e-FiscalCloud

TU NEGOCIO CONECTADO
CON LA NUEVA ERA FISCAL

GUÍA TÉCNICA PARA POS, ERP Y SOFTWARE

Manual de Integración

API e-CF de e-Fiscal Cloud. Conecta tu sistema a la facturación electrónica de la DGII sin cambiar tu forma de vender.

API	api.efiscalcloud.com
PORTAL	app.efiscalcloud.com
VERSIÓN	1.1 · 04-11-2026
CONTRATO	OpenAPI 3.0.3

Una solución de

KARMA TECH CONSULTING

Documento para socios de integración
soporte@efiscalcloud.com

CONTENIDO

Índice

1. Conceptos clave
2. Ambientes y direcciones
3. Autenticación
4. Identificar la sucursal emisora
5. Emitir un e-CF
6. Estados y consulta del resultado DGII
7. Idempotencia (evitar duplicados)
8. Notas de Crédito y Débito (33/34)
9. Compras recibidas (e-CF de proveedores)
10. Contador de documentos (scope stats:read)
11. Consulta de RNC/Cédula (scope rnc:lookup)
12. Errores frecuentes
13. Reglas fiscales aprendidas en pruebas con la DGII
14. Buenas prácticas y salida a producción
15. Contrato OpenAPI y ejemplos de código
16. Glosario
17. Soporte

01 Conceptos clave

1.1 Qué es un e-CF

- **e-CF:** Comprobante Fiscal Electrónico. Cada documento tiene un **e-NCF**: `E` + 2 dígitos de tipo + 10 dígitos de secuencia (por ejemplo, `E320000000045`).
- **Emisor:** la empresa (RNC) que factura. Cada empresa tiene una o varias **sucursales**.

1.2 Los 10 tipos de comprobante que gestiona la plataforma

Tipo	Descripción	Uso típico
31	Crédito Fiscal	Venta a contribuyente que usa el ITBIS como crédito
32	Consumo	Venta a consumidor final
33	Nota de Débito	Aumenta el valor de un e-CF previo
34	Nota de Crédito	Disminuye o anula un e-CF previo
41	Compras	Compra a proveedor informal (el emisor es el comprador)
43	Gastos Menores	Gasto sin comprobante formal del proveedor
44	Regímenes Especiales	Venta exenta de ITBIS (zonas francas, etc.)
45	Gubernamental	Venta al Estado
46	Exportaciones	Venta al exterior (comprador extranjero, divisa)
47	Pagos al Exterior	Pago a beneficiario extranjero (con retención ISR)

Los tipos **41, 43, 46 y 47** tienen requisitos fiscales propios (retenciones, identificación extranjera, moneda). Revisa la matriz de la sección 5.6 y el estado de soporte de la sección 5.7 antes de integrarlos.

1.3 Qué hace e-Fiscal Cloud por ti

- Asigna la secuencia (e-NCF).
- Construye el XML según el esquema (XSD) de la DGII.
- Lo firma con el certificado digital de la empresa.
- Lo transmite a la DGII (canal íntegro o resumen RFCE, según tipo y monto).
- Da seguimiento al estado (TrackID).
- Opcionalmente, notifica al cliente por correo o WhatsApp y guarda la representación impresa.

Tu sistema solo origina la venta y la envía ya cerrada. No manejas secuencias ni firmas.

1.4 Flujo de extremo a extremo

```
Tu sistema (POS / ERP / e-commerce)
   |  1. Cierra la venta
   ▼
Autenticación (una vez): token del POS o API key de integrador
   |
   ▼
POST /emit  → e-Fiscal Cloud asigna e-NCF, arma el XML, firma y transmite
   |
   ▼
Respuesta inmediata: { documentId, clientUuid, encf, securityCode, signatureDate }
   |
   ▼
Consulta de estado (sondeo) hasta el resultado de la DGII
   |
   ▼
Aceptado · Aceptado condicional · Rechazado
```

02 Ambientes y direcciones

Recurso	Dirección
Portal de clientes	`https://app.efiscalcloud.com`
API	`https://api.efiscalcloud.com`
Información pública de integraciones	`https://efiscalcloud.com/integraciones`

Cada empresa y sucursal opera en uno de tres ambientes de la DGII:

Ambiente	Uso
`test` **(TesteCF)**	Pruebas libres
`cert` **(Certificación)**	Set de pruebas de certificación de la DGII
`prod` **(Producción)**	Facturación real

- El ambiente lo determina la **sucursal**; **no se envía en el payload**.
- La empresa tiene un ambiente por defecto (editable en el portal) y cada sucursal puede heredarlo o fijar el suyo.
- Las empresas y sucursales nuevas arrancan en `test`.
- Para pasar a producción se cambia el ambiente de la sucursal (o de la empresa) en el portal: **tu código no cambia**.

03 Autenticación

3.1 Dos credenciales, dos usos

Credencial	Formato	Para quién	Alcance
Connection key → token del POS	`ecft_…`	POS instalado en una tienda (PixelPoint, Ágora POS, ICG FrontRest/FrontRetail, entre otros)	Una sucursal
API key de integrador	`ik_live_…`	ERP, SaaS o software de terceros multiempresa	Las empresas y sucursales autorizadas a la key

Ambas se envían en el encabezado:

```
Authorization: Bearer <credencial>
```

No se mezclan. Un token del POS (`ecft_…`) no sirve en las rutas `/api/v1/integrator/...`.

3.2 Activar una sucursal (POS)

Cada sucursal se activa con una **connection key** que se genera desde el portal. La integración la canjea por un token:

```
POST /api/v1/portal/keys/validate
Content-Type: application/json

{ "key": "<CONNECTION-KEY-DE-LA-SUCURSAL>" }
```

Respuesta:

```
{ "token": "ecft_mi-sucursal-001_..." }
```

- La key es de **un solo uso** por activación.
- Guarda el token de forma segura: identifica a la sucursal emisora.

3.3 API key de integrador (ERP y plataformas de terceros)

La API key de integrador:

- pertenece al integrador, no a una sucursal;
- tiene **scopes** (permisos) definidos;
- está limitada a las **empresas y sucursales** que se autoricen;
- es **revocable** de forma independiente, sin afectar a los POS en producción.

