Ver Detalle de una Factura

Este servicio devuelve el **detalle completo de una factura de venta** a partir de su identificador interno en nuby (`factura_id`): estado del documento, consecutivo, valor total, saldo, estado ante la DIAN, CUFE, enlace seguro del PDF y el detalle de ítems con sus impuestos y descuentos.

Es la consulta de **un solo documento**: no pagina y no filtra. Si todavía no tienes el `factura_id`, el listado de facturas de venta te lo entrega.

> **¿Para qué sirve este servicio?**
> Úsalo siempre que necesites **todos los datos de una factura concreta** sin recorrer el listado. Los casos de uso más frecuentes son:

- **Mostrar la factura en tu portal o aplicación.** Reúne cabecera, ítems, impuestos y el enlace del PDF en una sola llamada.
- **Confirmar el estado antes de operar.** `estado_id` indica si el documento sigue en borrador, si ya se facturó, si se pagó o si se anuló.
- **Conciliar la factura electrónica.** `cufe`, `estado_dian` y `fecha_envio_dian` del listado te permiten rastrear el documento ante la DIAN.
- **Validar los impuestos de una línea.** Cada ítem trae IVA, retención, reteiva y reteica en valor y en porcentaje.
- **Enviar el comprobante al cliente.** `pdf_link` llega listo para abrir en el navegador.
- **Auditar un saldo.** Compara `valor_total` con `valor_saldo` para saber cuánto falta por cobrar.

---

1. El Endpoint (La dirección web)

Método y dirección

Dato	Valor
Método	`GET`
Recurso	`/invoices/{factura_id}`
Cuerpo de la petición	No se envía cuerpo

```
GET https://{{instancia}}/service/v2/public/invoices/{factura_id}
```

¿Qué debo colocar en `{{instancia}}`?

Es la **dirección web principal** con la que ingresas a tu plataforma, incluyendo la terminación `.nuby.app` o `.arrendasoft.co`. Si entras a nuby escribiendo `inmobiliaria.nuby.app` en el navegador, esa es tu instancia.

> **Importante:** no incluyas `https://` ni barras diagonales (`/`) al final. La dirección se compone así: `https://` + tu instancia + `/service/v2/public/invoices/` + el identificador de la factura.

Encabezados requeridos

Encabezado	¿Obligatorio?	Valor
`Authorization`	Sí	`Bearer <token>`, con el token obtenido en el servicio **Login**.
`Content-Type`	No	`application/json` si tu cliente lo envía. Como la petición no lleva cuerpo, el servicio no lo necesita.

Permiso requerido (scope): `catalogs:read`

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

> **¿Por qué `catalogs:read` y no un permiso de facturas?** El libro publicado de la API documenta el permiso genérico `read` para los endpoints de consulta anteriores (los que ya existían). Los endpoints nuevos del ciclo administrativo usan el permiso por recurso, y en tu cliente OAuth los permisos de consulta disponibles son `catalogs:read`, `invoices:create`, `invoices:update`, `notes:create` y `third-parties:create`. Como no existe un permiso de lectura propio de facturas, leer facturas se autoriza con `catalogs:read`: es el permiso de consulta con el que ya cuentas.

> Si aún no tienes un token, consulta la página *Login* del libro **API nuby V2**.

Requisitos previos de esta operación

- La factura debe **existir** en nuby: si no, la respuesta es `404 FACTURA_NO_ENCONTRADA`.
- La factura debe ser de **venta**. Si se generó desde un contrato de arrendamiento, se rechaza con `409 FACTURA_CONTRATO_NO_SOPORTADA` y `warnings[]`.
- El cliente OAuth del token debe tener el permiso `catalogs:read`.

2. La Petición (¿Qué debes enviarnos?)

Este servicio **no requiere cuerpo**. Todo lo que necesitas enviar es el encabezado de autorización y el identificador de la factura en la dirección.

Dato	Valor
Método	`GET`
`Content-Type`	`application/json` (opcional: la petición no tiene cuerpo)
`Authorization`	`Bearer <token>`, token obtenido al consumir el servicio **Login**

Parámetros de la ruta (path)

Parámetro	Tipo	Requerido	Por defecto	Descripción
`factura_id`	integer	Sí	—	Identificador interno de la factura. Debe ser un entero **mayor que 0**. Va en la dirección, no en el cuerpo ni en la query string.

