Listar Facturas de Venta

Este servicio devuelve una **lista paginada de las facturas de venta** registradas en nuby. Cada elemento incluye el número de factura, la resolución, las fechas, el valor total, el saldo pendiente, el tercero facturado, el estado del documento, la información de la DIAN, los datos de auditoría y el enlace seguro de su PDF.

Es el listado del **ciclo administrativo**: aplica el predicado que **excluye las facturas de contrato** y avisa cuántas descartó. Si ya tienes el identificador de una factura y necesitas todos sus datos, usa el detalle; si solo necesitas el enlace de impresión, usa el de PDF.

> **¿Para qué sirve este servicio?**
> Úsalo para **sincronizar la información de facturación** con tu sistema externo y para alimentar cualquier proceso que necesite recorrer las facturas de venta de la inmobiliaria. Los casos de uso más frecuentes son:

- **Sincronizar la cartera.** Recorre las facturas pendientes con `pendientes=true` y actualiza los saldos en tu sistema con `saldo`.
- **Generar reportes de facturación.** Filtra por rango de fechas con tu propio proceso sobre `fecha_factura`, o por resolución con `resolucion_id`.
- **Alimentar dashboards.** Publica indicadores de facturación y recaudo con `valor_total`, `saldo` y `estado`.
- **Validar el estado de cobro.** Distingue lo facturado (`estado=1`), lo pagado (`estado=3`) y lo anulado (`estado=2` o `estado=6`).
- **Exportar a Excel o Power BI.** Recorre las páginas y vuelca el resultado en una tabla; en la sección de ejemplos tienes el código listo para Power Query M.
- **Conciliar con un cliente puntual.** Envía el documento del tercero en `documento` y obtén exactamente sus facturas.

---

1. El Endpoint (La dirección web)

Método y dirección

Dato	Valor
Método	`GET`

Dato	Valor
Recurso	`/invoices/sales/list`
Cuerpo de la petición	No se envía cuerpo

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

¿Qué debo colocar en `{{instancia}}`?

Es muy sencillo: corresponde a la **dirección web principal** que utilizas a diario para ingresar a tu plataforma, incluyendo la terminación `.nuby.app` o `.arrendasoft.co`. Por ejemplo, si para entrar a tu sistema escribes `inmobiliaria.nuby.app` en el navegador, esa será exactamente tu instancia.

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

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 que consume la API. 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, como `GET /invoices/list`). 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. Exigir uno que no está provisionado dejaría la consulta inutilizable.

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

> **No uses `GET /invoices/list` para este propósito.** Ese endpoint es el **listado legacy y quedó congelado**: conserva el comportamiento anterior y **puede devolver facturas de contrato**. No es la superficie de solo venta. Usa siempre `GET /invoices/sales/list`, que nunca incluye facturas de contrato y además te avisa cuántas descartó. El endpoint legacy sigue documentado en el libro (página *Listar Facturas*), pero su contrato ya no evoluciona.

2. La Petición (¿Qué debes enviarnos?)

Este servicio **no requiere cuerpo** en la petición. Envía el encabezado de autorización y, opcionalmente, los parámetros de consulta en la dirección (query string).

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 consulta (query string)

Todos los parámetros son **opcionales**. Si no envías ninguno, recibes la primera página con `10` facturas de venta.

Parámetro	Tipo	Requerido	Por defecto	Descripción
`page`	integer	No	`1`	Número de página que se desea recuperar, empezando en `1`. Debe ser numérico y positivo; `page=0`, `page=-3` o `page=abc` producen `400`.