Ventaja: integras una sola vez y emites para todas las empresas y sucursales permitidas.

Cómo se obtiene: la emite el equipo de e-Fiscal Cloud a solicitud del socio. El secreto se muestra **una sola vez** al crearla; la plataforma guarda solo su hash. Si se pierde, se revoca y se emite otra.

Scopes disponibles

Scope	Permite	Endpoints
`ecf:emit`	Emitir e-CF y consultar su estado	`POST /api/v1/integrator/ecf/emit`, consultas `/ecf/...`, `GET /api/v1/integrator/branches`
`purchases:read`	Listar y consultar compras recibidas	`GET /api/v1/integrator/purchases`, `/purchases/{encf}`
`purchases:approve`	Aprobar o rechazar compras	`POST /api/v1/integrator/purchases/{encf}/approve` y `/reject`
`stats:read`	Contador de documentos (ventas y compras por tipo)	`GET /api/v1/integrator/stats/documents`
`rnc:lookup`	Consulta RNC/Cédula → Razón Social	`GET /api/v1/integrator/rnc/{rnc}`, `/rnc/search`

Verificar la key antes de emitir

```
GET /api/v1/integrator/verify
Authorization: Bearer ik_live_xxxxxxxx
```

Devuelve el nombre de la key, sus `scopes`, `allowedCompanies`, `allowedBranches` y su estado. Si responde `401`, la key está revocada o es inválida.

3.4 Ciclo de vida de la API key

Una sola key para varias empresas. No necesitas una credencial por empresa ni canjear tokens por sucursal: cada emisión identifica su empresa y sucursal (sección 4).

Las keys no se editan. Por seguridad, una key creada no cambia de scopes ni de empresas: solo se **revoca**. Para ampliar su alcance:

1. Solicita una **key nueva** con la configuración ampliada (empresas y scopes anteriores más los nuevos).
2. Configúrala en tu software, en sustitución de la anterior.
3. Pide revocar la **key vieja** cuando confirmes que la nueva funciona.

Recomendación: solicita desde el inicio todas las empresas y todos los scopes que prevés usar (`ecf:emit`, `purchases:read`, `purchases:approve`, `stats:read`, `rnc:lookup`). Usa varias keys solo si quieres poder revocar una integración sin afectar a otra.

3.5 Custodia de credenciales

- Guarda tokens y keys en un gestor de secretos o en variables de entorno del servidor; nunca en el código fuente ni en aplicaciones del lado del cliente.
- Usa una key distinta por integración o por ambiente cuando necesites revocarlas por separado.
- Si una key se expone, pide su revocación de inmediato: los POS en producción no se ven afectados.

04 Identificar la sucursal emisora

Una sucursal tiene **dos identificadores** y es importante no confundirlos:

Identificador	Qué es	¿Único?
`id` **global**	Identificador de la sucursal en toda la plataforma (por ejemplo, `PRINCIPAL`, `EFACTURA-003` o un UUID)	Sí
`branch_code`	Número de sucursal dentro de su empresa (`"001"`, `"002"`...)	**No**: se repite entre empresas

Enviar solo `branchId: "001"` es ambiguo: muchas empresas tienen su sucursal `"001"`. Hay dos formas correctas de identificar la sucursal con una API key de integrador.

Forma A (recomendada): `companyId` + `branchId` como código

```
{
  "companyId": "mi-empresa",
  "branchId": "001",
  "ecfType": "32",
  "...": "resto del payload"
}
```

La plataforma resuelve la sucursal por la combinación (empresa, código), que sí es única. Es el modelo natural para un ERP multiempresa: guardas el `companyId` del cliente y su código de sucursal tal como lo ve el usuario.

Forma B (alternativa): `branchId` como id global, sin `companyId`

```
{ "branchId": "EFACTURA-003", "ecfType": "32", "...": "..." }
```

Funciona igual, pero obliga a tu sistema a guardar el id global.

> **Obligatorio en keys nuevas.** Las API keys creadas a partir del **4 de noviembre de 2026** exigen `companyId` + `branchId` (código) en cada emisión y nota de crédito. Sin `companyId`, la API responde `400`: *"Esta API key exige identificar la empresa: envíe companyId y branchId..."*. Las keys anteriores siguen aceptando la Forma B, pero se recomienda migrar a la Forma A.

Fuente de verdad: `GET /api/v1/integrator/branches` (scope `ecf:emit`). Devuelve cada sucursal autorizada de la key. Úsalo para construir tu mapeo y no adivinar identificadores.

```
{ "branches": [
    { "branchId": "EFACTURA-003", "branchCode": "003", "name": "Sucursal 3",
      "companyId": "mi-empresa", "companyName": "MI EMPRESA SRL",
      "rnc": "101010101", "environment": "test" }
] }
```

Convivencia con el POS. El cliente POS no cambia: su token (`ecft_…`) ya lleva el id global de su sucursal y no envía `companyId`. Las tres formas conviven sin interferir:

1. **Token del POS** → se usa la sucursal del token; `companyId` se ignora.
2. **Integrador con `companyId`** → se resuelve por (empresa, código). Forma A.
3. **Integrador sin `companyId`** → `branchId` como id global. Forma B (keys anteriores).

05 Emitir un e-CF

5.1 Endpoints

Credencial	Ruta
Token del POS (`ecft_…`)	`POST /api/v1/ecf/emit`
API key de integrador (`ik_live_…`, scope `ecf:emit`)	`POST /api/v1/integrator/ecf/emit`

Ambas rutas reciben el mismo cuerpo y devuelven la misma respuesta.

