

Notas débito y crédito [BETA]

Bienvenido al **módulo de notas** de tu inmobiliaria. En esta sección centralizamos las conexiones para que tus plataformas externas ajusten o anulen una factura de venta sin volver a digitar el documento en la interfaz de nuby.

¿Qué puedes lograr en este módulo?

- 1. Ajustar la cartera:** tu aplicación puede crear una **nota de crédito** para reducir el saldo de una factura o una **nota de débito** para aumentarlo. nuby calcula el IVA, las retenciones y el descuento de cada ítem, recalcula el total y actualiza el saldo de la factura en el acto. Si una nota de crédito deja el saldo en cero, la factura queda anulada por nota crédito.
- 2. Consulta sincronizada:** podrás listar las notas de una factura o de un rango de fechas, filtrar por tipo de nota y obtener el detalle con sus ítems, impuestos y el estado y saldo resultantes de la factura.
- 3. Control sin sorpresas:** la nota nace en **borrador** y la emisión a la DIAN la realiza un usuario en nuby. La API solo opera con facturas de venta activas: las facturas de contrato se rechazan de forma explícita, y cada operación es idempotente para que un reintento no duplique el documento.

Explora las opciones a continuación para automatizar los ajustes de tu facturación y mantener tu cartera conciliada.

- [Listar Notas \[BETA\]](#)
- [Buscar Nota por Código \[BETA\]](#)
- [Crear Nota \[BETA\]](#)

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	GET
Recurso	/notes/list
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 400
page	Entero positivo. 0 , valores negativos o texto no son válidos.	El parámetro "page" debe ser un número positivo.
page_size	Entero positivo, con 1000 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 409 FACTURA_CONTRATO_NO_SOPORTADA .	VALIDATION_ERROR con el mensaje del parámetro.
tipo	Solo admite credito o debito .	VALIDATION_ERROR : el valor no está permitido.
desde , hasta	Fecha en formato AAAA-MM-DD .	VALIDATION_ERROR con el mensaje del parámetro.
Rango de fechas	desde no puede ser mayor que hasta .	VALIDATION_ERROR 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](#).

Buscar Nota por Código [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 el **detalle completo de una nota** de crédito o débito a partir de su identificador interno en nuby (`nota_id`): cabecera, ítems con sus impuestos y retenciones, y el estado y saldo actuales de la factura anotada.

Úsalo después de crear una nota, o cuando ya guardaste su `nota_id` y necesitas verificarla sin recorrer el listado.

¿Para qué sirve este servicio?
Úsalo cuando necesites **ver el detalle de una nota** sin recorrer el listado. Los casos de uso más frecuentes son:

- **Verificar una nota recién creada.** La creación ya devuelve el `nota_id`; con este servicio confirmas su contenido.
- **Mostrar el detalle del ajuste en tu plataforma:** cabecera e ítems con sus impuestos y retenciones.
- **Auditar los montos que calculó nuby** ítem por ítem: IVA, retenciones y descuento.
- **Confirmar el estado y el saldo de la factura anotada** antes de emitir el borrador en la interfaz de nuby.

1. El Endpoint (La dirección web)

Método y dirección

Dato	Valor
Método	GET
Recurso	/notes/{nota_id}
Cuerpo de la petición	No se envía cuerpo

```
GET https://{{instancia}}/service/v2/public/notes/{nota_id}
```

¿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/` + el identificador de la nota.

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?)

Este servicio **no requiere cuerpo**. Solo necesitas el encabezado de autorización y el identificador de la nota en la dirección.

Parámetros de la ruta (path)

Parámetro	Tipo	Requerido	Descripción
<code>nota_id</code>	entero	Sí	Identificador interno de la nota. Debe ser un número: la dirección solo acepta dígitos. Va en la dirección, después del segmento <code>/notes/</code> .

¿De dónde saco el `nota_id`?

- **De la creación de la nota** (`POST /notes`): la respuesta lo devuelve en `body.nota_id`.
- **Del listado** (`GET /notes/list`): cada fila del `body` lo trae en `nota_id`.

Ejemplos de petición

```
GET https://tu-inmobiliaria.nuby.app/service/v2/public/notes/789
```

3. La Respuesta (¿Qué te entregaremos?)

Si la nota existe y pertenece a una factura de venta, el sistema devuelve el envelope y el cuerpo con el detalle de la nota:

```
{
  "status": 200,
  "message": null,
  "body": {
    "nota_id": 789,
    "factura_id": 456,
    "tipo": "credito",
    "consecutivo": 1024,
    "fecha": "2026-10-21",
    "resolucion_id": 3,
    "valor_total": 59500,
    "estado_id": 5,
    "moneda": "COP",
    "tasa_cambio": 1,
    "observaciones": "Ajuste acordado",
    "clasificacion": 1,
    "tercero_id": 123,
    "concepto_id": 6,
    "items": [
      {
        "producto_id": 159,
        "descripcion": "Ajuste plan X",
        "cantidad": 1,
        "valor_unitario": 50000,
        "por_iva": 19,
        "iva": 9500,
        "por_retencion": 0,
```

```

    "retencion": 0,
    "por_reteiva": 0,
    "reteiva": 0,
    "por_reteica": 0,
    "reteica": 0,
    "por_descuento": 0,
    "descuento": 0
  }
],
"factura_estado_id": 1,
"factura_saldo": 150000
},
"pagination": null,
"filters_applied": null
}

```

Nota sobre el envelope: es una consulta de un solo registro, así que `pagination` y `filters_applied` siempre vienen en `null`.

