

Emitir Factura a la DIAN

[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 **emite a la DIAN** una factura de venta que está en borrador: nuby asigna el **consecutivo real** con la resolución del documento, pasa la factura a `ACTIVA` (`estado_id = 1`) y envía el documento electrónico al proveedor tecnológico.

Emitir es un **acto separado y explícito**: no existe `emitir: true` al crear, y la emisión no vuelve a crear el detalle ni la contabilidad —reutiliza lo que el borrador ya guardó—. Una vez emitida, la factura no se puede editar. Y como el envío no es transaccional con la DIAN, el reintento forma parte del flujo normal: está diseñado para que puedas repetirlo sin duplicar el documento.

¿Para qué sirve este servicio?

Sirve para **convertir un borrador en un documento legal**. Es el único momento en que la factura recibe su número definitivo y en que sale hacia la DIAN. Hasta aquí el borrador era un documento interno sin numerar; después de emitir es una factura electrónica con CUFE, número y estado ante la DIAN.

Casos de uso concretos

- **Cerrar el ciclo de facturación.** Creas el borrador, lo revisas, lo ajustas y lo emites cuando el área contable da el visto bueno.
- **Facturar por lotes de negocio.** Emites una factura por petición, en el orden que decidas: por cliente, por fecha de corte o por contrato.
- **Recuperar un envío fallido.** Si la DIAN rechazó o no respondió, la factura queda activa con el envío fallido y puedes reintentar con la misma `referencia_externa` sin duplicar el documento.
- **Reaccionar a caídas de red.** Si la petición se cortó y no sabes si la emisión ocurrió, repites la operación: si la factura ya tiene CUFE, nuby te devuelve su estado sin reenviar nada.

- **Sincronizar el estado fiscal.** Después de emitir consultas el detalle para obtener `cufe` y `estado_dian`, y pides el PDF para enviarlo a tu cliente.

Resumen del servicio

Aspecto	Valor
Método y ruta	POST /service/v2/public/invoices/{factura_id}/issue
Permiso (scope) requerido	invoices:create
Respuesta exitosa	200 con <code>factura_id</code> , <code>estado_id</code> , <code>consecutivo</code> , <code>cufe</code> , <code>estado_dian</code> , <code>fecha_envio_dian</code> , <code>dian_result</code> e <code>idempotent_replay</code>
Efecto del documento	Estado 1 (Activa), con consecutivo real, CUFÉ y estado DIAN persistidos
Protección contra duplicados	Idempotencia por <code>referencia_externa</code> + emisión idempotente por <code>cufe</code>
¿Se puede repetir?	Sí: repetir una emisión nunca duplica el documento

Lo que este servicio NO hace

- **No edita el documento.** No acepta cambios de cabecera ni de ítems: usa el borrador tal como está. Si necesitas corregirlo, hazlo antes con [Editar Factura en Borrador](#).
- **No reenvía una factura ya emitida.** Si la factura ya tiene CUFÉ, nuby devuelve su estado actual y no contacta a la DIAN.
- **No revalida el período contable.** Se validó al crear el borrador.
- **No cambia la resolución.** Usa la que el documento ya tiene asignada.
- **No es transaccional con la DIAN.** El envío puede fallar después de asignar el consecutivo; por eso el reintento es parte del flujo normal y no un caso excepcional.
- **No acepta lotes.** Se emite una factura por petición.
- **No anula ni corrige.** Anulaciones y notas crédito se gestionan en la interfaz de nuby.

Dónde encaja en el ciclo

1. Crear borrador	POST /invoices	(guía anterior)
2. Editar borrador	PUT PATCH /invoices/{factura_id}	
3. Emitir a la DIAN	POST /invoices/{factura_id}/issue	(esta página)
4. Consultar detalle	GET /invoices/{factura_id}	
5. Obtener el PDF	GET /invoices/{factura_id}/pdf	

El borrador tiene que existir y ser tuyo. Si todavía no lo creaste, empieza por [Crear Factura en Borrador](#): el `factura_id` que devuelve es el que va en el

path de esta operación.