5.2 Cuerpo de la petición

Ejemplo de integrador con la Forma A (un POS omite `companyId` y envía el `branchId` de su token):

```
{
  "companyId": "mi-empresa",
  "branchId": "001",
  "ecfType": "32",
  "posTicketId": "12345",
  "posOpenDate": "2026-10-16",
  "buyer": {
    "rncCedula": "130123456",
    "name": "CLIENTE EJEMPLO SRL",
    "email": "cliente@correo.com",
    "phone": "8095551234",
    "whatsappConsent": true
  },
  "items": [
    {
      "lineNumber": 1,
      "productName": "Producto A",
      "quantity": 2,
      "unitPrice": 100.00,
      "discount": 0,
      "itbisAmount": 36.00,
      "itbisRate": 18,
      "subtotal": 200.00
    }
  ],
  "subtotal": 200.00,
  "taxItbis": 36.00,
  "taxLaw": 0,
  "discount": 0,
  "total": 236.00,
  "payments": [
    { "methodName": "Tarjeta", "formaPago": "3", "monto": 236.00 }
  ],
  "clientUuid": "5b3c8f0e-6a4d-5c1e-9f3a-2d7e1b0c4a91"
}
```

5.3 Campos

Campo	Req.	Notas
`companyId`	cond.	Solo integradores. Obligatorio con keys nuevas (sección 4). Con `companyId`, `branchId` es el **código** de sucursal. No lo uses con el token del POS.
`branchId`	sí	Con token del POS: debe coincidir con el token. Sin `companyId`: id global. Con `companyId`: código de sucursal.
`ecfType`	sí	31, 32, 33, 34, 41, 43, 44, 45, 46 o 47 (matriz en 5.6).
`posTicketId`	reco.	Id del ticket en **tu** sistema. Es la clave de idempotencia (sección 7).

Campo	Req.	Notas
`posOpenDate`	reco.	Fecha de la venta (`YYYY-MM-DD`).
`buyer`	cond.	Obligatorio para 31, 44 y 45, y para 32 desde RD$250,000. Opcional en consumo menor.
`buyer.rncCedula`	cond.	9 dígitos (RNC) u 11 (cédula).
`items[]`	sí	Al menos una línea.
`subtotal`	sí	Base del documento, en **DOP**.
`taxItbis`	sí	Total de ITBIS, en **DOP**.
`taxLaw`	cond.	**Propina legal (10 %)**, en DOP. Se transmite como `MontoPropinaLegal`, se suma al total y **no** es base de ITBIS. `0` si no aplica.
`discount`	opc.	Descuento global del documento.
`total`	sí	Subtotal + ITBIS + propina, en **DOP**.
`payments[]`	reco.	Formas de pago. Si se omite, se asume efectivo por el total.
`clientUuid`	reco.	UUID **determinístico** por ticket: mismo ticket, mismo UUID.

5.4 Formas de pago (`payments[]`)

Cada pago lleva `monto` (en DOP) y, de preferencia, el **código DGII** en `formaPago`. Si no lo envías, la plataforma lo deduce de `methodName`.

Código	Forma de pago
1	Efectivo
2	Cheque / Transferencia
3	Tarjeta de Crédito/Débito
4	Venta a Crédito
5	Bonos / Certificados de Regalo
6	Permuta
7	Nota de Crédito
8	Otras

Se admiten **pagos mixtos** (varias entradas). Si un pago es en moneda extranjera, puedes enviar además `montoForeign`, `changeForeign` y `currencyIso` (USD o EUR) para que la representación impresa muestre ambas monedas. El e-CF siempre va en DOP.

5.5 Factura de Consumo (32): umbral de RD$250,000 y propina

(a) Consumo menor de RD$250,000: comprador opcional

```
{
  "companyId": "mi-empresa", "branchId": "001", "ecfType": "32", "posTicketId": "12346",
  "items": [{ "lineNumber": 1, "productName": "Producto A", "quantity": 2,
              "unitPrice": 100, "itbisAmount": 36, "itbisRate": 18, "subtotal": 200 }],
  "subtotal": 200, "taxItbis": 36, "taxLaw": 0, "total": 236,
  "payments": [{ "formaPago": "1", "monto": 236 }],
  "clientUuid": "22222222-2222-4222-8222-222222222222"
}
```

(b) Consumo desde RD$250,000: comprador obligatorio (RNC y razón social). Sin comprador, la emisión se rechaza.

```
{
  "companyId": "mi-empresa", "branchId": "001", "ecfType": "32", "posTicketId": "12360",
  "buyer": { "rncCedula": "130123456", "name": "CLIENTE MAYORISTA SRL" },
  "items": [{ "lineNumber": 1, "productName": "Lote de mercancía", "quantity": 1,
              "unitPrice": 254237.29, "itbisAmount": 45762.71, "itbisRate": 18,
              "subtotal": 254237.29 }],
  "subtotal": 254237.29, "taxItbis": 45762.71, "taxLaw": 0, "total": 300000,
  "payments": [{ "formaPago": "2", "monto": 300000 }],
  "clientUuid": "32250000-2500-4250-8250-322500002500"
}
```

(c) Consumo con propina legal (restaurantes): la propina va en `taxLaw`, se suma al total y no es base de ITBIS.

```
{
  "companyId": "mi-empresa", "branchId": "001", "ecfType": "32", "posTicketId": "12361",
  "items": [{ "lineNumber": 1, "productName": "Plato del día", "quantity": 2,
              "unitPrice": 500, "itbisAmount": 180, "itbisRate": 18, "subtotal": 1000 }],
  "subtotal": 1000, "taxItbis": 180, "taxLaw": 100, "total": 1280,
  "payments": [{ "formaPago": "3", "monto": 1280 }],
  "clientUuid": "32990000-9900-4990-8990-329900009900"
}
```

5.6 Matriz por tipo de comprobante

Tipo	Comprador	ITBIS	Retención	Referencia	Notas
31	RNC obligatorio	Sí	—	—	Crédito fiscal
32	Opcional (< 250K); RNC obligatorio desde 250K	Sí	—	—	Consumo
33	Igual que el original	Sí	—	**Sí** (`referenceEncf`)	Nota de Débito

Tipo	Comprador	ITBIS	Retención	Referencia	Notas
34	Igual que el original	Sí	—	**Sí** (`referenceEncf`)	Nota de Crédito
41	RNC del proveedor	Sí	**ITBIS/ISR**	—	El emisor es el comprador
43	Sin comprador	No	—	—	Gasto menor
44	RNC y razón social	**Exento**	—	—	Régimen especial
45	RNC gubernamental	Sí	—	—	Venta al Estado
46	**Extranjero** (`foreignId`, `country`)	0 %	—	—	Exportación en divisa
47	**Extranjero** (beneficiario)	Exento	**ISR**	—	Pago al exterior

