Academia

# Casos prácticos

Anonymous · 05/11/2026

# Tabla de contenidos

**Casos prácticos** ..... 3
- ***Factura de principio a fin*** ..... 4
- ***Nota de crédito*** ..... 11
- ***Guía de despacho*** ..... 15
- ***Factura exenta e ítems exentos*** ..... 18
- ***Boleta de principio a fin*** ..... 20

# Casos prácticos

Anonymous · 05/11/2026

## Casos prácticos

Última actualización el 05/11/2026

## Factura de principio a fin

Un programa completo en Python que construye, valida, envía al SII, consulta el estado y genera el PDF de una factura electrónica.

Anonymous · 05/11/2026

# Factura de principio a fin

En este caso juntamos los seis pasos en un solo programa. Emite una factura afecta (`33`) en el **ambiente de certificación** del SII.

## Lo que necesitas

| Archivo o dato | Dónde se explica |
|---|---|
| Token de API Gateway | Primeros pasos |
| Certificado digital en PEM: `cert.pem` y `key.pem` | El certificado digital |
| CAF de facturas (`33`) del ambiente de certificación | Folios y CAF |
| Fecha de la resolución en certificación | Resolución de autorización |

**¿Aún no tienes CAF de certificación?**

Puedes probar los pasos 1, 2, 3 y 6 con un CAF y un certificado de prueba. Te explicamos cómo en Probar sin el SII.

## El programa

Instala `requests` (`pip install requests`) y define la variable de entorno `API_TOKEN`. Deja en la misma carpeta los archivos `cert.pem`, `key.pem` y `caf_33.xml`.

```
"""Emite una factura electrónica en el ambiente de certificación del SII."""
import base64
import os
import time

import requests

API_URL = 'https://app.apigateway.cl/api/v2/libredte'
HEADERS = {
    'Authorization': f'Token {os.environ["API_TOKEN"]}',
```

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

AMBIENTE = 1  # 1: certificación, 0: producción.
SII_RUT = '60803000-K'

# Estados del envío que ya no cambian (ver el paso 5). Los demás significan
# que el SII sigue procesando, o que la consulta falló y hay que repetirla.
ESTADOS_FINALES = {'EPR', 'LOK', 'RPR', 'RLV', 'RCH', 'RCO', 'RFR', 'RSC', 'RCT',
'RPT', 'VOF'}
ESTADOS_ACEPTADOS = {'EPR', 'LOK', 'RPR', 'RLV'}


def libredte(operacion: str, parameters: dict) -> dict | list | None:
    """Ejecuta una operación de LibreDTE y entrega el contenido de `data`."""
    response = requests.post(
        f'{API_URL}/{operacion}',
        json={'parameters': parameters},
        headers=HEADERS,
        timeout=120,
    )
    if response.status_code != 200:
        try:
            detalle = response.json().get('detail', response.text)
        except ValueError:  # La respuesta de error no viene en JSON.
            detalle = response.text
        raise RuntimeError(f'{operacion}: HTTP {response.status_code}: {detalle}')
    return response.json()['data']


def leer_texto(ruta: str) -> str:
    """Lee un archivo de texto, como el certificado o la llave en PEM."""
    with open(ruta, encoding='ascii') as archivo:
        return archivo.read()


def leer_base64(ruta: str) -> str:
    """Lee un archivo y lo entrega codificado en base64."""
    with open(ruta, 'rb') as archivo:
        return base64.b64encode(archivo.read()).decode()


def guardar_base64(ruta: str, contenido: str) -> None:
    """Decodifica un base64 y lo guarda como archivo, sin tocar sus bytes."""
    with open(ruta, 'wb') as archivo:
        archivo.write(base64.b64decode(contenido))


# Datos que normalmente vienen de tu base de datos.
CERTIFICADO = {
    'certificate': leer_texto('cert.pem'),
```

```
    'privateKey': leer_texto('key.pem'),
}
CAF = leer_base64('caf_33.xml')
EMISOR = {
    'rut': '88888888-8',
    'razon_social': 'Mi Empresa SpA',
    'autorizacion_dte': {
        'fecha_resolucion': '2025-11-20',  # Fecha de autorización en certificación.
        'numero_resolucion': 0,            # En certificación siempre es 0.
    },
}
FOLIO = 1  # El siguiente folio libre del CAF, según tu control de folios.