Cabecera de la nota

Campo	Tipo	Puede ser <code>null</code>	Descripción
<code>nota_id</code>	entero	No	Identificador interno de la nota.
<code>factura_id</code>	entero	No	Factura anotada.
<code>tipo</code>	texto	No	<code>credito</code> o <code>debito</code> .
<code>consecutivo</code>	entero	No	Número de la nota.
<code>fecha</code>	fecha	No	Fecha de la nota en formato <code>AAAA-MM-DD</code> .
<code>resolucion_id</code>	entero	Sí	Resolución heredada de la factura; puede ser <code>null</code> si la factura no tiene resolución.
<code>valor_total</code>	número	No	Valor total de la nota, calculado desde el detalle.
<code>estado_id</code>	entero	No	<code>5</code> = borrador.
<code>moneda</code>	texto	No	Moneda heredada de la factura (por ejemplo, <code>COP</code>).

Campo	Tipo	Puede ser <code>null</code>	Descripción
<code>tasa_cambio</code>	número	No	Tasa heredada de la factura; es <code>1</code> cuando la moneda es la local.
<code>observaciones</code>	texto	No	Observaciones registradas; puede venir vacía si no se enviaron.
<code>clasificacion</code>	entero	No	<code>1</code> = vinculante.
<code>tercero_id</code>	entero	Sí	Tercero de la factura; puede ser <code>null</code> si la factura no lo tiene.
<code>concepto_id</code>	entero	No	Concepto aplicado a la nota.

Campos de cada ítem

Cada ítem trae la base y los porcentajes junto con los montos ya calculados por nuby:

Campo	Tipo	Descripción	Puede ser <code>null</code>
<code>producto_id</code>	entero	Producto o servicio del ítem.	No
<code>descripcion</code>	texto	Descripción del ajuste.	No (puede venir vacía)
<code>cantidad</code>	entero	Cantidad del ítem.	No
<code>valor_unitario</code>	número	Base unitaria enviada por el consumidor.	No
<code>por_iva</code>	número	Porcentaje de IVA aplicado.	No
<code>iva</code>	número	IVA calculado: $\text{base} \times \text{por_iva} / 100$.	No
<code>por_retencion</code>	número	Porcentaje de retención en la fuente.	No
<code>retencion</code>	número	Retención calculada sobre la base.	No
<code>por_reteiva</code>	número	Porcentaje de reteIVA.	No
<code>reteiva</code>	número	ReteIVA calculado sobre el IVA.	No
<code>por_reteica</code>	número	Porcentaje de reteICA.	No
<code>reteica</code>	número	ReteICA calculado sobre la base.	No
<code>por_descuento</code>	número	Porcentaje de descuento.	No

Campo	Tipo	Descripción	Puede ser <code>null</code>
<code>descuento</code>	número	Descuento calculado sobre la base.	No

Cómo se compone el valor total

Monto	Fórmula	Se calcula sobre
<code>iva</code>	<code>base × por_iva / 100</code>	La base del ítem (<code>cantidad × valor_unitario</code>).
<code>retencion</code>	<code>base × por_retencion / 100</code>	La base del ítem.
<code>reteiva</code>	<code>iva × por_reteiva / 100</code>	El IVA calculado.
<code>reteica</code>	<code>base × por_reteica / 100</code>	La base del ítem.
<code>descuento</code>	<code>base × por_descuento / 100</code>	La base del ítem.

El total del ítem suma la base y el IVA, y resta la retención, el reteIVA, el reteICA y el descuento. El `valor_total` de la nota es la suma de los totales de sus ítems, en la moneda original y con redondeo a dos decimales.

Estado y saldo de la factura

`factura_estado_id` indica el estado actual de la factura (`1` activa, `6` anulada por nota crédito) y `factura_saldo` su saldo después de las notas.

Variantes de respuesta

El mismo endpoint devuelve una nota de crédito, una de débito o un rechazo de factura de contrato. Los tres casos cambian el contenido del `body` y, en el último, el estado HTTP:

Nota de crédito

```
{
  "status": 200,
  "message": null,
  "body": {
    "nota_id": 789,
    "factura_id": 456,
    "tipo": "credito",
    "consecutivo": 1024,
    "fecha": "2026-10-21",
```

```
"resolucion_id": 3,
"valor_total": 59500,
"estado_id": 5,
"moneda": "COP",
"tasa_cambio": 1,
"observaciones": "Ajuste acordado",
"clasificacion": 1,
"tercero_id": 123,
"concepto_id": 6,
"items": [
  {
    "producto_id": 159,
    "descripcion": "Ajuste plan X",
    "cantidad": 1,
    "valor_unitario": 50000,
    "por_iva": 19,
    "iva": 9500,
    "por_retencion": 0,
    "retencion": 0,
    "por_reteiva": 0,
    "reteiva": 0,
    "por_reteica": 0,
    "reteica": 0,
    "por_descuento": 0,
    "descuento": 0
  }
],
"factura_estado_id": 1,
"factura_saldo": 150000
},
"pagination": null,
"filters_applied": null
}
```

La nota de crédito reduce el saldo de la factura. Si el saldo llega a cero, `factura_estado_id` pasa a `6`.

Nota de débito

```
{
  "status": 200,
  "message": null,
  "body": {
    "nota_id": 790,
    "factura_id": 456,
    "tipo": "debito",
    "consecutivo": 1025,
    "fecha": "2026-10-21",
    "resolucion_id": 3,
    "valor_total": 20000,
    "estado_id": 5,
    "moneda": "COP",
    "tasa_cambio": 1,
    "observaciones": "Cargo adicional acordado",
    "clasificacion": 1,
    "tercero_id": 123,
    "concepto_id": 8,
    "items": [
      {
        "producto_id": 159,
        "descripcion": "Cargo adicional octubre",
        "cantidad": 1,
        "valor_unitario": 20000,
        "por_iva": 0,
        "iva": 0,
        "por_retencion": 0,
        "retencion": 0,
        "por_reteiva": 0,
        "reteiva": 0,
        "por_reteica": 0,
        "reteica": 0,
        "por_descuento": 0,
        "descuento": 0
      }
    ],
    "factura_estado_id": 1,
    "factura_saldo": 170000
  },
}
```

```
"pagination": null,  
"filters_applied": null  
}
```

La nota de débito aumenta el saldo. La factura conserva el estado **1** mientras siga ACTIVA.