5.7 Estado de soporte de campos por tipo

Campo	Para	Estado
`buyer`, `items`, `payments`, totales	31, 32, 33, 34, 44, 45	**Soportado y en producción**
`referenceEncf`, `referenceDate`	33, 34	**Soportado**
`payments[].montoForeign`, `changeForeign`, `currencyIso`	Todos (impresión)	**Soportado** (solo representación impresa)
`withholdings` **(ITBIS/ISR retenido del documento)**	41, 47	**Soportado**: `TotalITBISRetenido` / `TotalISRRetencion`
`items[].withholding` **(retención por línea)**	41, 47	**Soportado**: `Retencion` por ítem
`buyer.foreignId`, `buyer.country`	46, 47	**Soportado**: `IdentificadorExtranjero` + `PaisOrigen`
`currency` (`currencyIso`, `exchangeRate`, `totalForeign`)	46	**Soportado**: sección `OtraMoneda`
`emisorOverride` **(dirección fiscal por venta)**	Todos	**Soportado**

Antes de usar 41, 46 y 47 en producción, valida una emisión real en ambiente `test`: la estructura está implementada, pero cada tipo tiene reglas fiscales que conviene confirmar con tus datos reales. Los tipos 31, 32, 33, 34, 44 y 45 están en uso en producción.

5.8 Campos extendidos (41, 46, 47)

Retenciones (41, 47), a nivel de documento o por línea:

```
{
  "withholdings": { "itbisRetenido": 1800, "isrRetenido": 1000 },
  "items": [{ "lineNumber": 1, "...": "...",
              "withholding": { "itbisRetenido": 1800, "isrRetenido": 1000 } }]
}
```

Comprador extranjero (46, 47), en `buyer`:

```
{ "buyer": { "name": "FOREIGN BUYER INC", "foreignId": "US-123456", "country": "US" } }
```

Moneda del documento (46), sección `OtraMoneda`:

```
{ "currency": { "currencyIso": "USD", "exchangeRate": 58.0, "totalForeign": 1000.0 } }
```

Dirección del emisor por venta (opcional, todos los tipos):

```
{ "emisorOverride": { "address": "...", "municipio": "010100", "provincia": "010000" } }
```

Si no envías `emisorOverride`, se usa la dirección de la sucursal o la empresa. Los códigos de municipio y provincia son los del catálogo de la DGII (6 dígitos).

5.9 Respuesta de la emisión

`202 Accepted`: el documento fue recibido y firmado; la DGII responde después.

```
{
  "status": "signed",
  "documentId": 1234,
  "clientUuid": "5b3c8f0e-6a4d-5c1e-9f3a-2d7e1b0c4a91",
  "encf": "E320000000045",
  "companyRnc": "101010101",
  "branchCode": "001",
  "securityCode": "aBc123",
  "signatureDate": "16-10-2026 10:15:30",
  "sequenceExpiry": "31-12-2027"
}
```

Campo	Uso
`status`	Estado **técnico** al emitir: `signed` (firmado y en cola de envío) o `queued` (recibido, pendiente de firma). **No es la respuesta de la DGII** (sección 6).
`documentId`, `clientUuid`	Guárdalos: son tus llaves para consultar el estado.
`encf`	e-NCF asignado por la plataforma.

Campo	Uso
`securityCode`, `signatureDate`	Para la representación impresa y el QR.
`sequenceExpiry`	Vencimiento de la secuencia (algunos tipos lo imprimen).
`companyRnc`, `branchCode`	RNC y código de sucursal que emitió: úsalos como control de consistencia.

- Si reenvías un ticket ya emitido (mismo `posTicketId`), recibes `200` con el documento existente y `"alreadyExisted": true`. No se consume otra secuencia.
- La emisión **no devuelve** `trackId`, URL de QR, `errors` ni `warnings`. El `trackId` y el resultado de la DGII se obtienen al consultar el estado.

06 Estados y consulta del resultado DGII

6.1 Estado técnico frente a estado DGII

La emisión es **asíncrona**: la plataforma firma y encola el documento, y la DGII responde después. El campo `status` recorre este ciclo:

`status`	Qué significa	¿Final?
`queued`	Recibido; pendiente de firma (por ejemplo, la empresa aún no ha cargado su certificado)	No
`signed`	Firmado; en cola para enviarse a la DGII	No
`sent`	Enviado a la DGII; esperando respuesta (ya tiene `trackId`)	No
`accepted`	**Aceptado** por la DGII	**Sí**
`accepted_conditional`	**Aceptado con observaciones** (fiscalmente válido; revisa los mensajes)	**Sí**
`rejected`	**Rechazado** por la DGII (motivo en `dgiiResponse.mensajes`)	**Sí**
`error`	No llegó a la DGII o agotó reintentos; la plataforma reintenta y, si persiste, hay que contactar a soporte	No (requiere atención)
`cancelled`	Anulado	Sí

6.2 Respuesta de la DGII (`dgiiResponse`)

```
{ "codigo": "4", "estado": "Aceptado Condicional", "mensajes": [
    { "codigo": "1934", "valor": "El campo MontoGravadoI1 del área Totales ... no coincide ..." } ]
}
```

- `codigo`: `1` Aceptado · `2` Rechazado · `3` En proceso · `4` Aceptado condicional.
- `mensajes[]`: motivos u observaciones exactos de la DGII (vacío si aceptó sin observaciones).

6.3 Cómo consultar

No hay webhooks de salida: **tu sistema consulta el estado** (sondeo). Usa el identificador que prefieras:

```
GET /api/v1/ecf/uuid/{clientUuid}    # por tu UUID de operación (recomendado)
GET /api/v1/ecf/{documentId}         # por id de documento
GET /api/v1/ecf/track/{trackId}      # por TrackID de la DGII (existe tras el envío)
GET /api/v1/ecf?companyId=…&posTicketId=…&limit=…   # listado filtrable
```

Las mismas rutas existen bajo `/api/v1/integrator/ecf/...`.