Parámetro	Tipo	Requerido	Por defecto	Descripción
`page_size`	integer	No	`10`	Cantidad de facturas por página. Debe ser numérico y positivo. Valor máximo permitido: `1000`; si pides más, nuby lo ajusta a `1000` sin devolver error.
`estado`	integer	No	Sin filtro	Estado del documento. Valores permitidos: `1` Facturada, `2` Anulada, `3` Pagada, `5` Borrador, `6` Anulada por NC. Cualquier otro valor produce `400` con el catálogo completo en el mensaje.
`documento`	string	No	Sin filtro	Documento de identidad o NIT del tercero. La comparación es **exacta**, no parcial: sirve para traer todas las facturas de un cliente concreto. Un valor vacío no filtra.
`pendientes`	boolean	No	`false`	Con `true`, devuelve solo facturas con saldo pendiente: estado Facturada y `saldo > 0`. Acepta `true`/`false` (también `1`/`0`); cualquier otro valor se interpreta como `false`.
`incluir_borrador`	boolean	No	Sin valor	Solo tiene efecto **junto con** `pendientes=true`: agrega las facturas en Borrador (estado `5`) que tengan saldo pendiente. Si lo envías sin `pendientes`, se publica en `filters_applied` pero no cambia el resultado.

Parámetro	Tipo	Requerido	Por defecto	Descripción
`resolucion_id`	string (lista)	No	Sin filtro	Uno o varios identificadores de resolución de facturación separados por coma (por ejemplo `1,2,3`). Debe contener solo enteros positivos. Se usan únicamente los que existen en el sistema: si ninguno existe, la respuesta es `400`.

> **Importante:** `page` y `page_size` deben ser números positivos. Un valor en cero, negativo o con letras produce `400`; no se corrige automáticamente. El servicio valida primero `page_size` y después `page`, así que si envías los dos inválidos, el mensaje será el de `page_size`.

Reglas de validación y mensajes exactos

Parámetro	Valor inválido	Mensaje devuelto (`400`)
`page_size`	`0`, `-5`, `abc`, vacío	`El parámetro "page_size" debe ser un número positivo.`
`page`	`0`, `-5`, `abc`, vacío	`El parámetro "page" debe ser un número positivo.`
`estado`	Cualquier valor fuera de `1`, `2`, `3`, `5`, `6` (incluye `0` y texto)	`El parámetro "estado" debe ser uno de los siguientes valores: 1 (Facturada), 2 (Anulada), 3 (Pagada), 5 (Borrador), 6 (Anulada por NC).`
`resolucion_id`	Algún elemento que no sea entero positivo (por ejemplo `1,x`)	`El parámetro "resolucion_id" debe contener solo IDs numéricos enteros positivos separados por coma (ej: 1,2,3).`
`resolucion_id`	Lista sin ningún identificador utilizable (por ejemplo `, ,`)	`El parámetro "resolucion_id" no contiene IDs válidos.`
`resolucion_id`	Identificadores válidos que no existen en el sistema	`Ninguna de las resoluciones enviadas existe en el sistema.`

Cómo se combinan los filtros

- **Los filtros se suman.** Si envías `estado` y `documento` al mismo tiempo, el resultado debe cumplir ambos.

- `pendientes` **reemplaza el efecto de** `estado`**.** Cuando `pendientes=true`, la condición de estado la fija el servicio (Facturada, y Borrador si además envías `incluir_borrador=true`), y se agrega `saldo > 0`.
- `incluir_borrador` **no funciona solo.** Sin `pendientes=true` no modifica las filas devueltas.
- `documento` **compara el valor completo.** Si envías un fragmento del documento, no habrá coincidencias. Usa el documento tal como está registrado en nuby.
- `resolucion_id` **es tolerante con los IDs inexistentes.** Si envías `4,7` y solo existe el `7`, el listado filtra por el `7`. El `400` solo aparece cuando **ninguno** existe.

Ejemplos de petición

```
GET https://mi-inmobiliaria.nuby.app/service/v2/public/invoices/sales/list
GET https://mi-inmobiliaria.nuby.app/service/v2/public/invoices/sales/list?page=2
GET https://mi-
inmobiliaria.nuby.app/service/v2/public/invoices/sales/list?page=3&page_size=100
GET https://mi-inmobiliaria.nuby.app/service/v2/public/invoices/sales/list?estado=1
GET https://mi-
inmobiliaria.nuby.app/service/v2/public/invoices/sales/list?documento=900123456&pendientes=tru
e
GET https://mi-
inmobiliaria.nuby.app/service/v2/public/invoices/sales/list?pendientes=true&incluir_borrador=t
rue
GET https://mi-inmobiliaria.nuby.app/service/v2/public/invoices/sales/list?resolucion_id=1,2
```