1. El Endpoint (La dirección web)

Apunta tu sistema a la siguiente dirección de petición **POST**:

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

¿Qué debo colocar en `{{instancia}}`?

La **dirección web principal** que utilizas a diario para ingresar a tu plataforma, incluyendo la terminación `.nuby.app` o `.arrendasoft.co`. Por ejemplo `inmobiliaria.nuby.app`.

Importante: no incluyas `https://` ni barras diagonales (`/`) al final. La dirección se compone así: `https://` + tu instancia + `/service/v2/public/invoices/{factura_id}/issue`.

¿Qué debes colocar en `{factura_id}`? El identificador interno del borrador (campo `body.factura_id` de la creación). Debe ser un **entero mayor que 0**. Si no lo es, la respuesta es `400` con *Debe indicar un factura_id entero mayor que 0 para operar sobre la factura*. No lo construyas a partir del consecutivo del borrador: ese valor todavía no es un número de factura.

Encabezados requeridos

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

Permiso requerido (scope)

Este servicio exige el permiso `invoices:create`, el **mismo permiso de escritura de facturas** que la creación. No existe un permiso `invoices:issue`: emitir consume consecutivo y envía a la DIAN, así que se autoriza con el permiso de escritura que ya tienes.

- `403 CONSUMIDOR_NO_AUTENTICADO` → el token es válido pero no corresponde a un cliente OAuth habilitado. Mensaje: *El cliente autenticado no está autorizado para consumir este endpoint.*
- `403 RECURSO_NO_AUTORIZADO` → al cliente le falta `invoices:create`. Mensaje: *El consumidor no tiene habilitado el recurso solicitado.*

Si aún no tienes un token, consulta la página [Login](#) del libro **API nuby V2**.

Precondiciones de la emisión

nuby las evalúa en este orden y se detiene en la primera que falle. Ninguna consume consecutivo.

#	Precondición	Qué pasa si no se cumple
1	La factura es de venta (no nació de un contrato de arrendamiento).	<code>409 FACTURA_CONTRATO_NO_SOPORTADA</code> con <code>warnings[]</code> . No se evalúa nada más.
2	La factura la creó tu consumidor por API.	<code>403 RECURSO_AJENO</code> .
3	La facturación electrónica está habilitada (<code>ENABLE_FACT_ELECT = 1</code>).	<code>422 FACTURACION_ELECTRONICA_DESHABILITADA</code> . No se toca la factura.
4	La factura existe.	<code>404 FACTURA_NO_ENCONTRADA</code> .
5	La factura no tiene CUFE . Si ya lo tiene, no se reenvía nada.	No es un error: responde <code>200</code> con <code>idempotent_replay: true</code> .
6	La factura está en borrador (<code>estado_id = 5</code>).	<code>409 FACTURA_NO_EMITIBLE</code> con el estado actual en el mensaje.

Período contable y resolución. La emisión **no vuelve a evaluar** el período contable: se validó al crear el borrador. Tampoco recibe una resolución en el cuerpo: se usa la que el documento ya tiene asignada. Si el período se cerró entre la creación y la emisión, esta API no lo detecta por sí sola: verifícalo con tu área contable antes de emitir.

2. La Petición (¿Qué debes enviarnos?)

El cuerpo es un **objeto JSON** con un solo dato: la referencia externa de **esta emisión**. La factura que se emite la decide el `factura_id` del path, no el cuerpo.

Elemento	Valor
Método	POST
Content-Type	application/json
Authorization	Bearer <token>
Cuerpo	Objeto JSON con referencia_externa

Campos del cuerpo

Parámetro	Tipo	Requerido	Por defecto	Descripción
referencia_externa	texto	Sí	—	Identificador único de esta emisión , en tu sistema. Si falta o viene vacía: 422 con El campo "referencia_externa" es obligatorio. No reutilices la referencia de la creación ni la de una edición: son operaciones distintas.

No cambies el cuerpo entre reintentos. Cualquier campo adicional que envíes se ignora para la emisión, pero **cuenta para la huella de idempotencia**. Si reintentas tras un fallo, envía exactamente el mismo cuerpo con la misma referencia_externa; si lo cambias, la operación se trata como distinta (409 IDEMPOTENCY_PAYLOAD_MISMATCH).