# Paso 1: construir el DTE. Primero, los datos del documento.
datos = {
    'Encabezado': {
        'IdDoc': {'TipoDTE': 33, 'Folio': FOLIO, 'FmaPago': 1},
        'Emisor': {
            'RUTEmisor': EMISOR['rut'],
            'RznSoc': EMISOR['razon_social'],
            'GiroEmis': 'Venta de software',
            'Acteco': 620100,
            'DirOrigen': 'Av. Siempre Viva 742',
            'CmnaOrigen': 'Providencia',
        },
        'Receptor': {
            'RUTRecep': '77777777-7',
            'RznSocRecep': 'Cliente Ejemplo Ltda.',
            'GiroRecep': 'Comercio',
            'DirRecep': 'Calle Falsa 123',
            'CmnaRecep': 'Santiago',
        },
    },
    'Detalle': [
        {'NmbItem': 'Servicio de desarrollo', 'QtyItem': 1, 'PrcItem': 100000},
        {'NmbItem': 'Soporte técnico (horas)', 'QtyItem': 3, 'PrcItem': 25000},
    ],
}

# Con los datos, el CAF y el certificado se obtiene el DTE timbrado y firmado.
dte = libredte('billing/document/builder/build', {
    'bag': {'parsedData': datos, 'caf': CAF, 'certificate': CERTIFICADO},
})
xml_dte = dte['document_xml']
guardar_base64(f'dte_33_{FOLIO}.xml', xml_dte)
totales = dte['document_normalized']['Encabezado']['Totales']
print('Documento construido:', dte['document_id'], 'total', totales['MntTotal'])

# Paso 2: validar el esquema antes de enviar.
libredte('billing/document/validator/validateSchema', {'source': xml_dte})
```

```
# Paso 3: armar el sobre dirigido al SII.
sobre = libredte('billing/document/dispatcher/create', {
    'bag': {
        'xmlDocument': xml_dte,
        'certificate': CERTIFICADO,
        'emisor': EMISOR,
        'receptor': {'rut': SII_RUT},
    },
})
guardar_base64(f'sobre_sii_33_{FOLIO}.xml', sobre['xml'])

# Paso 4: enviar al SII.
request_sii = {'certificate': CERTIFICADO, 'options': {'environment': AMBIENTE}}
envio = libredte('billing/integration/sii_dte/sendXmlDocument', {
    'request': request_sii,
    'doc': sobre['xml'],
    'company': EMISOR['rut'],
})
track_id = envio['track_id']
print('Enviado al SII, Track ID:', track_id)  # ¡Guárdalo en tu base de datos!

# Paso 5: consultar el estado hasta tener un resultado final.
estado = None
for espera in [30, 60, 120, 300, 600]:
    time.sleep(espera)
    try:
        estado = libredte('billing/integration/sii_dte/checkXmlDocumentSentStatus', {
            'request': request_sii,
            'trackId': track_id,
            'company': EMISOR['rut'],
        })
    except RuntimeError as error:
        # La consulta falló (por ejemplo, el SII no respondió): se reintenta.
        print('Falló la consulta, se reintenta:', error)
        continue
    print('Estado:', estado['status'], estado['description'])
    if estado['status'] in ESTADOS_FINALES:
        break

resumen = estado['resume'] if estado else {}
if not estado or estado['status'] not in ESTADOS_FINALES:
    print('Sin estado final todavía. Vuelve a consultar más tarde con el Track ID',
track_id)
elif estado['status'] == 'RPT':
    print('El SII ya había recibido este sobre: consulta el estado con el Track ID del
primer envío.')
elif estado['status'] == 'VOF':
    print('El SII no encontró el archivo: envía de nuevo el mismo sobre (no lo
construyas de nuevo).')
elif estado['status'] in ESTADOS_ACEPTADOS and resumen.get('rejected', 0) == 0:
    resultado = 'aceptado con reparos' if resumen.get('repairs') else 'aceptado'
```