Encabezados de una petición completa

```
GET /service/v2/public/invoices/sales/list?estado=1&page=1&page_size=50 HTTP/1.1
Host: mi-inmobiliaria.nuby.app
Content-Type: application/json
Authorization: Bearer eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9...
```

3. La Respuesta (¿Qué te entregaremos?)

Si tu token es válido, el sistema devuelve un objeto JSON con el estado de la respuesta, el listado de facturas en `body`, la información de paginación en `pagination` y los filtros usados en `filters_applied`. Los resultados están ordenados por **fecha de la factura descendente** (más recientes primero).

```
{
  "status": 200,
  "message": "",
```

```
"body": [
  {
    "factura_id": 10231,
    "factura_numero": 1534,
    "resolucion": "FE - 18764001234567",
    "fecha_factura": "2026-10-21",
    "fecha_vencimiento": "2026-11-20",
    "valor_total": "1190000.00",
    "saldo": "1190000.00",
    "tercero_id": 4821,
    "documento_tercero": "900123456",
    "nombre_tercero": "ACME SAS",
    "tipo_documento_tercero": "NIT",
    "estado_id": 1,
    "estado": "Facturada",
    "estado_dian": "Aprobado",
    "estado_cliente_dian": "Recibida",
    "observaciones": "Factura de administración octubre 2026",
    "notas": 0,
    "creado_por": "Sandra",
    "fecha_envio_dian": "2026-10-21 08:30:15",
    "fecha_creacion": "2026-10-21 07:45:22",
    "pdf_link": "https://<dominio-publico-pdf>?hash=...",
    "detalles": [
      {
        "detalle_id": 55231,
        "producto": "ADMINISTRACION",
        "producto_tipo": null,
        "descripcion": "Administración octubre 2026",
        "contrato_id": null,
        "contrato_numero": null,
        "fecha_inicio_periodo": null,
        "fecha_fin_periodo": null,
        "cantidad": "1",
        "valor_unitario": "1000000.00",
        "iva": "190000.00",
        "por_iva": "19.00",
        "retencion": "0.00",
        "por_retencion": "0.00",
        "reteiva": "0.00",
```

```
        "por_reteiva": "0.00",
        "reteica": "0.00",
        "por_reteica": "0.00",
        "descuento": "0.00",
        "por_descuento": "0.00"
      }
    ]
  },
  {
    "factura_id": 10230,
    "factura_numero": 1533,
    "resolucion": "FE - 18764001234567",
    "fecha_factura": "2026-10-20",
    "fecha_vencimiento": "2026-11-19",
    "valor_total": "4500000.00",
    "saldo": "0.00",
    "tercero_id": 5117,
    "documento_tercero": "1098745231",
    "nombre_tercero": "JULIANA PATRICIA RÍOS MENDOZA",
    "tipo_documento_tercero": "CC",
    "estado_id": 3,
    "estado": "Pagada",
    "estado_dian": "Aprobado",
    "estado_cliente_dian": null,
    "observaciones": "",
    "notas": 1,
    "creado_por": "Sandra",
    "fecha_envio_dian": "2026-10-20 09:12:40",
    "fecha_creacion": "2026-10-20 08:03:11",
    "pdf_link": "https://<dominio-publico-pdf>?hash=...",
    "detalles": []
  }
],
"pagination": {
  "total_records": 15230,
  "total_pages": 305,
  "current_page": 1,
  "page_size": 50,
  "current_page_records": 2,
  "has_next_page": true,
```

```
    "has_previous_page": false
  },
  "filters_applied": { "estado": 1 },
  "warnings": [
    {
      "code": "FACTURAS_CONTRATO_EXCLUIDAS",
      "message": "Se excluyeron 2 facturas de contrato. Esta API solo lista facturas de
venta.",
      "count": 2
    }
  ]
}
```


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 hay resultados. Cuando no hay resultados contiene el mensaje informativo `No se encontraron facturas con los criterios especificados.`
`body`	array	Listado de facturas de la página consultada. Arreglo vacío `[]` si no hay resultados.
`pagination`	object	Información para recorrer los resultados. Ver la tabla de paginación.
`filters_applied`	object | array	Filtros que aplicó el servicio. Lista vacía `[]` si no enviaste filtros; objeto con los filtros usados si enviaste alguno.
`warnings`	array	Advertencias de la consulta. **La clave solo aparece** cuando el servicio descartó facturas de contrato.