Nota de una factura de contrato (409)

Cuando la nota pertenece a una factura de contrato, nuby no devuelve sus datos: responde **409** con el código estable y la advertencia estructurada. El **scope** siempre es **solo_facturas_de_venta**.

```
{  
  "success": false,  
  "status": 409,  
  "message": "La factura 456 fue generada desde un contrato de arrendamiento. Esta API solo  
admite facturas de venta; las facturas de contrato se gestionan en el flujo de contratos de  
nuby.",  
  "error_code": "FACTURA_CONTRATO_NO_SOPORTADA",  
  "warnings": [  
    {  
      "code": "FACTURA_CONTRATO_NO_SOPORTADA",  
      "factura_id": 456,  
      "scope": "solo_facturas_de_venta"  
    }  
  ]  
}
```

Estados que puedes recibir

Campo	Valor	Significado
estado_id	5	Borrador. Es el estado de toda nota creada por esta API.
factura_estado_id	1	Factura ACTIVA; el saldo todavía es mayor que cero.
factura_estado_id	6	Factura anulada por nota crédito porque el saldo llegó a cero.

Campos que no expone el detalle

El detalle se limita a la nota y al estado y saldo de la factura. Estos datos no forman parte de la respuesta y debes consultarlos en otro servicio:

Dato	Por qué no aparece	Dónde consultarlo
Nombre y documento del tercero	Solo se devuelve <code>tercero_id</code> .	Buscar Tercero por Código.
Número, fecha y valor total de la factura	La nota solo informa <code>factura_estado_id</code> y <code>factura_saldo</code> .	Ver Detalle de una Factura.
Ítems de la factura original	El arreglo <code>items[]</code> corresponde a la nota, no a la factura.	Ver Detalle de una Factura.
Emisión DIAN de la nota (<code>cufe</code> , <code>dian_id</code>)	La API crea y consulta el borrador; la emisión es un paso posterior en nuby.	Interfaz de nuby.
Contactos, archivos y documentos del tercero	No forman parte de esta API.	Módulo de terceros de nuby.

Importante: los campos sin valor vienen en `null`, no se omiten. Verifica `null` antes de usar un dato en tu interfaz.

4. Seguridad y Posibles Errores

HTTP	<code>error_code</code>	Cuándo ocurre y cómo resolverlo
<code>401</code>	—	Falta el token o expiró. Vuelve a consumir Login .
<code>403</code>	<code>RECURSO_NO_AUTORIZADO</code>	Al cliente OAuth le falta <code>catalogs:read</code> .
<code>404</code>	<code>NOTA_NO_ENCONTRADA</code>	No existe una nota con ese identificador. Verifica el <code>nota_id</code> o búscala en Listar Notas .
<code>409</code>	<code>FACTURA_CONTRATO_NO_SOPORTADA</code>	La nota pertenece a una factura de contrato; esa información no se expone por esta API. Incluye <code>warnings[]</code> .
<code>500</code>	<code>INTERNAL_ERROR</code>	Fallo interno al consultar. Reintenta; si persiste, contacta a soporte.

Ejemplo de nota inexistente (404)

```
{
  "success": false,
  "status": 404,
```

```
"message": "No se encontró la nota con el código solicitado.",
"error_code": "NOTA_NO_ENCONTRADA"
}
```

Ojo con el identificador: `nota_id` es el número interno de la nota, no el consecutivo ni el número de la factura. Para buscar las notas de una factura usa `/notes/list?factura_id=...`.

Errores frecuentes y solución

Código	Causa	Qué hacer
401	No enviaste el token o ya expiró.	Vuelve a consumir Login y repite la consulta con el token nuevo.
403 RECURSO_NO_AUTORIZADO	Al cliente OAuth le falta el permiso <code>catalogs:read</code> .	Solicita el scope a tu administrador en nuby.
404 NOTA_NO_ENCONTRADA	El <code>nota_id</code> no existe en nuby.	Verifica el identificador o búscalo con Listar Notas .
409 FACTURA_CONTRATO_NO_SOPORTADA	La nota pertenece a una factura de contrato.	Gestiona el ajuste en el flujo de contratos y revisa <code>warnings[]</code> .
500 INTERNAL_ERROR	Fallo interno al consultar.	Reintenta; si persiste, contacta a soporte con el <code>nota_id</code> consultado.

5. Ejemplos de integración

cURL

```
TOKEN="TU_TOKEN_AQUI"
INSTANCIA="tu-inmobiliaria.nuby.app"
NOTA_ID=789

curl "https://${INSTANCIA}/service/v2/public/notes/${NOTA_ID}" \
-H "Authorization: Bearer ${TOKEN}"
```

PHP

```
<?php

$instancia = 'tu-inmobiliaria.nuby.app';
$token = 'TU_TOKEN_AQUI';
$notaId = 789;

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

$ch = curl_init($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);

if ($codigo === 200) {
    $nota = $datos['body'];
    echo "{$nota['tipo']} {$nota['consecutivo']} - total {$nota['valor_total']}\n";
    echo "Factura {$nota['factura_id']}: estado {$nota['factura_estado_id']}, saldo
    {$nota['factura_saldo']}\n";

    foreach ($nota['items'] as $item) {
        echo "{$item['descripcion']} - {$item['cantidad']} x {$item['valor_unitario']}\n";
    }
} else {
    echo "HTTP {$codigo}: " . $datos['message'] . "\n";
}
```