```
    print('Resultado:', resultado)

    # Paso 6: generar el PDF.
    pdf = libredte('billing/document/renderer/render', {
        'bag': {
            'xmlDocument': xml_dte,
            'emisor': EMISOR,  # Con la resolución, que se imprime en el PDF.
            'options': {'renderer': {'format': 'pdf', 'renderings': {'tributaria': 1,
'cedible': 1}}},
        },
    })
    for archivo in pdf['renderings']:
        guardar_base64(archivo['filename'], archivo['content'])
        print('PDF generado:', archivo['filename'])
else:
    # Para saber el motivo, pide el detalle por correo (ver el paso 5).
    print('El documento fue rechazado:', estado['status'], estado['description'])
```

## Probar sin el SII

Si todavía no tienes un CAF y un certificado reales del ambiente de certificación, puedes probar los pasos 1, 2, 3 y 6 con datos de prueba. Los pasos 4 y 5 no funcionan con ellos: el SII rechaza la autenticación y el programa se detiene en el envío.

1. **El certificado:** genera uno de prueba (ver Un certificado de prueba) y guarda `cert` como `cert.pem` y `pkey` como `key.pem`.
2. **El CAF:** genera uno de prueba y guárdalo como `caf_33.xml`. Ejecuta esto una vez, con las funciones `libredte()` y `guardar_base64()` del programa. El CAF viene en base64 en el campo `xml`, y `guardar_base64()` lo decodifica al guardarlo:

```
caf = libredte('billing/identifier/caf_faker/create', {
    'emisor': {'rut': '88888888-8', 'razon_social': 'Mi Empresa SpA'},
    'codigoDocumento': 33,
    'folioDesde': 1,
    'folioHasta': 100,
})
guardar_base64('caf_33.xml', caf['xml'])
```

3. **El programa:** borra todo lo que sigue a `# Paso 4: enviar al SII.` y pon en su lugar solo el paso 6, sin la condición de que el SII haya aceptado:

```
# Paso 6: generar el PDF, sin pasar por el SII.
pdf = libredte('billing/document/renderer/render', {
    'bag': {
        'xmlDocument': xml_dte,
        'emisor': EMISOR,
        'options': {'renderer': {'format': 'pdf', 'renderings': {'tributaria': 1,
```

```
'cedible': 1}}},
    },
})
for archivo in pdf['renderings']:
    guardar_base64(archivo['filename'], archivo['content'])
    print('PDF generado:', archivo['filename'])
```

El documento que obtienes no tiene validez tributaria: el CAF es ficticio. Sirve para revisar los datos, los totales y el PDF.

## Qué hace, en orden

| Paso | Llamada | Qué guarda |
|---|---|---|
| 1 | `builder/build` | `dte_33_1.xml`: el documento timbrado y firmado |
| 2 | `validator/validateSchema` | (nada; falla si el XML no es válido) |
| 3 | `dispatcher/create` con `receptor` del SII | `sobre_sii_33_1.xml` |
| 4 | `sii_dte/sendXmlDocument` | El Track ID |
| 5 | `sii_dte/checkXmlDocumentSentStatus` | El estado final |
| 6 | `renderer/render` | Los PDF tributario y cedible |

## Para llevarlo a tu sistema

Este programa hace todo seguido para que lo entiendas de una vez. En un sistema real conviene separarlo:

1. **Al emitir** (cuando el usuario confirma la venta): reservar el folio, construir, validar, armar el sobre, enviar y guardar el Track ID. Responder al usuario.
2. **En segundo plano** (un proceso programado): consultar el estado de los envíos pendientes y actualizar cada documento.
3. **Cuando el documento es aceptado**: generar el PDF y entregarlo al cliente. Si el cliente es una empresa que emite documentos electrónicos, enviarle además el XML a su correo de intercambio (ver Entregar el documento al receptor).

Así ningún usuario espera al SII, y si algo falla en un paso, sabes exactamente desde dónde continuar.

#### El mismo flujo en otros lenguajes

Son solo solicitudes HTTP con JSON. Puedes traducir este programa a PHP, JavaScript, Java, C# o cualquier otro lenguaje usando su cliente HTTP habitual. Si usas un asistente de IA, cópiale esta página (ver Cómo usar esta documentación).

Última actualización el 05/11/2026

## Nota de crédito

Cómo anular una factura, corregir un texto o corregir un monto con una nota de crédito, y cuándo usar una nota de débito.

Anonymous · 05/11/2026

# Nota de crédito

Un documento aceptado por el SII **no se puede modificar ni eliminar**. Si tiene un error, o si cambió algo después de la venta, se corrige emitiendo una **nota de crédito** (`61`) que hace referencia al documento original.

El flujo de emisión es exactamente el mismo de una factura (construir, sobre, enviar, estado, PDF). Lo único que cambia son los datos: el `TipoDTE` es `61`, necesitas un **CAF de notas de crédito** y el documento lleva una **referencia**.

## Caso 1: anular una factura completa

Emitiste la factura folio `1` del `2026-10-29` (la del programa Factura de principio a fin, con dos ítems) y la venta se canceló. La nota de crédito repite los ítems de la factura y la anula:

```
{
    "Encabezado": {
        "IdDoc": { "TipoDTE": 61, "Folio": 1 },
        "Emisor": {
            "RUTEmisor": "88888888-8",
            "RznSoc": "Mi Empresa SpA",
            "GiroEmis": "Venta de software",
            "Acteco": 620100,
            "DirOrigen": "Av. Siempre Viva 742",
            "CmnaOrigen": "Providencia"
        },
        "Receptor": {
            "RUTRecep": "77777777-7",
            "RznSocRecep": "Cliente Ejemplo Ltda.",
            "GiroRecep": "Comercio",
            "DirRecep": "Calle Falsa 123",
            "CmnaRecep": "Santiago"
        }
    },
    "Detalle": [
        { "NmbItem": "Servicio de desarrollo", "QtyItem": 1, "PrcItem": 100000 },
        { "NmbItem": "Soporte técnico (horas)", "QtyItem": 3, "PrcItem": 25000 }
```

```
    ],
    "Referencia": [
        {
            "TpoDocRef": 33,
            "FolioRef": 1,
            "FchRef": "2026-10-29",
            "CodRef": 1,
            "RazonRef": "Anula factura"
        }
    ]
}
```

Los totales de la nota (neto $175.000, IVA $33.250, total $208.250) quedan iguales a los de la factura: así el SII descuenta la venta completa.

**El folio es de la nota, no de la factura**

`IdDoc.Folio` es el folio de la **nota de crédito** (de su propio CAF). El folio de la factura que anulas va en `Referencia.FolioRef`, y su fecha de emisión (`FchEmis`) en `Referencia.FchRef`.

## Caso 2: corregir un texto

La factura salió con un error en un dato que no afecta los montos, por ejemplo el giro del receptor. La nota lleva `CodRef: 2` y un ítem que describe la corrección, **sin precio**: `"PrcItem": false` le indica a la API que el ítem no lleva precio.

```
"Detalle": [
    { "NmbItem": "Corrige giro del receptor", "QtyItem": 1, "PrcItem": false }
],
"Referencia": [
    {
        "TpoDocRef": 33,
        "FolioRef": 1,
        "FchRef": "2026-10-29",
        "CodRef": 2,
        "RazonRef": "DONDE DICE Comercio DEBE DECIR Comercio al por mayor"
    }
]
```

Los totales de esta nota quedan en `0`: no cambia ningún monto.

## Caso 3: corregir un monto

Cobraste de más: los 3 servicios de soporte eran a $20.000 y no a $25.000. La nota de crédito lleva solo la **diferencia** ($5.000 × 3) y `CodRef: 3`:

```
"Detalle": [
    { "NmbItem": "Soporte técnico (horas) - diferencia de precio", "QtyItem": 3,
"PrcItem": 5000 }
],
"Referencia": [
    {
        "TpoDocRef": 33,
        "FolioRef": 1,
        "FchRef": "2026-10-29",
        "CodRef": 3,
        "RazonRef": "Corrige precio del soporte técnico"
    }
]
```

Totales de la nota: neto $15.000, IVA $2.850, total $17.850.

## ¿Y la nota de débito?

La **nota de débito** (`56`) funciona igual, pero **aumenta** el monto: por ejemplo, si cobraste de menos, o para cobrar intereses. También se usa para **anular una nota de crédito** emitida por error:

```
"Referencia": [
    {
        "TpoDocRef": 61,
        "FolioRef": 1,
        "FchRef": "2026-10-29",
        "CodRef": 1,
        "RazonRef": "Anula nota de crédito"
    }
]
```

## Errores frecuentes

| Mensaje o situación | Causa |
|---|---|
| `Falta incluir documento de referencia.` | La nota no tiene `Referencia`. |
| El SII rechaza la nota | La factura referenciada no existe, tiene otra fecha, o es de otro receptor. Revisa `TpoDocRef`, `FolioRef` y `FchRef`. |
| `El CAF ... es del tipo de documento 33, pero el documento ... es del tipo 61.` | Usaste el CAF de facturas (`33`) en vez del de notas de crédito (`61`). |

### ¿Se puede revertir una anulación?

Sí. Una vez anulada con una nota de crédito, la factura ya no tiene efecto. Si después necesitas

“revivirla”, se emite una nota de débito que anula esa nota de crédito, como en el ejemplo anterior.

Última actualización el 05/11/2026

## Guía de despacho

Cómo emitir una guía de despacho: tipo de despacho, motivo del traslado, datos del transporte y su relación con la factura.

Anonymous · 05/11/2026

# Guía de despacho

La **guía de despacho** (`52`) acompaña a la mercadería mientras se traslada. Es obligatoria cuando los productos salen de tu bodega, aunque todavía no se hayan facturado (o aunque no sea una venta, como un traslado entre sucursales).

## Lo que la hace distinta

| Campo | Dónde | Qué indica |
|---|---|---|
| `IndTraslado` | `IdDoc` | **Obligatorio.** El motivo del traslado (venta, traslado interno, devolución, etc.). |
| `TipoDespacho` | `IdDoc` | Quién despacha: `1` el receptor, `2` el emisor a las instalaciones del cliente, `3` el emisor a otras instalaciones. |
| `Transporte` | `Encabezado` | Patente del vehículo, RUT del transportista, chofer y dirección de destino. |

Los motivos de `IndTraslado` son:

| Código | Motivo | Ejemplo |
|---|---|---|
| `1` | Operación constituye venta | Despachas hoy y facturas después |
| `2` | Ventas por efectuar | Mercadería que se venderá más adelante |
| `3` | Consignaciones | Productos que otra tienda venderá por ti |
| `4` | Entrega gratuita | Muestras o donaciones |
| `5` | Traslados internos | De la bodega central a una sucursal |
| `6` | Otros traslados no venta | Envío a reparación |
| `7` | Guía de devolución | El cliente te devuelve productos |
| `8` | Traslado para exportación (no venta) | |
| `9` | Venta para exportación | |

## Ejemplo: despacho de una venta

```
{
    "Encabezado": {
        "IdDoc": {
            "TipoDTE": 52,
            "Folio": 1,
            "TipoDespacho": 2,
            "IndTraslado": 1
        },
        "Emisor": {
            "RUTEmisor": "88888888-8",
            "RznSoc": "Mi Empresa SpA",
            "GiroEmis": "Venta de equipos computacionales",
            "Acteco": 465100,
            "DirOrigen": "Av. Siempre Viva 742",
            "CmnaOrigen": "Providencia"
        },
        "Receptor": {
            "RUTRecep": "77777777-7",
            "RznSocRecep": "Cliente Ejemplo Ltda.",
            "GiroRecep": "Comercio",
            "DirRecep": "Calle Falsa 123",
            "CmnaRecep": "Santiago"
        },
        "Transporte": {
            "Patente": "ABCD12",
            "RUTTrans": "88888888-8",
            "Chofer": {
                "RUTChofer": "11222333-9",
                "NombreChofer": "Pedro Pérez"
            },
            "DirDest": "Calle Falsa 123",
            "CmnaDest": "Santiago"
        }
    },
    "Detalle": [
        { "NmbItem": "Notebook 14 pulgadas", "QtyItem": 5, "PrcItem": 450000 }
    ]
}
```

Como `IndTraslado` es `1` (constituye venta), la guía lleva precios y la API calcula sus totales con IVA, igual que una factura.

#### Traslados que no son venta

Si el traslado no es una venta (por ejemplo `5`, traslado interno), no necesitas valorizar la mercadería: puedes enviar los ítems solo con nombre y cantidad.

# Facturar una o varias guías

Cuando la guía es por una venta (`IndTraslado: 1`), después tienes que emitir la **factura**. La factura hace referencia a la guía (o a varias guías, en una factura a fin de mes):