Campos de cada factura (`body[]`)

Cada elemento del arreglo `body` contiene las siguientes claves:

Campo	Tipo	Descripción

`factura_id`	integer	Identificador interno de la factura. Úsalo para el detalle y para el enlace del PDF.
`factura_numero`	integer	Número consecutivo de la factura. En el detalle, este mismo dato se llama `consecutivo`.
`resolucion`	string	Prefijo y número de la resolución de facturación, en formato `PREFIJO - NÚMERO`. Si la resolución no tiene prefijo, se muestra solo el número.
`fecha_factura`	string	Fecha de la factura en formato `AAAA-MM-DD`. En el detalle se llama `fecha`.
`fecha_vencimiento`	string	Fecha de vencimiento de la factura (`AAAA-MM-DD`).
`valor_total`	decimal	Valor total de la factura.
`saldo`	decimal	Saldo pendiente de pago. Vale `0.00` cuando la factura está pagada. En el detalle se llama `valor_saldo`.
`tercero_id`	integer	Identificador interno del tercero facturado.
`documento_tercero`	string	Número de documento de identidad o NIT del tercero facturado.
`nombre_tercero`	string	Nombre completo del tercero facturado, compuesto por los campos de nombre y apellido que nuby concatena para el tercero.
`tipo_documento_tercero`	string | null	Tipo de documento del tercero (por ejemplo NIT o CC), tal como está configurado en nuby. `null` si el tercero no tiene tipo de documento asignado.
`estado_id`	integer	Código del estado del documento: `1` Facturada, `2` Anulada, `3` Pagada, `5` Borrador, `6` Anulada por NC.
`estado`	string	Estado textual del documento: `Facturada`, `Anulada`, `Pagada`, `Borrador`, `Anulada por NC`. Para cualquier otro código interno muestra `Contabilizada`.
`estado_dian`	string | null	Estado de la factura ante la DIAN. `null` si la factura no se ha enviado.
`estado_cliente_dian`	string | null	Estado de aceptación o rechazo del cliente ante la DIAN. `null` si no aplica o si el documento no se ha enviado.

`observaciones`	string	Observaciones registradas en la factura. Cadena vacía si no tiene.
`notas`	integer	Cantidad de notas débito y crédito asociadas a la factura. `0` cuando no tiene.
`creado_por`	string | null	Nombre del usuario que creó la factura en nuby. `null` si el usuario ya no existe.
`fecha_envio_dian`	string | null	Fecha y hora del envío a la DIAN (`AAAA-MM-DD HH:MM:SS`). `null` si no se ha enviado.
`fecha_creacion`	string	Fecha y hora de creación del registro (`AAAA-MM-DD HH:MM:SS`).
`pdf_link`	string	Enlace seguro de impresión del PDF de la factura. Se genera con una vigencia de **30 días** y no expone la ruta interna del archivo.
`detalles`	array	Ítems de la factura. Arreglo vacío `[]` si la factura no tiene detalle. Ver la tabla siguiente.

> **Sobre los tipos numéricos.** Los importes (`valor_total` , `saldo`) y los contadores pueden llegar como texto decimal (por ejemplo `"1190000.00"`) o como número, según cómo esté configurada la conexión del servicio. Conviértelos explícitamente antes de sumar o comparar: no asumas que `saldo` es siempre un número JSON.

Campos de cada ítem (`detalles[]`)

El listado incluye el detalle de los ítems de cada factura. Para el contrato normalizado de ítems, con los valores convertidos a número, usa el detalle de la factura.