Si el identificador no es un entero positivo (por ejemplo `/invoices/abc` o `/invoices/0`), la respuesta es `400` con el mensaje "*Debe indicar un factura_id entero mayor que 0 para operar sobre la factura.*" y el detalle del campo en `errors.factura_id`:

```
{
  "success": false,
  "status": 400,
  "message": "Debe indicar un factura_id entero mayor que 0 para operar sobre la factura.",
  "error_code": "VALIDATION_ERROR",
  "errors": {
    "factura_id": [
      "Debe indicar un factura_id entero mayor que 0 para operar sobre la factura."
    ]
  }
```

```
}
```

¿De dónde saco el `factura_id`?

- **De la creación de la factura** (`POST /invoices`): la respuesta lo devuelve en `body.factura_id`.
- **Del listado** (`GET /invoices/sales/list`): cada fila del `body` lo trae en `factura_id`.

> **No es el número de factura.** `factura_id` es el identificador **interno** de nuby; no es el consecutivo ni el número que aparece impreso. Si envías el número de factura en la dirección (por ejemplo `/invoices/1534`), no obtendrás la factura que buscas salvo que coincida con el identificador interno. Para encontrar el `factura_id` correcto, usa el **listado de facturas de venta**.

Ejemplos de petición

```
GET https://mi-inmobiliaria.nuby.app/service/v2/public/invoices/10231
GET https://mi-inmobiliaria.nuby.app/service/v2/public/invoices/10231?page=1
GET https://mi-inmobiliaria.nuby.app/service/v2/public/invoices/10230
```

> **Sobre el segundo ejemplo:** el endpoint no lee parámetros de consulta, así que `?page=1` se ignora. Úsalo solo si tu cliente los agrega de forma automática.

Encabezados de una petición completa

```
GET /service/v2/public/invoices/10231 HTTP/1.1
Host: mi-inmobiliaria.nuby.app
Content-Type: application/json
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...
```

> **Orden de respuesta:** nuby primero revisa si la factura es de contrato (`409`), después si existe (`404`) y solo entonces devuelve los datos (`200`). Por eso una factura de contrato nunca se expone por esta vía.

3. La Respuesta (¿Qué te entregaremos?)

Si el token es válido y la factura existe, el sistema devuelve un JSON con el estado de la respuesta y todos los datos del documento en `body`. La respuesta se verá similar a

esta:


```
{
  "status": 200,
  "message": "",
  "body": {
    "factura_id": 10231,
    "estado_id": 1,
    "consecutivo": 1534,
    "prefijo": "FE",
    "valor_total": 1190000,
    "valor_saldo": 1190000,
    "estado_dian": "Aprobado",
    "cufe": "b3f1c0d2e5a94f7c8d10b6a2e4f70c93",
    "pdf_link": "https://<dominio-publico-pdf>?hash=...",
    "fecha": "2026-10-21",
    "fecha_vencimiento": "2026-11-20",
    "tercero_id": 4821,
    "items": [
      {
        "detalle_id": 55231,
        "producto_id": 340,
        "producto": "ADMINISTRACION",
        "descripcion": "Administración octubre 2026",
        "cantidad": 1,
        "valor_unitario": 1000000,
        "descuento": 0,
        "por_descuento": 0,
        "iva": 190000,
        "por_iva": 19,
        "retencion": 0,
        "por_retencion": 0,
        "reteiva": 0,
        "por_reteiva": 0,
        "reteica": 0,
        "por_reteica": 0
      }
    ]
  },
  "pagination": [],
```

```
  "filters_applied": []
}
```

Estructura principal de la respuesta

Campo	Tipo	Descripción
`status`	integer	Código del estado de la respuesta. `200` cuando la consulta fue exitosa.
`message`	string	Cadena vacía cuando la consulta fue exitosa.
`body`	object	Datos de la factura. Ver la tabla de campos siguiente. En un error, este campo no existe.
`pagination`	array	Siempre una lista vacía `[]`: esta consulta no pagina.
`filters_applied`	array	Siempre una lista vacía `[]`: esta consulta no tiene filtros.

> **Nota sobre el envelope:** es una consulta de un solo registro, así que `pagination` y `filters_applied` siempre vienen como listas vacías. En una respuesta exitosa **no aparece** la clave `warnings`.