Python

```

import requests

instancia = "tu-inmobiliaria.nuby.app"
token = "TU_TOKEN_AQUI"
nota_id = 789
url = f"https://{instancia}/service/v2/public/notes/{nota_id}"

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

respuesta = requests.get(url, headers=headers, timeout=30)

if respuesta.status_code == 200:
    nota = respuesta.json()["body"]
    print(f"{nota['tipo']} {nota['consecutivo']} – total {nota['valor_total']}")
    print(f"Factura {nota['factura_id']}: estado {nota['factura_estado_id']}, saldo {nota['factura_saldo']}")

    for item in nota["items"]:
        print(item["descripcion"], item["cantidad"], item["valor_unitario"])
else:
    print(respuesta.status_code, respuesta.json())

```

JavaScript (Fetch API)

```

const instancia = 'tu-inmobiliaria.nuby.app';
const token = 'TU_TOKEN_AQUI';
const notaId = 789;

const url = `https://${instancia}/service/v2/public/notes/${notaId}`;

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

```

```
const data = await respuesta.json();

if (respuesta.ok) {
  const nota = data.body;
  console.log(`${nota.tipo} ${nota.consecutivo} – total ${nota.valor_total}`);
  console.log(`Factura ${nota.factura_id}: estado ${nota.factura_estado_id}, saldo
  ${nota.factura_saldo}`);
  nota.items.forEach(i => console.log(i.descripcion, i.cantidad, i.valor_unitario));
} else {
  console.error(`${respuesta.status} ${data.error_code}: ${data.message}`);
}
```

Power Query M (Excel / Power BI)

El detalle se modela como una tabla de cabecera y otra de ítems. Power Query consulta una vez y expande ambas partes del `body`:

```
let
  // 1. Configura tus datos de acceso
  instancia = "tu-inmobiliaria.nuby.app",
  token = "TU_TOKEN_AQUI",
  notaId = "789",

  // 2. Prepara la dirección de la petición
  url = "https://" & instancia & "/service/v2/public/notes/" & notaId,

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

  // 4. Decodifica la respuesta JSON
  jsonResponse = Json.Document(response),
```

```

body = jsonResponse[body],

// 5. Convierte la cabecera en tabla
cabecera = Table.FromList({body}, Splitter.SplitByNothing(), null, null,
ExtraValues.Error),
columnas = Table.ExpandRecordColumn(cabecera, "Column1",
{"nota_id", "factura_id", "tipo", "consecutivo", "fecha", "valor_total",
"estado_id", "concepto_id", "factura_estado_id", "factura_saldo"},
{"NotaID", "FacturaID", "Tipo", "Consecutivo", "Fecha", "ValorTotal",
"EstadoID", "ConceptoID", "FacturaEstadoID", "FacturaSaldo"}),
tipos = Table.TransformColumnTypes(columnas, {
{"NotaID", Int64.Type},
{"FacturaID", Int64.Type},
{"Consecutivo", Int64.Type},
{"Fecha", type date},
{"ValorTotal", type number},
{"FacturaSaldo", type number}
}),

// 6. Expande los ítems en una tabla aparte
items = Table.FromList(body[items], Splitter.SplitByNothing(), null, null,
ExtraValues.Error),
itemsExpandidos = Table.ExpandRecordColumn(items, "Column1",
{"producto_id", "descripcion", "cantidad", "valor_unitario", "por_iva", "iva",
"por_retencion", "retencion", "por_reteiva", "reteiva",
"por_reteica", "reteica", "por_descuento", "descuento"},
{"ProductoID", "Descripcion", "Cantidad", "ValorUnitario", "PorIVA", "IVA",
"PorRetencion", "Retencion", "PorReteIVA", "ReteIVA",
"PorReteICA", "ReteICA", "PorDescuento", "Descuento"})
in
tipos

```

6. Buenas prácticas y límites conocidos

Recomendaciones

- **Consulta el detalle solo cuando lo necesites.** La creación ya devuelve el `nota_id` y el saldo resultante.
- Usa `factura_saldo` y `factura_estado_id` para reflejar el estado real de la cartera sin consultar la factura.
- **Trata el `404` como un caso de negocio**, no como una falla técnica: el identificador no existe en nuby.
- **Recuerda que los campos ausentes vienen en `null`**; no supongas cadenas vacías.

Límites

- Solo devuelve notas de **facturas de venta**; las de contrato se rechazan con `409`.
- El detalle no incluye la emisión DIAN de la nota: la API solo crea y consulta el borrador.
- No incluye datos del tercero más allá de su identificador: para eso usa los servicios de terceros.

El detalle no admite paginación ni filtros: el `nota_id` identifica un único registro. Si necesitas recorrer varias notas, usa **Listar Notas** y consulta el detalle solo de las que vayas a mostrar. Así evitas una petición por cada nota del período.

Para crear una nota, revisa la página [Crear Nota](#). Para listar notas, revisa [Listar Notas](#).

Crear Nota [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 una **nota de crédito** o una **nota de débito** sobre una factura de venta de nuby. La nota se guarda en **borrador** y actualiza el saldo de la factura en el acto: una nota de crédito lo reduce y puede anular la factura si llega a cero; una nota de débito lo aumenta.

Es el servicio que necesitas cuando tu plataforma debe ajustar o anular una factura emitida desde nuby, sin volver a digitar el documento en la interfaz.

¿Para qué sirve este servicio?

Úsalo cuando necesites **ajustar una factura de venta** desde tu plataforma sin volver a digitar el documento en nuby. Los casos de uso más frecuentes son:

- **Anular una factura por nota de crédito.** Cuando el ajuste iguala el saldo, la factura queda anulada por nota crédito.
- **Registrar un cargo adicional posterior** con una nota de débito, por ejemplo un cobro que no estaba incluido en la factura original.
- **Replicar en nuby los ajustes que originas en tu plataforma**, con una `referencia_externa` que evita crear la nota dos veces.
- **Dejar la nota lista para emitir.** La API crea el borrador; un usuario lo emite a la DIAN desde nuby.