Campo	Tipo	Descripción
`detalle_id`	integer	Identificador interno de la línea.
`producto`	string	Nombre del producto o servicio facturado.
`producto_tipo`	string | null	Relación inmobiliaria configurada en el producto.
`descripcion`	string	Descripción de la línea.

Campo	Tipo	Descripción
`contrato_id`, `contrato_numero`	integer | null	Contrato asociado al movimiento de la línea. En una factura de venta siempre llegan en `null`: esta API no expone facturas de contrato.
`fecha_inicio_periodo`, `fecha_fin_periodo`	string | null	Periodo del movimiento asociado a la línea. `null` cuando la línea no proviene de un movimiento.
`cantidad`	decimal	Cantidad facturada en la línea.
`valor_unitario`	decimal	Valor unitario de la línea.
`descuento`, `por_descuento`	decimal	Descuento de la línea, en valor y en porcentaje.
`iva`, `por_iva`	decimal	IVA de la línea, en valor y en porcentaje.
`retencion`, `por_retencion`	decimal	Retención en la fuente de la línea, en valor y en porcentaje.
`reteiva`, `por_reteiva`	decimal	Retención de IVA de la línea, en valor y en porcentaje.
`reteica`, `por_reteica`	decimal	Retención de ICA de la línea, en valor y en porcentaje.

Campos de paginación (`pagination`)

El objeto `pagination` contiene las siguientes claves:

Campo	Tipo	Descripción
`total_records`	integer	Total de facturas de venta que cumplen los filtros, en todas las páginas.
`total_pages`	integer	Cantidad total de páginas (total de registros dividido por el tamaño de página, redondeado hacia arriba). Vale `0` cuando no hay resultados.
`current_page`	integer	Número de la página devuelta.
`page_size`	integer	Tamaño de página efectivamente aplicado (recuerda que se ajusta a `1000` como máximo).
`current_page_records`	integer	Cantidad de registros efectivamente devueltos en esta página.
`has_next_page`	boolean	`true` si existe una página posterior que se puede consultar.

Campo	Tipo	Descripción
`has_previous_page`	boolean	`true` si existe una página anterior.

Nota sobre facturas sin resultados

> Cuando ninguna factura coincide con los filtros, el servicio responde `200` con `body: []`, la paginación en cero y el mensaje informativo. Una lista vacía **no es un error**: revisa `pagination.total_records` para confirmar que no hay coincidencias.


```
{
  "status": 200,
  "message": "No se encontraron facturas con los criterios especificados.",
  "body": [],
  "pagination": {
    "total_records": 0,
    "total_pages": 0,
    "current_page": 1,
    "page_size": 10,
    "current_page_records": 0,
    "has_next_page": false,
    "has_previous_page": false
  },
  "filters_applied": { "estado": 1 }
}
```


Observa que `current_page` y `page_size` conservan lo que solicitaste; lo que queda en cero es `total_records`, `total_pages` y `current_page_records`.

Orden de los resultados

Las facturas se devuelven ordenadas por **fecha de la factura de forma descendente** y, cuando dos facturas comparten fecha, por **identificador descendente**. Es decir: primero las más recientes y, entre las de la misma fecha, la que se registró después. La misma consulta repetida devuelve el mismo orden, así que puedes recorrer el listado con un criterio estable.

> **No ordenes por tu cuenta los resultados de una página para "reconstruir" el listado.** La paginación se calcula con ese orden en el servidor; si reordenas en tu sistema, las páginas dejan de ser comparables. Y si los datos cambian mientras recorres las páginas (por ejemplo, porque se factura un documento nuevo), puedes ver corrimientos: recorre el listado en una ventana corta y vuelve a empezar si necesitas el corte exacto.