Campo de la respuesta	Para qué
`status`	Estado actual (6.1). Termina el sondeo con `accepted`, `accepted_conditional` o `rejected`.
`encf`, `securityCode`	Comprobante y código de seguridad.
`trackId`	TrackID de la DGII.
`dgiiResponse.codigo`, `estado`, `mensajes[]`	Respuesta exacta de la DGII.
`environment`	`test`, `cert` o `prod`.

El resto de los campos puede cambiar y no forma parte del contrato.

6.4 Frecuencia de consulta recomendada

Tiempos observados en producción (referencia, **no es un compromiso de nivel de servicio**):

Documento	Tiempo observado
Consumo (32) menor de RD$250,000 (resumen RFCE)	Segundos
e-CF íntegros (31, 33, 34, 41, 44, 45, 46, 47)	Unos 30–60 segundos

Tramo	Frecuencia
Primeros 30 segundos	Cada ~5 s
Hasta 5 minutos	Cada 15–30 s
A los 10 minutos sin estado final	Contacta a soporte con el `documentId`

Respeta el límite de **30 peticiones por minuto** (emisión y consultas suman). Si lo superas, la API responde `429`.

6.5 Si perdiste la respuesta de la emisión

Ante un timeout o un corte de red, **no emitas con otro** `posTicketId`. Reenvía **la misma** petición (mismo `posTicketId` y `clientUuid`): la plataforma devuelve el documento ya creado con `"alreadyExisted": true` y su estado actual. Así recuperas el `documentId` y continúas el sondeo.

6.6 Control de consistencia

La consulta por `trackId` incluye `companyId` y `companyRnc`:

```
{
  "trackId": "5acbc27c-397d-4476-9626-242d16a925a3",
  "encf": "E450000000005",
  "status": "accepted",
  "companyId": "mi-empresa",
  "companyRnc": "101010101",
  "dgiiResponse": { "estado": "Aceptado", "codigo": "1", "mensajes": [] }
}
```

Compara `companyId` y `companyRnc` con la empresa para la que creías facturar. Si no coinciden, detén el flujo y revisa tu mapeo. La respuesta de emisión trae `companyRnc` y `branchCode`, así que puedes cuadrar el RNC de punta a punta.

6.7 Acceso y aislamiento

- Todas las consultas exigen credencial: token del POS o API key con scope `ecf:emit`. Sin credencial → `401`.
- Solo ves documentos de las empresas y sucursales autorizadas a tu credencial. Un documento ajeno responde `404`, igual que uno inexistente.

6.8 QR de la representación impresa

La API no devuelve una URL de QR. Si tu sistema imprime el QR, arma la URL de la DGII con los datos de la respuesta:

Caso	URL base	Parámetros
Consumo (32) menor de RD$250,000	`https://fc.dgii.gov.do/eCF/ConsultaTimbreFC`	`RncEmisor`, `ENCF`, `MontoTotal`, `CodigoSeguridad`
Resto de tipos (incluido 32 desde RD$250,000)	`https://ecf.dgii.gov.do/eCF/ConsultaTimbre`	`RncEmisor`, `RncComprador`, `ENCF`, `FechaEmision`, `MontoTotal`, `FechaFirma`, `CodigoSeguridad`

En ambiente `test`, el host cambia al de TesteCF de la DGII.

> **Regla de oro: un QR o un código de seguridad NO significan aceptación fiscal.** Un documento puede estar firmado, con QR y código, y aun así ser rechazado. La aceptación se confirma **solo** cuando `status` es `accepted` o `accepted_conditional`.

07 Idempotencia (evitar duplicados)

La plataforma elimina duplicados por **empresa + sucursal + ambiente + tipo +** `posTicketId`. Si reenvías el mismo ticket (timeout, reintento o reinstalación de tu cliente), se reconoce el reintento y **no** se crea un comprobante duplicado ante la DGII.

Para máxima seguridad, genera un `clientUuid` **determinístico** por ticket. Por ejemplo, un UUID v5 calculado a partir de empresa, sucursal y número de ticket: el mismo ticket produce siempre el mismo UUID.

08 Notas de Crédito y Débito (33/34)

Para anular o ajustar un comprobante emitido, envía el tipo 33 o 34 con la referencia al comprobante afectado:

```
{
  "companyId": "mi-empresa",
  "branchId": "001",
  "ecfType": "34",
  "referenceEncf": "E320000000045",
  "referenceDate": "16-10-2026",
  "...": "resto igual que una emisión normal"
}
```

- `referenceDate` **(FechaNCFModificado):** la DGII exige que coincida exactamente con la fecha de emisión del original. Si el comprobante referenciado se emitió por e-Fiscal Cloud para tu empresa, la plataforma **deriva la fecha correcta del original** e ignora la que envíes si no coincide. Envía siempre la fecha más correcta que tengas: es el respaldo cuando el original no está en la plataforma.
- El monto de una nota de crédito no puede exceder el balance del comprobante que modifica. Para anular, el total de la nota debe igualar el del original.
- Las notas requieren un comprobante **ya aceptado**. Si el `referenceEncf` fue rechazado, la nota se rechaza.
- El `IndicadorNotaCredito` (0 = dentro de 30 días, 1 = después) lo calcula la plataforma.

09 Compras recibidas (e-CF de proveedores)

Con los scopes `purchases:read` y `purchases:approve`, tu sistema consulta los e-CF que los proveedores emitieron a tus empresas y les da aprobación comercial.

```
GET  /api/v1/integrator/purchases?estado=pendiente&limit=50&offset=0   # listar
GET  /api/v1/integrator/purchases/{encf}                               # detalle + XML
POST /api/v1/integrator/purchases/{encf}/approve                       # aprobar
POST /api/v1/integrator/purchases/{encf}/reject  { "motivo": "..." }   # rechazar
```

Filtro	Valores
`estado`	`pendiente`, `aprobado`, `rechazado`
`ecfType`	Tipo de e-CF
`environment`	`prod` (por defecto), `cert`, `test`
`limit` / `offset`	Paginación (máximo 200)

- e-Fiscal Cloud recibe los e-CF de la DGII por ti; tu sistema los consulta (sondeo).
- Al aprobar o rechazar, la plataforma firma automáticamente el acuse (ACECF) con el certificado de la empresa compradora.

