo": "credito",
    "consecutivo": 1024,
    "fecha": "2026-10-21",
    "valor_total": 59500,
    "estado_id": 5,
    "clasificacion": 1,
    "concepto_id": 6,
    "observaciones": "Ajuste acordado"
  },
  {
    "nota_id": 790,
    "factura_id": 457,
    "tipo": "debito",
    "consecutivo": 1025,
    "fecha": "2026-10-20",
    "valor_total": 20000,
    "estado_id": 5,
    "clasificacion": 1,
    "concepto_id": 8,
    "observaciones": "Cargo adicional"
  }
],
"pagination": { "page": 1, "page_size": 50, "total": 2, "total_pages": 1 },
"filters_applied": null
}

```

Lista vacía

No es un error: el servicio responde `200`, `body` vacío y un mensaje informativo. En este ejemplo sí se enviaron filtros, por eso `filters_applied` no es `null`.

```

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

```

```
"desde": "2026-09-01",
"hasta": "2026-09-30"
}
}
```

Campos de cada nota

Campo	Descripción
<code>nota_id</code>	Identificador interno de la nota. Úsalo para consultar el detalle.
<code>factura_id</code>	Factura anotada.
<code>tipo</code>	<code>credito</code> o <code>debito</code> .
<code>consecutivo</code>	Número de la nota.
<code>fecha</code>	Fecha de la nota.
<code>valor_total</code>	Valor total de la nota.
<code>estado_id</code>	<code>5</code> = borrador.
<code>clasificacion</code>	<code>1</code> = vinculante.
<code>concepto_id</code>	Concepto aplicado a la nota.
<code>observaciones</code>	Observaciones registradas.

Campos de paginación

Campo	Significado	Caso sin resultados
<code>page</code>	Número de página devuelta, empezando en <code>1</code> .	Conserva la página solicitada (<code>1</code> por defecto).
<code>page_size</code>	Cantidad de registros por página efectivamente aplicada.	Conserva el tamaño solicitado (<code>50</code> por defecto).
<code>total</code>	Total de notas que cumplen los filtros.	<code>0</code> .
<code>total_pages</code>	Total de páginas disponibles para ese resultado.	<code>0</code> . Úsalo como condición de fin al recorrer páginas.

`filters_applied` y `message`

- Sin filtros, `filters_applied` es `null`.
- Con filtros, es un objeto con los filtros usados.
- Con resultados, `message` es `null`.
- Sin resultados, `message` es `No se encontraron notas con los filtros aplicados.` y `body` es `[]`.

Campos del envelope

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

Una lista vacía no es un error. Se responde `200`; revisa `pagination.total` para confirmar que no hay coincidencias.

Cómo interpretar el resultado

Señal en la respuesta	Qué significa	Qué hacer
<code>body</code> con notas y <code>message: null</code>	La consulta encontró coincidencias.	Procesa cada nota y, si hay más páginas, avanza con <code>page</code> .
<code>body: []</code> y <code>total: 0</code>	Ninguna nota cumple los filtros enviados.	Revisa los filtros; no lo trates como un fallo técnico.
<code>filters_applied: null</code>	No se aplicó ningún filtro.	Confirma que enviaste los parámetros en la dirección web.
<code>400 VALIDATION_ERROR</code>	Un parámetro no cumple su regla y la consulta no se ejecutó.	Corrige el parámetro que indica el mensaje y repite.
<code>409 FACTURA_CONTRATO_NO_SOPORTADA</code>	El <code>factura_id</code> pertenece a una factura de contrato.	Gestiona esas notas desde el flujo de contratos; no reintentes igual.

Orden: las notas se devuelven por fecha descendente y, a igualdad de fecha, por identificador descendente. El orden es estable entre páginas.

4. Seguridad y Posibles Errores

Reglas y exclusiones

La API solo expone notas de **facturas de venta**:

- El listado **nunca** incluye notas de facturas de contrato, aunque no envíes `factura_id`.
- Si filtras por una factura de contrato, la respuesta es `409 FACTURA_CONTRATO_NO_SOPORTADA` con `warnings[]`.

La nota se devuelve con el estado que tiene en nuby. Las notas creadas por la API están en **borrador** hasta que un usuario las emite en la interfaz.

Errores del contrato de la operación

HTTP	error_code	Cuándo ocurre y cómo resolverlo
400	VALIDATION_ERROR	Algún parámetro tiene un valor inválido. Revisa <code>page</code> , <code>page_size</code> , <code>factura_id</code> , <code>tipo</code> , <code>desde</code> o <code>hasta</code> .
401	—	Falta el token o expiró. Vuelve a consumir Login .
403	RECURSO_NO_AUTORIZADO	Al cliente OAuth le falta <code>catalogs:read</code> .
409	FACTURA_CONTRATO_NO_SOPORTADA	Filtraste por una factura de contrato. Incluye <code>warnings[]</code> .
500	INTERNAL_ERROR	Fallo interno al consultar. Reintenta; si persiste, contacta a soporte.

Errores frecuentes al paginar

Situación	Qué ocurre	Cómo corregir
Envías <code>page=0</code> o <code>page=abc</code>	<code>400 VALIDATION_ERROR</code> ; la consulta no se ejecuta.	Usa un entero desde <code>1</code> .
Envías <code>page_size=0</code> o un valor negativo	<code>400 VALIDATION_ERROR</code> .	Usa un entero positivo, con <code>1000</code> como máximo.
Recorres páginas hasta recibir una vacía	Puedes hacer una petición de más o detenerte antes de tiempo.	Recorre hasta <code>pagination.total_pages</code> .
Esperas que una lista vacía traiga <code>filters_applied: null</code>	<code>filters_applied</code> refleja los filtros enviados, no el resultado.	Revisa <code>body</code> y <code>pagination.total</code> para detectar la lista vacía.

Atención: una factura de contrato no es un error de parámetro: el servicio la rechaza explícitamente para no exponer notas de ese flujo.

5. Ejemplos de integración

cURL

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

# Notas de una factura
curl -G "https://${INSTANCIA}/service/v2/public/notes/list" \
  --data-urlencode "factura_id=456" \
  -H "Authorization: Bearer ${TOKEN}"

# Notas de crédito de octubre, página 1 de 20
curl -G "https://${INSTANCIA}/service/v2/public/notes/list" \
  --data-urlencode "tipo=credito" \
  --data-urlencode "desde=2026-10-01" \
  --data-urlencode "hasta=2026-10-31" \
  --data-urlencode "page=1" \
  --data-urlencode "page_size=20" \
  -H "Authorization: Bearer ${TOKEN}"
```