Advertencia `FACTURAS_CONTRATO_EXCLUIDAS`

El listado excluye las facturas de contrato. Cuando descarta alguna, agrega una advertencia con la cantidad para que sepas que el resultado no está incompleto por un error:

```
"warnings": [
  {
    "code": "FACTURAS_CONTRATO_EXCLUIDAS",
    "message": "Se excluyeron 2 facturas de contrato. Esta API solo lista facturas de venta.",
    "count": 2
  }
]
```

Campo	Tipo	Significado
`code`	string	Siempre `FACTURAS_CONTRATO_EXCLUIDAS`. Es el valor que debes evaluar en tu código; el `message` puede cambiar de redacción.
`message`	string	Texto legible para una persona. Incluye la cantidad de facturas descartadas.
`count`	integer	Cantidad de facturas de contrato que quedaron fuera del listado con los filtros enviados.

> **Si la clave `warnings` no aparece, no había facturas de contrato** en el rango consultado. Su ausencia nunca significa que el filtro falló: significa que no hubo nada que descartar.

> `filters_applied` **solo publica tus filtros.** El filtro que excluye las facturas de contrato es un detalle interno del alcance del endpoint y **no aparece** en

> `filters_applied`. Si no enviaste filtros, verás una lista vacía `[]`; si enviaste alguno, verás un objeto con los que se aplicaron.

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 pase expira y el sistema te bloquea el acceso con un error `401`. Cuando esto ocurra, tu sistema simplemente debe volver a consumir el servicio de **Login** para pedir un pase nuevo y continuar trabajando.

Así se ve el error que devuelve el sistema cuando el token expiró o es inválido. Esta respuesta la produce el **middleware de autenticación heredado** (el mismo de toda la API) y no el contrato de la operación:

```
{
  "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`. No intentes leerlos con el mismo parser de los errores de negocio. 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.`

Situación	HTTP	`error.description` exacta
El token expiró	`401`	`JWT Token expired.`

Errores del contrato de la operación

Los errores que produce la operación misma (parámetros, permisos del cliente, fallo interno) traen siempre `success: false`, el código HTTP, un `message` legible y un `error_code` estable:

```
{
  "success": false,
  "status": 400,
  "message": "El parámetro \"page\" debe ser un número positivo.",
  "error_code": "VALIDATION_ERROR"
}
```

Código HTTP	`error_code`	Cuándo ocurre y cómo resolverlo
`400`	`VALIDATION_ERROR`	Un parámetro de consulta no es válido. Posibles causas: `page` no es numérico o no es positivo; `page_size` no es numérico o no es positivo; `estado` no es uno de los valores permitidos; `resolucion_id` trae un valor que no es entero positivo, no contiene ningún ID utilizable, o ninguno de los IDs existe en el sistema. Corrige el parámetro y repite la consulta.
`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.` Revisa con tu administrador qué cliente emitió el token.
`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.

Código HTTP	`error_code`	Cuándo ocurre y cómo resolverlo
`500`	`INTERNAL_ERROR`	Fallo interno al consultar las facturas. Mensaje: `Error interno del servidor al obtener las facturas.` Reintenta; si persiste, contacta a soporte e informa el momento en que ocurrió.

Mensajes exactos de `400 VALIDATION_ERROR`

- `El parámetro "page_size" debe ser un número positivo.`
- `El parámetro "page" debe ser un número positivo.`
- `El parámetro "estado" debe ser uno de los siguientes valores: 1 (Facturada), 2 (Anulada), 3 (Pagada), 5 (Borrador), 6 (Anulada por NC).`
- `El parámetro "resolucion_id" debe contener solo IDs numéricos enteros positivos separados por coma (ej: 1,2,3).`
- `El parámetro "resolucion_id" no contiene IDs válidos.`
- `Ninguna de las resoluciones enviadas existe en el sistema.`

> **La advertencia `FACTURAS_CONTRATO_EXCLUIDAS` no es un error:** llega con `200` en `warnings[]` y solo informa cuántas facturas de contrato quedaron fuera del listado.

> **Buenas prácticas de seguridad:** el token viaja en el encabezado `Authorization`, nunca en la dirección (query string), porque las direcciones quedan en logs y proxies. No incluyas el token en tus URLs ni en tus capturas de pantalla, y no registres datos personales de los terceros 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 el listado de solo venta y recorren la paginación.

cURL

```
# Sin filtros: primera página, 10 facturas
curl "https://{{instancia}}/service/v2/public/invoices/sales/list" \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer TU_TOKEN_AQUI"