Campos de la factura (`body`)

Campo	Tipo	Descripción
`factura_id`	integer	Identificador interno de la factura consultada.
`estado_id`	integer	Código del estado del documento: `1` Facturada, `2` Anulada, `3` Pagada, `5` Borrador, `6` Anulada por NC.
`consecutivo`	integer	Número consecutivo de la factura. En un borrador es **cero o negativo**: el consecutivo real se asigna al emitir. En el listado, este mismo dato se llama `factura_numero`.
`valor_total`	decimal	Valor total del documento.
`valor_saldo`	decimal	Saldo pendiente del documento. Vale `0` cuando la factura está pagada. En el listado, este mismo dato se llama `saldo`.

Campo	Tipo	Descripción
`estado_dian`	string | null	Estado del documento ante la DIAN. Es `null` mientras la factura no se haya enviado.
`cufe`	string | null	Código único de factura electrónica. Es `null` mientras la factura no se haya enviado. **No lo devuelve el listado**: solo este detalle.
`pdf_link`	string	Enlace seguro de impresión del PDF, con una vigencia de **30 días**. Es el mismo enlace que devuelve el servicio de PDF.
`fecha`	string	Fecha de la factura (`AAAA-MM-DD`). En el listado, este mismo dato se llama `fecha_factura`.
`fecha_vencimiento`	string	Fecha de vencimiento del documento (`AAAA-MM-DD`).
`tercero_id`	integer	Identificador interno del tercero facturado. Para sus datos de identificación y contacto, consulta el servicio de terceros.
`items`	array	Detalle de ítems de la factura. Arreglo vacío `[]` si la factura no tiene líneas. Ver la tabla siguiente.

> **Ojo con los nombres.** En el detalle la fila trae `valor_saldo`, `consecutivo` y `fecha`; en el listado los mismos datos se llaman `saldo`, `factura_numero` y `fecha_factura`. Es intencional: cada servicio conserva su contrato.

Campos de cada ítem (`items[]`)

Campo	Tipo	Descripción
`detalle_id`	integer	Identificador interno de la línea.
`producto_id`	integer	Identificador interno del producto o servicio facturado.
`producto`	string	Nombre del producto o servicio tal como está registrado en nuby.
`descripcion`	string	Descripción de la línea facturada.
`cantidad`	integer	Cantidad facturada en la línea.
`valor_unitario`	decimal	Valor unitario de la línea, antes de impuestos y descuentos.

Campo	Tipo	Descripción
`descuento`	decimal	Descuento de la línea, en valor.
`por_descuento`	decimal	Porcentaje de descuento aplicado a la línea.
`iva`	decimal	IVA de la línea, en valor.
`por_iva`	decimal	Porcentaje de IVA aplicado a la línea.
`retencion`	decimal	Retención en la fuente de la línea, en valor.
`por_retencion`	decimal	Porcentaje de retención en la fuente aplicado a la línea.
`reteiva`	decimal	Retención de IVA de la línea, en valor.
`por_reteiva`	decimal	Porcentaje de retención de IVA aplicado a la línea.
`reteica`	decimal	Retención de ICA de la línea, en valor.
`por_reteica`	decimal	Porcentaje de retención de ICA aplicado a la línea.

`cufe` y `estado_dian` en `null`

Los dos campos son `null` mientras la factura **no se haya enviado** a la DIAN. `null` distingue "sin CUFE" de una cadena vacía: no asumas que el documento ya está emitido por ver la clave presente en la respuesta.

> **Un borrador también se consulta.** Si la factura sigue en `estado_id = 5`, el detalle devuelve sus datos con `consecutivo` en cero o negativo, el `cufe` y el `estado_dian` en `null`. Es la forma de revisar el borrador antes de emitirlo.

> **Los importes del detalle llegan como número.** A diferencia del listado, aquí `valor_total`, `valor_saldo` y los valores de cada ítem se entregan convertidos a número decimal. Aun así, valida el tipo en tu cliente antes de operar con ellos.

4. Seguridad y Posibles Errores

Tu pase tiene fecha de caducidad

Por medidas de seguridad, el token que te entregamos **solo dura 1 hora**. Una vez transcurrido ese tiempo, el sistema te bloquea el acceso con un error `401` y tu sistema

debe volver a consumir el servicio de **Login** para pedir un pase nuevo.