PHP

```
<?php

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

$query = http_build_query([
    'factura_id' => 456,
    'page' => 1,
    'page_size' => 50,
]);

$url = "https://{ $instancia }/service/v2/public/notes/list?{$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 $nota) {
    echo $nota['nota_id'] . ' - ' . $nota['tipo'] . ' - ' . $nota['valor_total'] . "\n";
}

```

Python

```

import requests

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

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

pagina = 1
total_paginas = 1

while pagina <= total_paginas:
    respuesta = requests.get(
        url,

```

```

        headers=headers,
        params={"tipo": "credito", "desde": "2026-10-01", "hasta": "2026-10-31",
                "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 nota in datos["body"]:
        print(nota["nota_id"], nota["tipo"], nota["valor_total"])

    pagina += 1

```

JavaScript (Fetch API)

```

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

const params = new URLSearchParams({
    factura_id: '456',
    page: '1',
    page_size: '50'
});

const url = `https://${instancia}/service/v2/public/notes/list?${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 ${data.pagination.total_pages}`);  
  data.body.forEach(n => console.log(n.nota_id, n.tipo, n.valor_total));  
} else {  
  console.error(`${respuesta.status} ${data.error_code}: ${data.message}`);  
}
```

Power Query M (Excel / Power BI)

Power Query consulta una página por vez. El paso `pagination` te permite parametrizar la página y combinar varias con `Table.Combine` si necesitas el historial completo.

```
let  
  // 1. Configura tus datos de acceso  
  instancia = "tu-inmobiliaria.nuby.app",  
  token = "TU_TOKEN_AQUI",  
  
  // 2. Prepara la dirección con los filtros y la página  
  url = "https://" & instancia &  
  "/service/v2/public/notes/list?factura_id=456&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",
        {"nota_id", "factura_id", "tipo", "consecutivo", "fecha", "valor_total",
        "estado_id", "clasificacion", "concepto_id", "observaciones"},
        {"NotaID", "FacturaID", "Tipo", "Consecutivo", "Fecha", "ValorTotal",
        "EstadoID", "Clasificacion", "ConceptoID", "Observaciones"}),
    tipos = Table.TransformColumnTypes(expandido, {
        {"NotaID", Int64.Type},
        {"FacturaID", Int64.Type},
        {"Consecutivo", Int64.Type},
        {"Fecha", type date},
        {"ValorTotal", type number}
    }),
    totalPaginas = pagination[total_pages]
in
    tipos

```

6. Buenas prácticas y límites conocidos

- **Filtra por** `factura_id` cuando busques las notas de un documento concreto.
- **Usa** `desde` y `hasta` **en lugar de paginar meses completos.**
- **Recorre hasta** `total_pages`, no hasta una página vacía.
- **No uses la lista como detalle:** los ítems y el saldo de la factura están en la página [Buscar Nota por Código](#).
- **Guarda el** `nota_id` para evitar búsquedas repetidas.

Límites del listado

- Solo expone notas de **facturas de venta**; las de contrato nunca aparecen.
- No devuelve los ítems ni el saldo de la factura: eso está en el detalle de la nota.
- El estado corresponde al que tiene la nota en nuby; las creadas por la API están en borrador.

Si necesitas crear una nota, revisa la página [Crear Nota](#).