1. El Endpoint (La dirección web)

Método y dirección

Dato	Valor
Método	POST
Recurso	/notes

Dato	Valor
Cuerpo de la petición	Sí: el JSON de la nota que se describe en La Petición

POST https://{{instancia}}/service/v2/public/notes

¿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`.

Encabezados requeridos

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

Permiso requerido (scope): `notes:create`

El cliente OAuth debe tener habilitado `notes:create`. Si el token es válido pero 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. ¿Qué hace este servicio?

La nota se crea reutilizando la misma lógica de la interfaz de nuby: se asigna su consecutivo, se guarda su detalle y se recalcula el saldo de la factura. La API **no reimplementa** numeración, contabilidad ni DIAN.

Qué Sí hace

- Crea la nota en estado **borrador** y le asigna consecutivo.
- Reduce el saldo de la factura si es nota de crédito, o lo aumenta si es nota de débito.
- Deja la factura en estado **6 (anulada por nota crédito)** cuando el saldo queda en cero.

- Resuelve por ti el concepto y los datos que hereda de la factura.

Qué NO hace

- **No envía la nota a la DIAN.** La nota queda en borrador; la emisión la realiza un usuario en nuby.
- **No edita ni anula** notas existentes.
- **No crea notas no vinculantes.** Toda nota creada por esta API queda vinculada a su factura.
- **No anota facturas de contrato** (inquilino o propietario): esas se gestionan en el flujo de contratos.

Importante: la factura debe estar en estado **ACTIVA (facturada)**. Una factura en borrador o anulada no admite notas y la API responde `409 FACTURA_NO_ANOTABLE`.

3. La Petición (¿Qué debes enviarnos?)

Campos obligatorios

Campo	Tipo	Requerido	Descripción
<code>referencia_externa</code>	texto	Sí	Identificador único de tu operación. Es la llave de idempotencia: un reintento no crea una segunda nota.
<code>factura_id</code>	entero	Sí	Identificador interno de la factura de venta que se va a anotar.
<code>tipo</code>	texto	Sí	<code>credito</code> para reducir el saldo o <code>debito</code> para aumentarlo.
<code>items[]</code>	arreglo	Sí	Al menos un ítem con <code>producto_id</code> , <code>cantidad</code> entero mayor que cero y <code>valor_unitario</code> distinto de cero.

`fecha` es opcional: si no la envías, se usa la fecha de hoy en formato `AAAA-MM-DD`.

Campos opcionales

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

Cabecera	concepto_id, observaciones, por_iva, por_retencion, fecha_ini, fecha_fin
Ítem	descripcion, por_iva, por_retencion, por_reteiva, por_reteica, por_descuento

Datos que hereda de la factura

No necesitas enviarlos y, de hecho, no se aceptan si difieren: `resolucion_id`, `tercero_id`, `moneda` y `tasa_cambio` se toman de la factura. Si envías una moneda o una tasa distinta, la respuesta es `422 CAMPO_NO_PERMITIDO`: el saldo se calcula sin conversión y un valor distinto lo descuadraría.

Campos que el consumidor no puede fijar

La nota hereda de la factura los datos que garantizan la coherencia del saldo y de la numeración. Si envías uno de estos campos con valor, la respuesta es `422 CAMPO_NO_PERMITIDO` y la nota no se crea:

Campo	Por qué no lo puedes fijar	Respuesta
<code>clasificacion</code>	El servidor fija <code>1</code> (vinculante). La API no crea notas no vinculantes.	<code>422 CAMPO_NO_PERMITIDO</code>
<code>tercero_id</code>	Se hereda de la factura anotada. Un tercero distinto descuadraría la cartera.	<code>422 CAMPO_NO_PERMITIDO</code>
<code>resolucion_id</code>	Se hereda de la factura; la nota no tiene resolución propia.	<code>422 CAMPO_NO_PERMITIDO</code>
<code>tipo_factura</code>	Lo determina la factura anotada, que debe ser de venta.	<code>422 CAMPO_NO_PERMITIDO</code>
<code>consecutivo</code>	Lo asigna nuby al crear la nota.	<code>422 CAMPO_NO_PERMITIDO</code>
<code>estado_id</code>	La nota nace en <code>5</code> (borrador); la emisión es un paso posterior en nuby.	<code>422 CAMPO_NO_PERMITIDO</code>
<code>dian_id</code> , <code>cufe</code>	La emisión a la DIAN no forma parte de esta API.	<code>422 CAMPO_NO_PERMITIDO</code>
<code>valor_total</code>	Se recalcula desde los ítems y sus porcentajes.	<code>422 CAMPO_NO_PERMITIDO</code>
<code>creadopor</code>	Lo registra nuby con el consumidor autenticado.	<code>422 CAMPO_NO_PERMITIDO</code>
<code>nota_id</code>	Lo asigna nuby; enviarlo sugiere una edición que la API no soporta.	<code>422 CAMPO_NO_PERMITIDO</code>

Campo	Por qué no lo puedes fijar	Respuesta
moneda, tasa_cambio	Se heredan de la factura. Solo se toleran si coinciden exactamente con ella.	422 CAMPO_NO_PERMITIDO si difieren
Montos calculados: iva, retencion, reteiva, reteica, descuento	El servidor los deriva de valor_unitario y de los porcentajes del ítem.	Se ignoran; el total se recalcula

Concepto de la nota

Si omites concepto_id, nuby aplica el concepto por defecto del tipo. El concepto se resuelve contra los target y codigo de la maestra:

Tipo de nota	target	codigo	Cuándo aplica
Nota de crédito	1	'2'	Se omite concepto_id y tipo es credito. Corresponde a la anulación de la factura electrónica.
Nota de débito	2	'4'	Se omite concepto_id y tipo es debito. Corresponde al concepto "Otros".