10 Contador de documentos (scope `stats:read`)

Devuelve la cantidad y el monto de comprobantes por tipo, sumando ventas (emitidos) y compras (recibidos), en un rango de fechas y acotado a las empresas de la key.

```
GET /api/v1/integrator/stats/documents?from=2026-10-01&to=2026-10-31
Authorization: Bearer ik_live_xxxxxxxx
```

Parámetro	Valores	Por defecto
`from`	`YYYY-MM-DD` (inclusive)	—
`to`	`YYYY-MM-DD` (inclusive)	—
`environment`	`prod`, `cert`, `all`	`prod`
`kind`	`all`, `sales`, `purchases`	`all`
`ecfType`	`31` … `47`	—

Respuesta (valores ilustrativos):

```
{
  "range": { "from": "2026-10-01", "to": "2026-10-31", "environment": "prod", "kind": "all" },
  "rows": [
    { "ecfType": "31", "name": "Factura de Crédito Fiscal Electrónica",
      "salesCount": 120, "salesTotal": 150000.00,
      "purchasesCount": 60, "purchasesTotal": 900000.00,
      "count": 180, "total": 1050000.00 },
    { "ecfType": "32", "name": "Factura de Consumo Electrónica",
      "salesCount": 3500, "salesTotal": 2800000.00,
      "purchasesCount": 0, "purchasesTotal": 0,
      "count": 3500, "total": 2800000.00 }
  ],
  "totals": { "salesCount": 3620, "purchasesCount": 60, "count": 3680,
              "salesTotal": 2950000.00, "purchasesTotal": 900000.00, "total": 3850000.00 }
}
```

11 Consulta de RNC/Cédula (scope `rnc:lookup`)

Consulta el padrón de contribuyentes de la DGII cargado en la plataforma para obtener la razón social y el estado de un RNC o cédula, o buscar por nombre. Úsalo para autocompletar el comprador antes de emitir y evitar rechazos por RNC inválido.

Consulta exacta (9 u 11 dígitos):

```
GET /api/v1/integrator/rnc/101010101
```

```
{
  "found": true,
  "contribuyente": {
    "rnc": "101010101",
    "razonSocial": "EMPRESA DEMOSTRACION SRL",
    "estado": "ACTIVO",
    "actividadEconomica": "..."
  }
}
```

Si no existe, responde `404` con `{ "found": false }`.

Búsqueda por nombre o prefijo de RNC:

```
GET /api/v1/integrator/rnc/search?q=DEMOSTRACION&limit=10
```

```
{ "results": [{ "rnc": "101010101", "razonSocial": "EMPRESA DEMOSTRACION SRL", "estado": "ACTIVO"
}],
  "count": 1, "query": "DEMOSTRACION" }
```

`q` requiere al menos 2 caracteres; `limit` es 20 por defecto (máximo 50).

12 Errores frecuentes

12.1 Errores HTTP de la API

HTTP	Mensaje (resumen)	Causa	Qué hacer
400	`Invalid payload` + `details`	El cuerpo no cumple el esquema (campo faltante o formato inválido, por ejemplo un RNC sin 9 u 11 dígitos)	Corrige según `details`; no reintentes en bucle
400	`Esta API key exige identificar la empresa…`	Key nueva sin `companyId`	Envía `companyId` + `branchId` (código), sección 4
401	`API key de integrador inválida o ausente` / `revocada o inválida`	Falta el encabezado, el prefijo no es `ik_live_` o la key fue revocada	Revisa `Authorization: Bearer ik_live_…`. Un token `ecft_…` no sirve en `/integrator`
403	`La API key no tiene el permiso requerido: ecf:emit`	La key no tiene ese scope	Solicita una key nueva con el scope (sección 3.4)
403	`La sucursal pertenece a otra marca (tenant)`	Mezclaste id global con código, o la empresa no está en tu alcance	Usa `companyId` + código de sucursal (sección 4)
403	`La API key no está autorizada para esta empresa / sucursal`	La key está acotada a otras empresas	Solicita una key que incluya esa empresa
404	`No existe la sucursal con código "…" en la empresa "…"`	`companyId` o `branchId` incorrectos	Consulta `GET /api/v1/integrator/branches`
404	`Document not found` (consulta)	El documento no existe o pertenece a otra empresa (no se distingue)	Verifica el identificador y el alcance de tu credencial
409	`No active sequence for type XX …` / `Sequence … exhausted`	La empresa no tiene secuencia activa o agotó el rango de ese tipo	Cargar o activar secuencias en el portal
429	`Limite de emisiones alcanzado`	Más de 30 peticiones por minuto	Espacia los envíos y reintenta

12.2 Observaciones y rechazos de la DGII

Llegan en `dgiiResponse.mensajes[]` al consultar el estado.

Código DGII	Mensaje (resumen)	Causa habitual	Prevención
1385	`RNCComprador no es válido`	RNC del comprador mal digitado o inexistente (queda Aceptado condicional)	Valida el RNC con la sección 11 antes de emitir
1934	`MontoGravadoI1 no coincide con la sumatoria del detalle gravado`	Líneas con precio bruto (ITBIS incluido) o redondeo distinto al del encabezado	Envía líneas con **precio neto** y `subtotal` / `taxItbis` cuadrados con ellas
11014	`TotalITBIS1 no coincide con MontoGravado1 × tasa`	Misma raíz que 1934	Igual que arriba
—	`FechaNCFModificado no coincide`	Fecha de referencia distinta a la del original	Envía el `referenceEncf` correcto (sección 8)
—	`Certificate not found` / documento en `queued`	La empresa no tiene certificado digital cargado	Subir el certificado en el portal

Un Aceptado condicional es fiscalmente válido, pero deja observaciones: corrige el origen del dato para que no se repita.