Esta respuesta la produce el **middleware de autenticación heredado** (el mismo de toda la API) y no el contrato de la operación. Así se ve:

```
{
  "statusCode": 401,
  "error": {
    "type": "SERVER_ERROR",
    "description": "JWT Token expired."
  }
}
```

> **Ojo con la forma:** los errores del middleware usan `statusCode` y `error.description`, no `success` ni `error_code`. El valor de `type` depende del manejador de errores de la versión desplegada (el libro publicado muestra `SERVER_ERROR` y el manejador actual responde `UNAUTHENTICATED`); lo estable es `statusCode` y la `description`.

Situación	HTTP	`error.description` exacta
No se envió el encabezado `Authorization`	`401`	`JWT Token required.`
El encabezado no tiene el formato `Bearer {token}`	`400`	`JWT Token not send.`
El token no fue encontrado en el sistema	`401`	`JWT Token not found.`
El token expiró	`401`	`JWT Token expired.`

Errores del contrato de la operación

Los errores que produce la operación misma traen `success: false`, el código HTTP, un `message` legible y un `error_code` estable:

```
{
  "success": false,
  "status": 404,
  "message": "La factura 990 no existe.",
  "error_code": "FACTURA_NO_ENCONTRADA"
}
```

Código HTTP	`error_code`	Cuándo ocurre y cómo resolverlo
`400`	`VALIDATION_ERROR`	El `factura_id` de la dirección no es un entero positivo. Mensaje: `Debe indicar un factura_id entero mayor que 0 para operar sobre la factura.` Corrige el identificador.
`401`	—	No enviaste el encabezado `Authorization`, o el token no existe o expiró. Vuelve a consumir el servicio **Login**. La respuesta tiene la forma del middleware (`statusCode` + `error.description`).
`403`	`CONSUMIDOR_NO_AUTENTICADO`	El token es válido, pero no corresponde a un cliente OAuth habilitado para consumir la API. Mensaje: `El cliente autenticado no está autorizado para consumir este endpoint.`
`403`	`RECURSO_NO_AUTORIZADO`	Al cliente OAuth le falta el permiso `catalogs:read`. Mensaje: `El consumidor no tiene habilitado el recurso solicitado.` Solicita el ajuste a tu administrador en nuby.
`404`	`FACTURA_NO_ENCONTRADA`	No existe una factura con ese identificador. Mensaje: `La factura {factura_id} no existe.` Verifica el `factura_id` o búscalo en el listado. La respuesta **no** incluye `warnings`.
`409`	`FACTURA_CONTRATO_NO_SOPORTADA`	La factura se generó desde un contrato de arrendamiento. Incluye `warnings[]` con el `factura_id` y el alcance. No es un error de tu petición: esa factura se gestiona desde el flujo de contratos de nuby.
`500`	`INTERNAL_ERROR`	Fallo técnico al leer la factura. Mensaje: `No fue posible leer la factura de venta. Intente nuevamente o contacte al administrador del sistema.` Reintenta; si persiste, contacta a soporte.

Forma exacta del rechazo `409`

Una factura de contrato se rechaza con el código `FACTURA_CONTRATO_NO_SOPORTADA`, un mensaje que explica el motivo y una advertencia que identifica la factura y el alcance del endpoint:

```
{
  "success": false,
  "status": 409,
  "message": "La factura 990 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": 990,
      "scope": "solo_facturas_de_venta"
    }
  ]
}
```

Campo de `warnings[]`	Tipo	Significado
`code`	string	Siempre `FACTURA_CONTRATO_NO_SOPORTADA`. Es el valor que debes evaluar en tu código.
`factura_id`	integer	Identificador de la factura rechazada.
`scope`	string	Siempre `solo_facturas_de_venta`: confirma que el rechazo es por alcance, no por permisos.

> **Trata el `404` como un caso de negocio**, no como una falla técnica: significa que el identificador no existe en nuby. No lo reintentes con el mismo valor.

> **Buenas prácticas de seguridad:** el token viaja en el encabezado `Authorization`, nunca en la dirección (query string). No registres datos personales del tercero en tus logs: para trazar la consulta te basta con el `factura_id`.