Si envías un concepto_id, nuby no aplica el default: valida que el concepto exista y que su target corresponda al tipo de nota. Si no existe o no corresponde, la respuesta es 422 CONCEPTO_NO_DISPONIBLE y no se crea la nota.

Montos: tú envías tasas, nuby calcula

Por cada ítem envía valor_unitario y los porcentajes. El servidor calcula el IVA, las retenciones y el descuento, y recalcula el total de la nota a partir del detalle. Si envías montos ya calculados, se ignoran.

Monto	Fórmula (porcentaje sobre 100)	Ejemplo con base 50000
iva	base × por_iva	50000 × 19 % = 9500
retencion	base × por_retencion	50000 × 0 % = 0
reteiva	iva × por_reteiva	9500 × 0 % = 0
reteica	base × por_reteica	50000 × 0 % = 0
descuento	base × por_descuento	50000 × 0 % = 0

El total del ítem se obtiene al sumar la base y el IVA, y restar la retención, el reteIVA, el reteICA y el descuento. Con el ejemplo anterior, el total de la nota es 59500. Todos

los montos se calculan en la moneda original de la factura y se redondean a dos decimales.

Importante: el IVA de un ítem requiere que el producto tenga configurado su tipo de IVA en nuby. Si no lo tiene, la operación falla y no se crea la nota.

Ejemplo: nota de crédito total

Cuando el total de la nota de crédito iguala el saldo de la factura, el saldo queda en `0` y la factura pasa al estado `6` (anulada por nota crédito). La respuesta lo confirma en `factura_estado_id` y `factura_saldo`.

```
{
  "referencia_externa": "NOTE-000046",
  "factura_id": 456,
  "tipo": "credito",
  "fecha": "2026-10-21",
  "observaciones": "Anulación total acordada",
  "items": [
    {
      "producto_id": 159,
      "descripcion": "Anulación total factura 456",
      "cantidad": 1,
      "valor_unitario": 100000,
      "por_iva": 19,
      "por_retencion": 0,
      "por_reteiva": 0,
      "por_reteica": 0,
      "por_descuento": 0
    }
  ]
}
```

La nota calcula $\text{iva} = 100000 \times 19 \% = 19000$ y un total de `119000`. Si el saldo de la factura es exactamente `119000`, la respuesta devuelve `factura_saldo: 0` y `factura_estado_id: 6`.

Ejemplo: nota de débito

La nota de débito aumenta el saldo. Úsala para cobrar un ajuste posterior, por ejemplo un cargo no incluido en la factura original. La factura debe seguir ACTIVA.

```
{
  "referencia_externa": "NOTE-000047",
  "factura_id": 456,
  "tipo": "debito",
  "fecha": "2026-10-21",
  "observaciones": "Cargo adicional acordado",
  "items": [
    {
      "producto_id": 159,
      "descripcion": "Cargo adicional octubre",
      "cantidad": 1,
      "valor_unitario": 20000,
      "por_iva": 0,
      "por_retencion": 0,
      "por_reteiva": 0,
      "por_reteica": 0,
      "por_descuento": 0
    }
  ]
}
```

Si el saldo era 119000 , después de esta nota de débito será 139000 . La factura conserva el estado 1 (ACTIVA).

Ejemplo completo

```
{
  "referencia_externa": "NOTE-000045",
  "factura_id": 456,
  "tipo": "credito",
  "concepto_id": null,
  "fecha": "2026-10-21",
  "observaciones": "Ajuste acordado",
  "items": [
    {
      "producto_id": 159,
```

```

    "descripcion": "Ajuste plan X",
    "cantidad": 1,
    "valor_unitario": 50000,
    "por_iva": 19,
    "por_retencion": 0,
    "por_reteiva": 0,
    "por_reteica": 0,
    "por_descuento": 0
  }
]
}

```

4. Cómo afecta el saldo de la factura

Escenario ? resultado en la factura

Escenario	Resultado en la factura
Nota de crédito parcial	El saldo se reduce en el valor de la nota. La factura sigue ACTIVA (1).
Nota de crédito total	El saldo llega a 0 y la factura queda en estado 6 (anulada por nota crédito).
Nota de débito	El saldo aumenta en el valor de la nota. La factura sigue ACTIVA (1).
Factura en borrador	409 FACTURA_NO_ANOTABLE . No se crea la nota ni se modifica el saldo.
Factura de contrato	409 FACTURA_CONTRATO_NO_SOPORTADA con warnings[] . No se crea la nota ni se modifica el saldo.
Factura inexistente	404 FACTURA_NO_ENCONTRADA . Verifica el factura_id antes de reintentar.

La nota se crea **en borrador** y aun así afecta el saldo, porque nubly recalcula la cartera al guardarla. La emisión a la DIAN es un paso posterior y no cambia el saldo.

Ojo con el estado de la factura: solo se anotan facturas **ACTIVAS**. Si la factura ya fue anulada o está en borrador, la respuesta es **409 FACTURA_NO_ANOTABLE**.