13 Reglas fiscales aprendidas en pruebas con la DGII

1. **Fecha de vencimiento de la secuencia:** obligatoria para 31, 33, 41, 43, 44, 45, 46 y 47; prohibida en 32 y 34. La plataforma la exige al crear la secuencia y la transmite sola; si está desalineada con la autorización de la DGII, el documento se rechaza.
2. **Notas de crédito (34):** el monto no puede exceder el balance del original; para anular, el total debe igualarlo.
3. **Las notas (33/34) requieren un comprobante ya aceptado.**
4. **Comprador por tipo:** 31, 41 y 45 exigen RNC y razón social; 32 solo desde RD$250,000; 44 exige razón social; 46 y 47 usan identificador extranjero; 43 no lleva comprador.
5. **Exportación (46)** va como gravado a tasa 0 %, **no** como exento; el 47 (pagos al exterior) sí va exento.
6. **Las secuencias son finitas:** solicita nuevos rangos antes de que se agoten.
7. **Un RNC inexistente no bloquea la emisión**, pero el comprobante queda Aceptado condicional (código 1385).

Registro de validación (20 de octubre de 2026): los 10 tipos de comprobante (31, 32, 33, 34, 41, 43, 44, 45, 46 y 47) se emitieron contra TesteCF mediante API key de integrador y fueron aceptados. Es un registro de esa prueba; el estado vigente campo por campo está en la sección 5.7.

14 Buenas prácticas y salida a producción

14.1 Buenas prácticas

- **Envía la venta ya cerrada**, con totales y formas de pago finales.
- **Un ticket = un `posTicketId` estable**; reutilízalo en los reintentos.
- **Ante un 4xx, no reintentes en bucle:** corrige y reenvía. Ante 5xx o timeout, reintenta con espera creciente (backoff); la idempotencia te protege.
- **Comprador obligatorio** para 31, 44 y 45, y para 32 desde RD$250,000.
- **Montos en DOP.** Si cobras en divisa, convierte y, si quieres, envía la moneda original en los campos `*Foreign` solo para la impresión.
- **Confirma la aceptación por `status`**, nunca por el QR o el código de seguridad.

14.2 Lista de verificación antes de pasar a producción

- ☐ Credencial verificada (`GET /api/v1/integrator/verify` o token del POS activo).
- ☐ Mapeo de empresas y sucursales construido desde `GET /api/v1/integrator/branches`.
- ☐ Emisión con Forma A (`companyId` + código de sucursal) si la key es nueva.
- ☐ `posTicketId` estable y `clientUuid` determinístico por ticket.
- ☐ Prueba de reenvío: el mismo ticket devuelve `"alreadyExisted": true`.
- ☐ Sondeo de estado implementado hasta `accepted`, `accepted_conditional` o `rejected`.
- ☐ Manejo de `rejected` y `error` con aviso al usuario y registro del `documentId`.
- ☐ Líneas con precio neto y totales cuadrados (sin observaciones 1934/11014 en `test`).
- ☐ Validación de RNC del comprador antes de emitir (si tienes `rnc:lookup`).
- ☐ Una emisión aceptada en `test` por cada tipo de e-CF que vas a usar.
- ☐ Certificado digital y secuencias activas en la empresa.
- ☐ Límite de 30 peticiones por minuto respetado.
- ☐ Credenciales guardadas en un gestor de secretos.
- ☐ Ambiente de la sucursal cambiado a `prod` en el portal.

15 Contrato OpenAPI y ejemplos de código

El contrato formal está en `openapi_efiscalcloud.yaml`: rutas, esquemas y seguridad. Puedes importarlo en Postman, Insomnia o un generador de clientes.

Los ejemplos usan una API key de integrador con la Forma A y un `clientUuid` determinístico.

cURL

```
curl -X POST "https://api.efiscalcloud.com/api/v1/integrator/ecf/emit" \
  -H "Authorization: Bearer $EFC_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "companyId": "mi-empresa",
    "branchId": "001",
    "ecfType": "32",
    "posTicketId": "12345",
    "items": [{ "lineNumber":1, "productName":"Producto A", "quantity":1,
                "unitPrice":100, "itbisAmount":18, "itbisRate":18, "subtotal":100 }],
    "subtotal": 100, "taxItbis": 18, "total": 118,
    "payments": [{ "formaPago":"1", "monto":118 }],
    "clientUuid": "11111111-1111-4111-8111-111111111111"
  }'
```

Python

```python
import os, time, uuid, requests

API = "https://api.efiscalcloud.com"
KEY = os.environ["EFC_API_KEY"]
H = {"Authorization": f"Bearer {KEY}"}

company, branch, ticket = "mi-empresa", "001", "12345"
# Mismo ticket → mismo UUID (idempotencia de extremo a extremo)
client_uuid = str(uuid.uuid5(uuid.NAMESPACE_URL, f"{company}:{branch}:{ticket}"))

payload = {
    "companyId": company, "branchId": branch,
    "ecfType": "32", "posTicketId": ticket,
    "items": [{"lineNumber": 1, "productName": "Producto A", "quantity": 1,
               "unitPrice": 100, "itbisAmount": 18, "itbisRate": 18, "subtotal": 100}],
    "subtotal": 100, "taxItbis": 18, "total": 118,
    "payments": [{"formaPago": "1", "monto": 118}],
    "clientUuid": client_uuid,
}
r = requests.post(f"{API}/api/v1/integrator/ecf/emit", json=payload, headers=H, timeout=30)
r.raise_for_status()
print(r.status_code, r.json())

# Sondeo del estado DGII (sección 6.4)
FINAL = {"accepted", "accepted_conditional", "rejected"}
for wait in [5] * 6 + [20] * 13:
    time.sleep(wait)
    doc = requests.get(f"{API}/api/v1/ecf/uuid/{client_uuid}", headers=H, timeout=30).json()
    if doc["status"] in FINAL:
        print(doc["status"], doc.get("dgiiResponse"))
        break
```