---

5. Ejemplos de integración

Aquí tienes ejemplos listos para que tus desarrolladores los adapten a tu plataforma. Todos consultan una factura concreta y tratan los dos rechazos de negocio más frecuentes (`404` y `409`).

cURL

```
TOKEN="TU_TOKEN_AQUI"
INSTANCIA="mi-inmobiliaria.nuby.app"
FACTURA_ID=10231

curl "https://${INSTANCIA}/service/v2/public/invoices/${FACTURA_ID}" \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer ${TOKEN}"
```

PHP

```
<?php
$instancia = 'mi-inmobiliaria.nuby.app'; // Tu dirección web completa
$token     = 'TU_TOKEN_AQUI';            // Token obtenido del servicio Login
$facturaId = 10231;

$url = "https://{$instancia}/service/v2/public/invoices/{$facturaId}";

$ch = curl_init($url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_HTTPHEADER, [
    'Content-Type: application/json',
    "Authorization: Bearer {$token}",
]);

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

$info = json_decode($respuesta, true);
```

```
switch ($httpCode) {
    case 200:
        $factura = $info['body'];
        echo "Factura {$factura['consecutivo']} del {$factura['fecha']}\n";
        echo "Total: {$factura['valor_total']} - Saldo: {$factura['valor_saldo']}\n";
        echo "Estado DIAN: " . ($factura['estado_dian'] ?? 'sin enviar') . "\n";
        echo "PDF: {$factura['pdf_link']}\n";

        foreach ($factura['items'] as $item) {
            echo " - {$item['descripcion']} x {$item['cantidad']} =
{$item['valor_unitario']}\n";
        }
        break;

    case 404:
        echo "La factura no existe en nuby: {$info['message']}\n";
        break;

    case 409:
        echo "Factura de contrato, no soportada por esta API: {$info['message']}\n";
        break;

    default:
        echo "Error HTTP {$httpCode}: {$info['message']}\n";
}
```

Python

```
import requests

# 1. Configura tus datos de acceso
instancia = "mi-inmobiliaria.nuby.app"  # Reemplaza con tu dirección web completa
token = "TU_TOKEN_AQUI"                 # Token obtenido del servicio Login
factura_id = 10231
```

```
# 2. Envía la petición
url = f"https://{instancia}/service/v2/public/invoices/{factura_id}"
headers = {
    "Content-Type": "application/json",
    "Authorization": f"Bearer {token}",
}

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

# 3. Procesa la respuesta
if respuesta.status_code == 200:
    factura = respuesta.json()["body"]
    print(f"Factura {factura['consecutivo']} del {factura['fecha']}")
    print(f"Total: {factura['valor_total']} - Saldo: {factura['valor_saldo']}")
    print(f"Estado DIAN: {factura['estado_dian'] or 'sin enviar'}")
    print(f"CUFE: {factura['cufe'] or 'sin CUFE'}")
    print(f"PDF: {factura['pdf_link']}")

    for item in factura["items"]:
        print(f" - {item['descripcion']} x {item['cantidad']} = {item['valor_unitario']}")
elif respuesta.status_code in (404, 409):
    error = respuesta.json()
    print(f"Rechazo de negocio {error['error_code']}: {error['message']}")
else:
    print(f"Error HTTP {respuesta.status_code}: {respuesta.text}")
```

JavaScript (Fetch API)

```
// 1. Configura tus datos de acceso
const instancia = 'mi-inmobiliaria.nuby.app'; // Reemplaza con tu dirección web completa
const token = 'TU_TOKEN_AQUI';                // Token obtenido del servicio Login

// 2. Función que consulta el detalle de una factura
async function obtenerFactura(facturaId) {
  const url = `https://${instancia}/service/v2/public/invoices/${facturaId}`;
```

```
  const response = await fetch(url, {
    method: 'GET',
    headers: {
      'Content-Type': 'application/json',
      'Authorization': `Bearer ${token}`,
    },
  });

  const info = await response.json().catch(() => ({}));

  if (!response.ok) {
    console.error(`Error HTTP ${response.status} ${info.error_code ?? ''}: ${info.message
?? ''}`);
    return null;
  }

  return info.body;
}

// 3. Consume el detalle
obtenerFactura(10231).then((factura) => {
  if (!factura) {
    return;
  }

  console.log(`Factura ${factura.consecutivo} del ${factura.fecha}`);
  console.log(`Total: ${factura.valor_total} - Saldo: ${factura.valor_saldo}`);
  console.log(`Estado DIAN: ${factura.estado_dian ?? 'sin enviar'}`);
  console.log(`PDF: ${factura.pdf_link}`);

  factura.items.forEach((item) => {
    console.log(` - ${item.descripcion} x ${item.cantidad} = ${item.valor_unitario}`);
  });
});
```