5. La Respuesta (¿Qué te entregaremos?)

Si la nota se crea correctamente, la respuesta es un JSON con el resultado de la operación y el estado en que queda la factura anotada:

```
{
  "success": true,
  "status": 201,
  "message": "Nota creada en borrador.",
  "body": {
    "nota_id": 789,
    "tipo": "credito",
    "consecutivo": 1024,
    "estado_id": 5,
    "factura_id": 456,
    "factura_estado_id": 1,
    "factura_saldo": 150000
  },
  "idempotent_replay": false
}
```

Estructura de la respuesta

Campo	Tipo	Descripción
nota_id	entero	Identificador interno de la nota. Guárdalo para consultarla después.
tipo	texto	credito o debito.
consecutivo	entero	Número asignado a la nota.
estado_id	entero	5 = borrador.
factura_estado_id	entero	Estado de la factura después de la nota (1 activa, 6 anulada por nota crédito).
factura_saldo	número	Saldo de la factura después de la nota.
idempotent_replay	booleano	false en la primera respuesta; true cuando se recupera una solicitud anterior.

Escenario de reintento (replay)

Si repites la petición con la misma `referencia_externa` y el mismo contenido, nuby no crea una segunda nota: devuelve la respuesta original con `idempotent_replay` en `true`.

```
{
  "success": true,
  "status": 201,
  "message": "Nota creada en borrador.",
  "body": {
    "nota_id": 789,
    "tipo": "credito",
    "consecutivo": 1024,
    "estado_id": 5,
    "factura_id": 456,
    "factura_estado_id": 1,
    "factura_saldo": 150000
  },
  "idempotent_replay": true
}
```

Para ver el detalle completo de la nota, consulta la página [Buscar Nota por Código](#).

6. Idempotencia y reintentos

La escritura exige `referencia_externa`. nuby recuerda qué respondió para esa referencia, así que un reintento no duplica la nota.

Situación	Qué responde nuby
Primera vez que se usa la referencia	Procesa y responde <code>201</code> .
Misma referencia y mismo contenido	La respuesta original con <code>idempotent_replay: true</code> .
Misma referencia con otro contenido	<code>409 IDEMPOTENCY_PAYLOAD_MISMATCH</code> .
Solicitud en proceso	<code>409 IDEMPOTENCY_IN_PROGRESS</code> .
Solicitud fallida	Se permite reintentar, incluso con datos corregidos.

Importante: un error `422` de validación **no consume** la referencia. Puedes corregir el payload y reintentar con la misma `referencia_externa`.

Token renovado: el contenido interno que agrega la autenticación no forma parte de la comparación de idempotencia. Puedes renovar el token y reintentar sin conflicto.

7. Seguridad y Posibles Errores

HTTP	error_code	Cuándo ocurre y cómo resolverlo
400	VALIDATION_ERROR	El cuerpo no es un objeto JSON. Envía JSON válido con <code>Content-Type: application/json</code> .
401	—	Falta el token o expiró. Vuelve a consumir Login .
403	RECURSO_NO_AUTORIZADO	Al cliente OAuth le falta <code>notes:create</code> .
404	FACTURA_NO_ENCONTRADA	El <code>factura_id</code> no existe en nuby.
409	FACTURA_CONTRATO_NO_SOPORTADA	La factura es de contrato; se gestiona en el flujo de contratos. Incluye <code>warnings[]</code> .
409	FACTURA_NO_ANOTABLE	La factura no está ACTIVA (borrador, anulada o pagada).
409	IDEMPOTENCY_IN_PROGRESS / IDEMPOTENCY_PAYLOAD_MISMATCH	Reintento con la misma referencia en curso o con contenido distinto.
422	VALIDATION_ERROR	Falta un obligatorio o un dato no es válido. Revisa <code>errors</code> por campo.
422	CAMPO_NO_PERMITIDO	Intentaste fijar un campo del sistema o una moneda/tasa distinta a la factura.
422	FACTURACION_ELECTRONICA_DESHABILITADA	La facturación electrónica está apagada en nuby. No se puede crear la nota por API.
422	CONCEPTO_NO_DISPONIBLE	El <code>concepto_id</code> no existe o no corresponde al tipo de nota.
422	DATOS_NOTA_INVALIDOS	Los datos no pasaron la validación de nuby. Revisa el mensaje.
500	INTERNAL_ERROR	Fallo interno. La solicitud queda fallida y es reintentable; si persiste, contacta a soporte.

Errores de validación frecuentes

Situación	Mensaje / error_code	Cómo corregir
-----------	----------------------	---------------

Falta <code>referencia_externa</code> o viene vacía	422 VALIDATION_ERROR	Envía una referencia única y no vacía. Sin ella no se aplica idempotencia.
Falta <code>factura_id</code> , <code>tipo</code> o <code>items[]</code>	422 VALIDATION_ERROR	Incluye la factura, un tipo válido (<code>credito</code> o <code>debito</code>) y al menos un ítem.
Un ítem no tiene <code>producto_id</code> , o su <code>cantidad</code> o <code>valor_unitario</code> no es válido	422 VALIDATION_ERROR	Usa <code>cantidad</code> entero mayor que cero y <code>valor_unitario</code> distinto de cero.
La factura no existe	404 FACTURA_NO_ENCONTRADA	Verifica el <code>factura_id</code> con el detalle o el listado de facturas de venta.
La factura es de contrato	409 FACTURA_CONTRATO_NO_SOPORTADA	Anula o ajusta desde el flujo de contratos. Revisa <code>warnings[].code</code> , <code>factura_id</code> y <code>scope</code> .
La factura no está ACTIVA	409 FACTURA_NO_ANOTABLE	Solo se anotan facturas en estado <code>1</code> . Emite o descarta el borrador antes de anotar.
La facturación electrónica está apagada	422 FACTURACION_ELECTRONICA_DESHABILITADA	Pide a tu administrador que habilite <code>ENABLE_FACT_ELECT = 1</code> en nuby.
El concepto no existe o no corresponde al tipo	422 CONCEPTO_NO_DISPONIBLE	Omite <code>concepto_id</code> para usar el default, o envía uno del <code>target</code> correcto.
Enviaste un campo reservado al sistema	422 CAMPO_NO_PERMITIDO	Quita <code>clasificacion</code> , <code>tercero_id</code> , <code>resolucion_id</code> , <code>consecutivo</code> , <code>estado_id</code> , <code>valor_total</code> , <code>nota_id</code> y los demás de la lista.
La moneda o la tasa no coincide con la factura	422 CAMPO_NO_PERMITIDO	Omite <code>moneda</code> y <code>tasa_cambio</code> , o envía exactamente los de la factura.
El producto no tiene configurado su tipo de IVA o los porcentajes no son válidos	422 DATOS_NOTA_INVALIDOS	Revisa la configuración del producto en nuby y envía porcentajes válidos por ítem.