La emisión no lleva datos del documento. Ni ítems, ni fecha, ni tercero: todo eso ya está en el borrador. Si necesitas cambiarlo, edita el borrador antes de emitir.

Ejemplo de petición

```
POST /service/v2/public/invoices/10231/issue
Authorization: Bearer <token>
Content-Type: application/json
```

```
{ "referencia_externa": "INV-2026-0001-ISSUE-1" }
```

Referencias: una por operación lógica

Operación	Referencia de ejemplo	Se puede reutilizar en...
Crear el borrador	INV-2026-0001	Nada más: ya quedó consumida por la creación.
Editar el borrador	INV-2026-0001-EDIT-1	Nada más (una referencia por edición).
Emitir	INV-2026-0001-ISSUE-1	Solo el reintento de esta misma emisión.

3. La Respuesta (¿Qué te entregaremos?)

Cuando la DIAN acepta el documento, la factura queda activa con su consecutivo real y la respuesta es 200:

```
{
  "success": true,
  "status": 200,
  "message": "La factura de venta se emitió correctamente.",
  "body": {
    "factura_id": 10231,
    "estado_id": 1,
    "consecutivo": 1534,
    "prefijo": "FE",
    "cufe": "b3f1c0d2e5a94f7c8d10b6a2e4f70c93",
    "estado_dian": "Aprobado",
    "fecha_envio_dian": "2026-10-21 10:42:11",
    "dian_result": { "Result": 0, "State": "Aprobado", "Message": null },
    "idempotent_replay": false
  },
  "idempotent_replay": false
}
```

Estructura principal de la respuesta

Campo	Tipo	Descripción
success	booleano	true cuando la emisión se completó (o cuando la factura ya estaba emitida).
status	entero	Código HTTP replicado en el cuerpo. Aquí siempre 200.

Campo	Tipo	Descripción
<code>message</code>	texto	<i>La factura de venta se emitió correctamente. En un replay: La factura ya había sido emitida; se devuelve su estado actual sin reenviarla a la DIAN.</i>
<code>body</code>	objeto	Estado del documento después de la emisión (tabla siguiente).
<code>idempotent_replay</code>	booleano	Viaja en la raíz y dentro de <code>body</code> : <code>true</code> cuando no se envió nada a la DIAN porque la factura ya estaba emitida, o cuando se devolvió una respuesta guardada.

Campos de `body`

Campo	Tipo	Descripción
<code>factura_id</code>	entero	Identificador de la factura emitida: siempre el del path.
<code>estado_id</code>	entero	Estado del documento después de emitir. Aquí <code>1</code> (Activa o Facturada).
<code>consecutivo</code>	entero	Consecutivo real asignado por nuby con la resolución del documento. Este sí es el número de la factura.
<code>cufe</code>	texto	Código único de factura electrónica que devuelve la DIAN. Es la marca de que el documento se envió: si está presente, la factura ya no se reenvía.
<code>estado_dian</code>	texto	Estado del documento ante la DIAN, tal como lo reportó el proveedor tecnológico (por ejemplo <code>Aprobado</code>).
<code>fecha_envio_dian</code>	texto <code>AAAA-MM-DD HH:MM:SS</code>	Fecha y hora del envío registrado en la factura. <code>null</code> cuando la factura no se ha enviado.
<code>dian_result</code>	objeto <code>null</code>	Respuesta cruda del proveedor tecnológico. Su forma la define el integrador; llega en <code>null</code> cuando no hubo respuesta o cuando la emisión fue un replay.
<code>idempotent_replay</code>	booleano	Copia del indicador de la raíz, dentro del cuerpo.