C

```
using System.Net.Http;
using System.Net.Http.Json;

var http = new HttpClient { BaseAddress = new Uri("https://api.efiscalcloud.com") };
http.DefaultRequestHeaders.Authorization =
    new System.Net.Http.Headers.AuthenticationHeaderValue(
        "Bearer", Environment.GetEnvironmentVariable("EFC_API_KEY"));

var payload = new {
    companyId = "mi-empresa",
    branchId = "001",
    ecfType = "32",
    posTicketId = "12345",
    items = new[] { new {
        lineNumber = 1, productName = "Producto A", quantity = 1,
        unitPrice = 100m, itbisAmount = 18m, itbisRate = 18, subtotal = 100m } },
    subtotal = 100m, taxItbis = 18m, total = 118m,
    payments = new[] { new { formaPago = "1", monto = 118m } },
    // Usa un UUID determinístico derivado del ticket en tu sistema
    clientUuid = "11111111-1111-4111-8111-111111111111"
};

var resp = await http.PostAsJsonAsync("/api/v1/integrator/ecf/emit", payload);
Console.WriteLine((int)resp.StatusCode);
Console.WriteLine(await resp.Content.ReadAsStringAsync());
```

PHP

```php
<?php
$api = "https://api.efiscalcloud.com";
$key = getenv("EFC_API_KEY");

$payload = [
  "companyId" => "mi-empresa",
  "branchId" => "001",
  "ecfType" => "32",
  "posTicketId" => "12345",
  "items" => [[ "lineNumber"=>1, "productName"=>"Producto A", "quantity"=>1,
                "unitPrice"=>100, "itbisAmount"=>18, "itbisRate"=>18, "subtotal"=>100 ]],
  "subtotal" => 100, "taxItbis" => 18, "total" => 118,
  "payments" => [[ "formaPago"=>"1", "monto"=>118 ]],
  "clientUuid" => "11111111-1111-4111-8111-111111111111",
];

$ch = curl_init("$api/api/v1/integrator/ecf/emit");
curl_setopt_array($ch, [
  CURLOPT_POST => true,
  CURLOPT_RETURNTRANSFER => true,
  CURLOPT_HTTPHEADER => ["Authorization: Bearer $key", "Content-Type: application/json"],
  CURLOPT_POSTFIELDS => json_encode($payload),
]);
$res = curl_exec($ch);
echo curl_getinfo($ch, CURLINFO_HTTP_CODE) . "\n" . $res . "\n";
```

16 Glosario

Término	Definición
e-CF	Comprobante Fiscal Electrónico
e-NCF	Número del e-CF: `E` + 2 dígitos de tipo + 10 de secuencia
TrackID	Identificador que asigna la DGII al recibir el comprobante
RFCE	Resumen de Factura de Consumo (canal del tipo 32 por debajo de RD$250,000)
ACECF	Aprobación Comercial del e-CF (respuesta firmada al aprobar o rechazar una compra)
Scope	Permiso de una API key
Token `ecft_…`	Credencial del POS por sucursal (sección 3.2)
API key `ik_live_…`	Credencial del integrador (sección 3.3)
Forma A / Forma B	Maneras de identificar la sucursal al emitir (sección 4)

17 Soporte

Canal	Detalle
Soporte técnico	soporte@efiscalcloud.com
Comercial y nuevas integraciones	info@efiscalcloud.com
WhatsApp	+1 849-379-4444 (lunes a viernes, 8:00 a. m. – 6:00 p. m.)
Web	efiscalcloud.com/integraciones

Al reportar un caso, incluye: `documentId` o `clientUuid`, `companyId`, código de sucursal, ambiente, hora de la emisión y el `dgiiResponse` recibido. **Nunca envíes tu API key por correo ni por chat.**

e-Fiscal Cloud · Plataforma de infraestructura fiscal digital · Una solución de KarmaTech Consulting. Calle César Nicolás Penson No. 70, Edif. Caromang, Suite 205, Santo Domingo, República Dominicana. Documento para socios de integración. Las marcas de terceros mencionadas pertenecen a sus respectivos titulares.

e-FiscalCloud
TU NEGOCIO CONECTADO CON LA NUEVA ERA FISCAL

Novedades del contrato de integración v1.2

Complemento del Manual de Integración v1.1 y del OpenAPI · noviembre de 2026

Todos los cambios son **aditivos**: ninguna integración existente deja de funcionar. Todo lo descrito está vigente en la API; el contrato completo está en `openapi_efiscalcloud_v1.2.yaml`.

1. Sucursales sin ambigüedad (§4 del manual)

Cambio	Detalle	Estado
`GET /integrator/branches`	Filtros `companyId` y `environment`; nunca devuelve sucursales de otra marca. `branchCode` es el valor que `branchId` acepta junto con `companyId`.	Vigente
`GET /ecf/branches`	Sucursal del token `ecft_…` del POS, mismo esquema.	Vigente
Campo `active`	Indica si la sucursal puede emitir; permite mostrar un selector solo con sucursales válidas.	Vigente

2. Errores accionables

errorCode	Cuándo
`BRANCH_NOT_FOUND`	El código no existe en la empresa indicada (404).
`COMPANY_ID_REQUIRED`	La key exige `companyId` y no se envió (400).
`BRANCH_NOT_IN_COMPANY`	La sucursal existe pero no pertenece a `companyId` (404).
`CREDENTIAL_SCOPE`	Falta el scope o el acceso a la empresa/sucursal (403).
`RATE_LIMITED`	Más de 30 peticiones por minuto (429) con cabecera `Retry-After`; reintenta con el mismo `clientUuid`.

El cuerpo de error incluye `error` (texto) y `errorCode` (estable).

3. Consulta de estado con identidad completa (§6)

`GET /ecf/uuid/{clientUuid}` es el endpoint recomendado y estable. El e-NCF se repite entre empresas, por lo que `EcfDocument` añade `companyId`, `branchId` y `clientUuid`, y separa `processingStatus` (técnico: queued,

signed, sent) de `dgiiStatus` (fiscal: accepted, accepted_conditional, rejected). `status` se conserva por compatibilidad. Aplica a la consulta por uuid, por id y al listado.

4. Sistemas con integración evaluada

Puntos de venta · Connection Key por sucursal
ICG FrontRest · ICG FrontRetail · Ágora POS · PixelPoint

ERP y contabilidad · API Key de integrador
ICG Software · ICG Manager · Odoo · QuickBooks

Alcance de cada integración sujeto a evaluación técnica. Las marcas pertenecen a sus respectivos titulares.