# Página 3 con 100 registros, solo facturas pendientes de un cliente
curl -G "https://{{instancia}}/service/v2/public/invoices/sales/list" \
  --data-urlencode "page=3" \
  --data-urlencode "page_size=100" \
  --data-urlencode "pendientes=true" \
  --data-urlencode "documento=900123456" \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer TU_TOKEN_AQUI"
```

PHP

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

$pagina      = 1;
$porPagina   = 100;
$facturas    = [];
$totalPaginas = 1;

do {
    $url = "https://{$instancia}/service/v2/public/invoices/sales/list?" .
http_build_query([
        'page'      => $pagina,
        'page_size' => $porPagina,
        'estado'    => 1,
    ]);

    $ch = curl_init($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);

    if ($httpCode !== 200) {
        echo "Error HTTP {$httpCode}: {$respuesta}\n";
        break;
    }

    $info = json_decode($respuesta, true);

    if (!empty($info['warnings'])) {
        foreach ($info['warnings'] as $advertencia) {
            echo "Aviso {$advertencia['code']}: {$advertencia['message']}\n";
        }
    }

    $facturas     = array_merge($facturas, $info['body'] ?? []);
    $totalPaginas = (int) ($info['pagination']['total_pages'] ?? 0);
    $pagina++;
} while ($pagina <= $totalPaginas);

foreach ($facturas as $factura) {
    echo "Factura {$factura['factura_numero']} - {$factura['nombre_tercero']}"
        . " - {$factura['estado']} - Total: {$factura['valor_total']}"
        . " - Saldo: {$factura['saldo']}\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

# 2. Prepara la dirección y los encabezados
url = f"https://{instancia}/service/v2/public/invoices/sales/list"
headers = {
    "Content-Type": "application/json",
    "Authorization": f"Bearer {token}",
}

# 3. Recorre la paginación
pagina = 1
total_paginas = 1

while pagina <= total_paginas:
    respuesta = requests.get(
        url,
        headers=headers,
        params={"page": pagina, "page_size": 100, "pendientes": "true"},
        timeout=30,
    )

    if respuesta.status_code != 200:
        print(f"Error HTTP {respuesta.status_code}: {respuesta.text}")
        break

    info = respuesta.json()
    pagination = info.get("pagination", {})
    total_paginas = pagination.get("total_pages", 0)

    for advertencia in info.get("warnings", []):
        print(f"Aviso {advertencia['code']}: {advertencia['message']}")

    for factura in info.get("body", []):
        print(
            f"#{factura['factura_numero']} - {factura['nombre_tercero']}"
            f" - {factura['estado']} - Total: {factura['valor_total']}"
```

```
            f" - Saldo: {factura['saldo']}"
        )

    pagina += 1
```

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 recorre todas las páginas
async function listarFacturasDeVenta() {
  const facturas = [];
  let pagina = 1;
  let totalPaginas = 1;

  while (pagina <= totalPaginas) {
    const params = new URLSearchParams({
      page: pagina,
      page_size: 100,
      estado: 1,
    });

    const url = `https://${instancia}/service/v2/public/invoices/sales/list?${params}`;

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

    if (!response.ok) {
      console.error(`Error HTTP ${response.status}:`, await response.text());
```

```
      return facturas;
    }

    const info = await response.json();
    totalPaginas = info.pagination?.total_pages ?? 0;

    (info.warnings ?? []).forEach((advertencia) => {
      console.warn(`Aviso ${advertencia.code}: ${advertencia.message}`);
    });

    facturas.push(...(info.body ?? []));
    pagina += 1;
  }

  return facturas;
}

// 3. Consume el listado
listarFacturasDeVenta().then((facturas) => {
  facturas.forEach((factura) => {
    console.log(
      `#${factura.factura_numero} - ${factura.nombre_tercero}` +
      ` - ${factura.estado} - Total: ${factura.valor_total} - Saldo: ${factura.saldo}`
    );
  });
});
```

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

    // 2. Prepara la dirección de la petición (máximo 1000 por página)
    url = "https://" & instancia &
"/service/v2/public/invoices/sales/list?page=1&page_size=1000",
```