4. Idempotencia y reintentos

Idempotencia por CUFE

Repetir la emisión de una factura que **ya tiene CUFE no reenvía nada a la DIAN**: responde `200` con el estado actual y `idempotent_replay: true`. Es la forma de salir de una duda por un corte de red: si no sabes si la emisión ocurrió, repite la operación y compara.

```
{
  "success": true,
  "status": 200,
  "message": "La factura ya había sido emitida; se devuelve su estado actual sin reenviarla a la DIAN.",
  "body": {
    "factura_id": 10231,
    "estado_id": 1,
    "consecutivo": 1534,
    "prefijo": "FE",
    "cufe": "b3f1c0d2e5a94f7c8d10b6a2e4f70c93",
    "estado_dian": "Aprobado",
    "fecha_envio_dian": "2026-10-21 10:42:11",
    "dian_result": null,
    "idempotent_replay": true
  },
  "idempotent_replay": true
}
```

La emisión tampoco reescribe el detalle ni la contabilidad. Reutiliza lo que el borrador ya guardó: no vuelve a grabar ítems, impuestos ni asientos. Eso evita que un reintento altere montos ya revisados.

Reglas de reintento

Situación	Estado de la factura	Qué debes hacer
Éxito <code>200</code>	Activa, con CUFE	Guarda <code>consecutivo</code> , <code>cufe</code> y <code>estado_dian</code> . No reutilices la referencia.
<code>200</code> con <code>idempotent_replay: true</code>	Activa, ya estaba emitida	Nada: la emisión ya había ocurrido. Sincroniza tu copia con el estado devuelto.
<code>502 DIAN_ERROR</code>	Activa, con envío fallido	Reintenta con la misma <code>referencia_externa</code> y el mismo cuerpo.

Situación	Estado de la factura	Qué debes hacer
500	Sin cambios o estado incierto	Reintenta con la misma referencia: si la emisión ya ocurrió, obtendrás un replay.
422 de negocio	Sin cambios	La referencia quedó liberada : corrige la causa y reintenta con la misma.
409 FACTURA_NO_EMITIBLE	Fuera de borrador	No insistas. Consulta el estado con Ver Detalle de una Factura .

5. 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 bloquea el acceso. Cuando esto ocurra, tu sistema simplemente debe volver a consumir el servicio de **Login** para pedir un pase nuevo y reintentar la emisión con la **misma** `referencia_externa`.

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

Ese 401 lo emite el middleware de autenticación, no el servicio de facturas, así que no trae `success` ni `error_code`. Los rechazos de la operación sí usan el sobre uniforme descrito abajo.

Sobre uniforme de error

Todos los rechazos de la operación traen `success: false`, `status`, `message` y `error_code`. Además:

- `errors`: solo en rechazos de validación, con las causas por campo.
- `warnings`: solo cuando el rechazo es por alcance. Sus elementos traen `code`, `factura_id` y `scope`.
- `dian_result`: solo en el fallo de la DIAN, cuando hubo respuesta del proveedor.

Orden de evaluación

nuby evalúa en este orden exacto. Importa porque determina qué error recibes cuando hay varios problemas:

#	Qué se evalúa	Rechazo
1	Scope del consumidor	403
2	El <code>factura_id</code> del path es un entero positivo	400 VALIDATION_ERROR
3	El cuerpo es un objeto JSON	400 VALIDATION_ERROR
4	Alcance: ¿es una factura de contrato?	409 FACTURA_CONTRATO_NO_SOPORTADA
5	Pertenencia: ¿la creó tu consumidor?	403 RECURSO_AJENO
6	<code>referencia_externa</code> presente	422 VALIDATION_ERROR
7	Idempotencia de la solicitud	409 IDEMPOTENCY_IN_PROGRESS / 409 IDEMPOTENCY_PAYLOAD_MISMATCH
8	<code>ENABLE_FACT_ELECT = 1</code>	422 FACTURACION_ELECTRONICA_DESHABILITADA
9	La factura existe	404 FACTURA_NO_ENCONTRADA
10	La factura no tiene CUFE	No es error: 200 con <code>idempotent_replay: true</code>
11	Estado 5 (Borrador)	409 FACTURA_NO_EMITIBLE