Power Query M (Excel / Power BI)

```
let
    // 1. Configura tus datos de acceso
    instancia = "mi-inmobiliaria.nuby.app", // Reemplaza con tu dirección web completa
    token = "TU_TOKEN_AQUI",                // Token obtenido del servicio Login
    facturaId = "10231",                    // Identificador interno de la factura

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

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

    // 4. Decodifica la respuesta JSON y toma el cuerpo
    jsonResponse = Json.Document(response),
    factura = jsonResponse[body],

    // 5. Cabecera como tabla de una fila
    cabecera = Table.FromRecords({[
        FacturaID = factura[factura_id],
        Consecutivo = factura[consecutivo],
        Fecha = factura[fecha],
        FechaVencimiento = factura[fecha_vencimiento],
        ValorTotal = factura[valor_total],
        ValorSaldo = factura[valor_saldo],
        EstadoDian = factura[estado_dian],
        Cufe = factura[cufe],
        PdfLink = factura[pdf_link],
        TerceroID = factura[tercero_id]
    ]}),

    // 6. Ítems como tabla aparte, unida por el identificador de la factura
    items = Table.FromList(factura[items], Splitter.SplitByNothing(), null, null,
ExtraValues.Error),
```

```
    itemsExpandidos = Table.ExpandRecordColumn(items, "Column1",
        {"detalle_id", "producto", "descripcion", "cantidad", "valor_unitario", "iva",
"por_iva"},
        {"DetalleID", "Producto", "Descripcion", "Cantidad", "ValorUnitario", "Iva",
"PorIva"}),
    itemsConFactura = Table.AddColumn(itemsExpandidos, "FacturaID", each
factura[factura_id], Int64.Type)
in
    itemsConFactura
```

> **Sobre Power Query M:** el detalle de una factura se modela mejor como dos tablas (cabecera e ítems) relacionadas por `FacturaID`. El ejemplo devuelve los ítems; usa `cabecera` si solo necesitas los datos del documento.

6. Buenas prácticas y límites conocidos

Recomendaciones

- **Consulta el detalle solo cuando lo necesites.** El listado ya entrega el `factura_id`, los valores y el estado; el detalle agrega CUFE, ítems y el enlace del PDF.
- **Verifica `null` antes de usar un dato.** `cufe` y `estado_dian` llegan en `null` mientras la factura no se haya enviado.
- **Usa el detalle para confirmar el estado antes de emitir.** `estado_id` decide si la factura todavía se puede editar o emitir.
- **No uses `pdf_link` como enlace permanente.** Tiene vigencia de 30 días; pídelo de nuevo con el servicio de PDF.
- **No guardes el detalle como si fuera inmutable.** El estado DIAN y el saldo cambian con el tiempo: vuelve a consultar antes de tomar decisiones de cobro.
- **No registres datos personales en tus logs.** Para trazar una consulta te basta con el `factura_id`.

Límites de este servicio

- **Solo devuelve facturas de venta.** Las de contrato se rechazan con `409 FACTURA_CONTRATO_NO_SOPORTADA`.
- **No devuelve el archivo PDF**, solo el enlace seguro de impresión.
- **No devuelve los datos del tercero.** Solo su `tercero_id`: para el nombre y el documento usa el listado o el servicio de terceros.

- **No devuelve las notas crédito o débito.** El conteo está en el listado (`notas`), pero el detalle de cada nota no forma parte de esta respuesta.
- **No pagina.** Es una consulta de un solo documento: `pagination` y `filters_applied` vienen como listas vacías.
- **No expone datos de contrato** (movimientos, documento de respaldo ni inmobiliaria): esta API es solo de venta.
- **No permite elegir campos.** La respuesta siempre trae el mismo conjunto de datos.

> **¿Necesitas el listado completo?** El **listado de facturas de venta** devuelve los documentos paginados con sus filtros. ¿Solo necesitas el comprobante? Pide el **enlace del PDF**.

Revisión #2
Creado el 28 octubre 2026 14:40:05 por Jorge Iván González
Actualizado el 20 noviembre 2026 14:48:28 por Isabel Higuita