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

    // 4. Decodifica la respuesta JSON
    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",
        {"factura_id", "factura_numero", "resolucion", "fecha_factura",
"fecha_vencimiento",
         "valor_total", "saldo", "tercero_id", "documento_tercero", "nombre_tercero",
         "estado", "estado_dian", "notas", "creado_por", "fecha_creacion", "pdf_link"},
        {"FacturaID", "Numero", "Resolucion", "FechaFactura", "FechaVencimiento",
         "ValorTotal", "Saldo", "TerceroID", "DocumentoTercero", "NombreTercero",
         "Estado", "EstadoDian", "Notas", "CreadoPor", "FechaCreacion", "PdfLink"}),
    tipos = Table.TransformColumnTypes(expandido, {
        {"FacturaID", Int64.Type},
        {"Numero", Int64.Type},
        {"FechaFactura", type date},
        {"FechaVencimiento", type date},
        {"ValorTotal", type number},
        {"Saldo", type number}
    }),
    // 6. El total de páginas queda disponible para repetir la consulta con page=2,
page=3, ...
    totalPaginas = pagination[total_pages]
in
    tipos
```

> **Sobre Power Query M:** una consulta `Web.Contents` devuelve una sola página. El paso anterior lee `pagination.total_pages`, así que puedes parametrizar el número de página en la función y combinar los resultados con `Table.Combine` para traer todas.

6. Buenas prácticas y límites conocidos

Recomendaciones

- **Usa `documento` y `estado` para acotar.** Son los filtros que más reducen el volumen de la respuesta.
- **No pidas `page_size=1000` "por si acaso".** Pagina con un tamaño razonable y recorre con `page` hasta `total_pages`.
- **Guarda el `factura_id`, no el número.** El identificador es la llave para el detalle y para el PDF; el número de factura depende de la resolución con la que se numeró el documento.
- **Recorre la paginación con `has_next_page`.** Es más robusto que comparar `current_page` con `total_pages`, porque no depende del tamaño de página que hayas pedido.
- **Interpreta `warnings[]` como información, no como falla.** Si aparece `FACTURAS_CONTRATO_EXCLUIDAS`, el listado simplemente descartó documentos de contrato.
- **Comprueba `total_records` antes de asumir un vacío.** Una lista vacía con `total_records: 0` significa que no hubo coincidencias; con `total_records > 0`, significa que pediste una página fuera de rango.
- **Convierte los importes explícitamente.** Si tu cliente recibe `"1190000.00"` como texto, conviértelo antes de sumar o comparar.

Límites de este servicio

- **Solo lista facturas de venta.** Las facturas de contrato quedan fuera por diseño y se informan en `warnings[]`.
- **No es `GET /invoices/list`.** Ese endpoint es legacy, quedó congelado y puede devolver facturas de contrato.
- **El tamaño máximo por página es 1000.** Un valor mayor se ajusta a `1000`, no se rechaza.
- **No devuelve CUFE.** El código único de factura electrónica solo está en el detalle.
- **Los ítems del listado no están normalizados.** Su contrato estable es el de `items` en el detalle, donde los valores llegan convertidos a número.
- **El `pdf_link` vence a los 30 días.** Si vas a guardar el enlace, revisa su fecha de generación o pídelo de nuevo con el servicio de PDF.
- **No filtra por rango de fechas.** El endpoint no acepta fechas: si necesitas un periodo, recorre las páginas y filtra en tu sistema con `fecha_factura`.
- **No ordena por otro criterio.** El orden es fijo (fecha descendente, identificador descendente) y no se puede cambiar con parámetros.

> **Siguiente paso:** cuando ya tengas el `factura_id`, consulta el **detalle de la factura** para obtener el CUFE, el estado DIAN y los ítems, o pide directamente el **enlace del PDF**.

Revisión #2
Creado el 28 octubre 2026 14:14:41 por Jorge Iván González
Actualizado el 20 noviembre 2026 14:46:35 por Isabel Higuita