Tabla de códigos HTTP, causas y mensajes

HTTP	error_code	Causa y mensaje exacto
400	VALIDATION_ERROR	El <code>factura_id</code> del path no es un entero positivo. <i>Debe indicar un <code>factura_id</code> entero mayor que 0 para operar sobre la factura.</i>
400	VALIDATION_ERROR	El cuerpo no es un objeto JSON. <i>El cuerpo de la solicitud debe ser un objeto JSON.</i>
400	—	El encabezado <code>Authorization</code> no tiene el formato <code>Bearer {token}</code> . <i>JWT Token not send.</i>
401	—	Falta el encabezado <code>Authorization</code> : <i>JWT Token required.</i>
401	—	El token no existe: <i>JWT Token not found.</i>

HTTP	error_code	Causa y mensaje exacto
401	—	El token expiró: <i>JWT Token expired</i> . Vuelve a consumir Login .
403	CONSUMIDOR_NO_AUTENTICADO	El token no corresponde a un cliente OAuth habilitado. <i>El cliente autenticado no está autorizado para consumir este endpoint</i> .
403	RECURSO_NO_AUTORIZADO	Al cliente le falta <code>invoices:create</code> . <i>El consumidor no tiene habilitado el recurso solicitado</i> .
403	RECURSO_AJENO	La factura la creó la interfaz de nuby u otro consumidor. <i>La factura indicada no pertenece al consumidor autenticado</i> .
404	FACTURA_NO_ENCONTRADA	La factura no existe. <i>La factura {factura_id} no existe</i> .
409	FACTURA_CONTRATO_NO_SOPORTADA	La factura nació de un contrato de arrendamiento. Incluye <code>warnings[1]</code> con <code>factura_id</code> . Mensaje: <i>La factura {factura_id} 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</i> .
409	FACTURA_NO_EMITIBLE	La factura no está en borrador y no tiene CUFÉ. Mensaje: <i>La factura {factura_id} no está en borrador (estado {estado_id})</i> . Solo se pueden emitir facturas en estado borrador.
409	IDEMPOTENCY_IN_PROGRESS	La misma referencia está siendo procesada por otra solicitud. <i>La solicitud con esta referencia externa está en proceso</i> .
409	IDEMPOTENCY_PAYLOAD_MISMATCH	La misma referencia se envió con contenido distinto (o el cuerpo cambió entre reintentos). <i>La referencia externa ya fue usada con un payload distinto</i> .
422	VALIDATION_ERROR	Falta <code>referencia_externa</code> . <i>El campo "referencia_externa" es obligatorio</i> .
422	FACTURACION_ELECTRONICA_DESHABILITADA	<code>ENABLE_FACT_ELECT</code> no vale <code>1</code> . No se toca la factura. <i>La facturación electrónica no está habilitada en este sistema, así que no se pueden emitir facturas de venta por API</i> .

HTTP	error_code	Causa y mensaje exacto
500	INTERNAL_ERROR	Fallo técnico, con mensaje genérico a propósito: el detalle queda en el log de nuby.
500	IDEMPOTENCY_STORAGE_ERROR	No se pudo registrar la solicitud para controlar duplicados. <i>No se pudo registrar la solicitud de idempotencia.</i>
502	DIAN_ERROR	La DIAN rechazó el envío o no respondió. Arrastra <code>dian_result</code> cuando hubo respuesta y la factura queda activa con el envío fallido. El <code>message</code> es la causa reportada por el proveedor; si no hay causa disponible: <i>No se pudo emitir la factura {factura_id} ante la DIAN. La factura quedó activa con el envío fallido y puede reintentarse.</i>

Forma exacta de `warnings[]` en el rechazo de contrato

```
{
  "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" }
  ]
}
```

Si la DIAN falla: `502 DIAN_ERROR`

Cuando el envío se rechaza o no hay respuesta:

- La factura **queda activa** (`estado_id = 1`) con el `estado_dian` fallido: el que reportó el proveedor o `Fallida` si no hubo respuesta.
- La causa se guarda en la factura y en el registro de errores de la DIAN de nuby.
- La solicitud de idempotencia queda **fallida**, así que puedes reintentar con la **misma** `referencia_externa`.
- El documento **no se duplica**: la emisión es idempotente por CUFÉ.

```
{
  "success": false,
  "status": 502,
  "message": "No se pudo emitir la factura 10231 ante la DIAN. La factura quedó activa con el envío fallido y puede reintentarse.",
  "error_code": "DIAN_ERROR",
  "dian_result": { "Result": 1, "State": "Rechazado", "Message": "Documento duplicado" }
}
```

El reintento es seguro: usa la MISMA `referencia_externa` y el MISMO cuerpo. Generar una referencia nueva para "volver a intentar" crea una operación distinta, pierde la trazabilidad del primer intento y es la única forma de complicar este flujo.