Atención: cuando la respuesta es `409` o `422`, nuby no deja una nota a medias. La operación se revierte completa.

8. Ejemplos de integración

cURL

```
TOKEN="TU_TOKEN_AQUI"
```

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

```
curl -X POST "https://${INSTANCIA}/service/v2/public/notes" \  
  -H "Content-Type: application/json" \  
  -H "Authorization: Bearer ${TOKEN}" \  
  -d '{  
    "referencia_externa": "NOTE-000045",  
    "factura_id": 456,  
    "tipo": "credito",  
    "fecha": "2026-10-21",  
    "observaciones": "Ajuste acordado",  
    "items": [  
      {  
        "producto_id": 159,  
        "descripcion": "Ajuste plan X",  
        "cantidad": 1,  
        "valor_unitario": 50000,  
        "por_iva": 19,  
        "por_retencion": 0,  
        "por_reteiva": 0,  
        "por_reteica": 0,  
        "por_descuento": 0  
      }  
    ]  
  }'
```

PHP

```
<?php
```

```
$instancia = 'tu-inmobiliaria.nuby.app';
```

```
$token = 'TU_TOKEN_AQUI';
```

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

```
$payload = [
    'referencia_externa' => 'NOTE-000045',
    'factura_id' => 456,
    'tipo' => 'credito',
    'fecha' => '2026-10-21',
    'observaciones' => 'Ajuste acordado',
    'items' => [
        [
            'producto_id' => 159,
            'descripcion' => 'Ajuste plan X',
            'cantidad' => 1,
            'valor_unitario' => 50000,
            'por_iva' => 19,
            'por_retencion' => 0,
            'por_reteiva' => 0,
            'por_reteica' => 0,
            'por_descuento' => 0,
        ],
    ],
];

$ch = curl_init($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/notes"

payload = {
    "referencia_externa": "NOTE-000045",
    "factura_id": 456,
    "tipo": "credito",
    "fecha": "2026-10-21",
    "observaciones": "Ajuste acordado",
    "items": [
        {
            "producto_id": 159,
            "descripcion": "Ajuste plan X",
            "cantidad": 1,
            "valor_unitario": 50000,
            "por_iva": 19,
            "por_retencion": 0,
            "por_reteiva": 0,
            "por_reteica": 0,
            "por_descuento": 0,
        }
    ],
}

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

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

if respuesta.status_code == 201:
```

```
    datos = respuesta.json()

    print(f"Nota {datos['body']['nota_id']} – saldo factura:
{datos['body']['factura_saldo']}")
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/notes`;

const payload = {
  referencia_externa: 'NOTE-000045',
  factura_id: 456,
  tipo: 'credito',
  fecha: '2026-10-21',
  items: [{
    producto_id: 159,
    descripcion: 'Ajuste plan X',
    cantidad: 1,
    valor_unitario: 50000,
    por_iva: 19,
    por_retencion: 0,
    por_reteiva: 0,
    por_reteica: 0,
    por_descuento: 0
  }]
};

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(`Nota ${data.body.nota_id} – saldo factura: ${data.body.factura_saldo}`);
} else {
  console.error(`${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`. La respuesta se convierte en una tabla de una fila. Si la petición falla, `Web.Contents` genera una excepción que puedes capturar con `try ... otherwise`.

```

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/notes",

    // 3. Construye el cuerpo JSON
    cuerpo = Json.FromValue([
        referencia_externa = "NOTE-000045",
        factura_id = 456,
        tipo = "credito",
        fecha = "2026-10-21",
        observaciones = "Ajuste acordado",
        items = {
            [
                producto_id = 159,

```

```

        descripcion = "Ajuste plan X",
        cantidad = 1,
        valor_unitario = 50000,
        por_iva = 19,
        por_retencion = 0,
        por_reteiva = 0,
        por_reteica = 0,
        por_descuento = 0
    ]
}
]),

```

// 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",
    {"nota_id", "tipo", "consecutivo", "estado_id", "factura_id", "factura_estado_id",
"factura_saldo"},
    {"NotaID", "Tipo", "Consecutivo", "EstadoID", "FacturaID", "FacturaEstadoID",
"FacturaSaldo"})
in
    expandido

```

9. Buenas prácticas y límites conocidos

Recomendaciones

- **Verifica el estado de la factura antes de anotar:** debe estar ACTIVA.
- **Usa una referencia externa por operación** y no la reutilices para otra nota.
- **Guarda el** `nota_id` **y el** `factura_saldo` **de la respuesta:** te ahorran una consulta posterior.
- **Envía los porcentajes por ítem:** los de cabecera no se usan.
- **Trata el** `422` **como reintentable tras corregir:** la referencia no se consume.

Límites

- La nota queda en borrador: la emisión a la DIAN no forma parte de esta API.
- No hay edición ni anulación de notas por API.
- No se admiten notas no vinculantes.
- `fecha_ini` y `fecha_fin` se aceptan pero no se persisten en la nota.
- No se valida el rango 0-100 de los porcentajes; envía valores válidos.

Para consultar notas ya registradas, revisa las páginas [Listar Notas](#) y [Buscar Nota por Código](#).