Revisa la causa antes de reintentar. Un `502` puede ser transitorio (timeout, caída del proveedor) o definitivo (documento rechazado por datos). Si el `dian_result.Message` describe un problema de datos, corrígelo —cuando sea posible, editando antes de emitir— en lugar de reintentar en bucle.

Un `factura_id` que no existe suele responder `403 RECURSO_AJENO`. La pertenencia se evalúa antes de la existencia: si nubly no tiene registrado que tu consumidor creó ese identificador, no puede distinguir "no es tuya" de "no existe". El `404 FACTURA_NO_ENCONTRADA` aparece cuando la pertenencia sí está registrada pero la factura ya no existe.

6. Ejemplos de integración

Todos los ejemplos tratan el `502` como un caso esperado: guardan el `error_code` y reintentan con la misma `referencia_externa`.

cURL

```
TOKEN="TU_TOKEN_AQUI"
```

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

```
FACTURA_ID=10231
```

```
curl -X POST "https://${INSTANCIA}/service/v2/public/invoices/${FACTURA_ID}/issue" \  
  -H "Authorization: Bearer ${TOKEN}" \  
  -H "Content-Type: application/json" \  
  -d '{ "referencia_externa": "INV-2026-0001-ISSUE-1" }'
```

PHP

```
<?php
```

```
$instancia = 'mi-inmobiliaria.nuby.app'; // Reemplaza con tu dirección web completa  
$token = 'TU_TOKEN_AQUI';                // Token obtenido del servicio Login  
$facturaId = 10231;                       // factura_id devuelto al crear el borrador  
$referencia = 'INV-2026-0001-ISSUE-1';    // Una referencia por emisión
```

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

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

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

```
if (curl_errno($ch)) {  
    // Timeout o caída de red: reintenta con la MISMA referencia y el MISMO cuerpo.  
    echo 'Error de conexión: ' . curl_error($ch);  
} else {
```

```

$info = json_decode($respuesta, true);

if ($httpCode === 200) {
    $body = $info['body'];

    echo "Factura emitida: factura_id={$body['factura_id']}"
        . " consecutivo={$body['consecutivo']}"
        . " estado_dian={$body['estado_dian']}\n";
    echo "CUFE: {$body['cufe']}\n";

    if (!empty($info['idempotent_replay'])) {
        echo "Nota: la factura ya estaba emitida; no se reenvió a la DIAN.\n";
    }
} elseif ($httpCode === 502) {
    echo "La DIAN no aceptó el envío: {$info['message']}\n";
    echo "Detalle del proveedor: " . json_encode($info['dian_result'] ?? null) . "\n";
    // Reintenta con la MISMA referencia: la emisión es idempotente por CUFE.
} else {
    echo "Rechazo HTTP {$httpCode}: {$info['message']}\n";
    echo "error_code: " . ($info['error_code'] ?? 'N/A') . "\n";
}
}

curl_close($ch);

?>

```

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                      # factura_id devuelto al crear el borrador
referencia = 'INV-2026-0001-ISSUE-1'   # Una referencia por emisión

```

```

# 2. Prepara la dirección y el cuerpo
url = f"https://{instancia}/service/v2/public/invoices/{factura_id}/issue"

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

payload = {"referencia_externa": referencia}

# 3. Envía la petición y procesa la respuesta
try:
    response = requests.post(url, json=payload, headers=headers, timeout=60)

    print(f"Código de estado HTTP: {response.status_code}")

    info = response.json()

    if response.status_code == 200:
        body = info.get("body", {})

        print(f"Factura emitida: factura_id={body.get('factura_id')}")
        print(f"Consecutivo real: {body.get('consecutivo')}")
        print(f"Estado DIAN: {body.get('estado_dian')} | CUFE: {body.get('cufe')}")

        if info.get("idempotent_replay"):
            print("Nota: la factura ya estaba emitida; no se reenvió a la DIAN.")
    elif response.status_code == 502:
        print(f"La DIAN no aceptó el envío: {info.get('message')}")
        print(f"Detalle del proveedor: {info.get('dian_result')}")
        # Reintenta con la MISMA referencia y el MISMO cuerpo.
    else:
        print(f"Mensaje: {info.get('message')}")
        print(f"error_code: {info.get('error_code')}")

    for aviso in info.get("warnings") or []:
        print(f" advertencia: {aviso}")

```

```
except Exception as e:
```

```
# Ante un timeout, reintenta con la MISMA referencia y el MISMO cuerpo.
```

```
print(f"Ocurrió un error de conexión: {e}")
```

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
```

```
const facturaId = 10231; // factura_id devuelto al crear el borrador
```

```
const referencia = 'INV-2026-0001-ISSUE-1'; // Una referencia por emisión
```

```
const url = `https://${instancia}/service/v2/public/invoices/${facturaId}/issue`;
```

```
// 2. Función para emitir la factura
```

```
async function emitirFactura() {
```

```
  try {
```

```
    const response = await fetch(url, {
```

```
      method: 'POST',
```

```
      headers: {
```

```
        'Content-Type': 'application/json',
```

```
        'Authorization': `Bearer ${token}`
```

```
      },
```

```
      body: JSON.stringify({ referencia_externa: referencia })
```

```
    });
```

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

```
    const info = await response.json();
```

```
    if (response.status === 200) {
```

```
      const body = info.body || {};
```

```
      console.log(`Factura emitida: factura_id=${body.factura_id}`);
```

```
      console.log(`Consecutivo real: ${body.consecutivo}`);
```

```
      console.log(`Estado DIAN: ${body.estado_dian} | CUFE: ${body.cufe}`);
```

```

        if (info.idempotent_replay) {
            console.log('Nota: la factura ya estaba emitida; no se reenvió a la
DIAN.');
```

```

        }
    } else if (response.status === 502) {
        console.log(`La DIAN no aceptó el envío: ${info.message}`);
        console.log('Detalle del proveedor:', info.dian_result);
        // Reintenta con la MISMA referencia y el MISMO cuerpo.
    } else {
        console.log(`Mensaje: ${info.message}`);
        console.log(`error_code: ${info.error_code}`);

        (info.warnings || []).forEach((aviso) => console.log(' advertencia:',
aviso));
    }
} catch (error) {
    // Ante un timeout, reintenta con la MISMA referencia y el MISMO cuerpo.
    console.error(`Ocurrió un error de conexión: ${error}`);
}

// 3. Ejecutamos la función
emitirFactura();

```

Power Query M (Excel / Power BI)

La emisión es `POST`, así que Power Query puede consumirla directamente. Si el envío falla, `Web.Contents` genera una excepción: captúrala con `try ... otherwise` para registrar el `error_code` y el detalle del proveedor antes de reintentar.

```

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",                    // factura_id devuelto al crear el borrador

    // 2. Prepara la dirección de la petición

```

```

url = "https://" & instancia & "/service/v2/public/invoices/" & facturaId & "/issue",

// 3. Construye el cuerpo JSON (una referencia por emisión)
cuerpo = Json.FromValue([ referencia_externa = "INV-2026-0001-ISSUE-1" ]),

// 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.FromRecords({body}),
expandido = Table.ExpandRecordColumn(tabla, "Column1",
    {"factura_id", "estado_id", "consecutivo", "prefijo", "cufe", "estado_dian",
"fecha_envio_dian"},
    {"FacturaID", "EstadoID", "Consecutivo", "CUFE", "EstadoDian", "FechaEnvioDian"})
in
expandido

```

Después de emitir, puedes confirmar el estado fiscal con [Ver Detalle de una Factura](#) y entregar el documento a tu cliente con [Obtener el PDF de una Factura](#).

Revisión #1

Creado el 28 octubre 2026 14:12:38 por Jorge Iván González

Actualizado el 28 octubre 2026 14:13:05 por Jorge Iván González