```
"Referencia": [
    {
        "TpoDocRef": 52,
        "FolioRef": 1,
        "FchRef": "2026-10-29",
        "RazonRef": "Factura guía de despacho"
    }
]
```

En las referencias de una factura no se usa `CodRef`: ese campo es solo para las notas de crédito y débito.

### Ejemplos de guías

La API incluye ejemplos para cada motivo de traslado: `052_guia_despacho/052_001_constituye_venta`, `052_002_traslado_interno`, `052_003_guia_devolucion`, `052_004_venta_por_efectuar`, `052_005_entrega_gratuita` y otros. Obtenlos con `billing/document/examples/get`.

---

Última actualización el 05/11/2026

## Factura exenta e ítems exentos

Cómo emitir una factura exenta de IVA y cómo incluir ítems exentos dentro de una factura afecta.

Anonymous · 05/11/2026

# Factura exenta e ítems exentos

Algunas ventas no pagan IVA (por ejemplo, ciertos servicios de educación, de salud o algunos arriendos). Para documentarlas hay dos caminos:

| Situación | Documento |
|---|---|
| **Toda** la venta es exenta | Factura exenta (`34`) |
| Una parte es afecta y otra exenta | Factura afecta (`33`) con algunos ítems marcados como exentos |

## Factura exenta (34)

Se arma igual que una factura, con `TipoDTE: 34` y los ítems marcados con `IndExe: 1`:

```
{
    "Encabezado": {
        "IdDoc": { "TipoDTE": 34, "Folio": 1 },
        "Emisor": {
            "RUTEmisor": "88888888-8",
            "RznSoc": "Mi Empresa SpA",
            "GiroEmis": "Capacitación",
            "Acteco": 855000,
            "DirOrigen": "Av. Siempre Viva 742",
            "CmnaOrigen": "Providencia"
        },
        "Receptor": {
            "RUTRecep": "77777777-7",
            "RznSocRecep": "Cliente Ejemplo Ltda.",
            "GiroRecep": "Comercio",
            "DirRecep": "Calle Falsa 123",
            "CmnaRecep": "Santiago"
        }
    },
    "Detalle": [
        { "IndExe": 1, "NmbItem": "Curso de capacitación", "QtyItem": 5, "PrcItem":
25000 }
    ]
}
```

La API calcula el total como monto exento: `MntExe` $125.000 y `MntTotal` $125.000, sin IVA.

## Ítems exentos dentro de una factura afecta

En una factura `33` puedes mezclar ítems afectos y exentos. Marca solo los exentos con `IndExe: 1`:

```
"Detalle": [
    { "NmbItem": "Licencia de software", "QtyItem": 1, "PrcItem": 100000 },
    { "IndExe": 1, "NmbItem": "Capacitación en el uso del software", "QtyItem": 1,
"PrcItem": 50000 }
]
```

La API calcula el IVA solo sobre la parte afecta: neto $100.000, IVA $19.000, exento $50.000 y total $169.000.

**Lo exento lo define la ley, no tu sistema**

Que una venta sea exenta depende de la ley y de la actividad del emisor. Si tienes dudas sobre qué ventas de tu cliente son exentas, consúltalo con su contador antes de configurar tu sistema.

**Recuerda el CAF**

La factura exenta (`34`) usa su propio CAF, distinto del de las facturas afectas (`33`).

Última actualización el 05/11/2026

## Boleta de principio a fin

Qué tienen de distinto las boletas y un programa completo en Python que construye, valida, envía al SII, consulta el estado y genera el PDF de una boleta electrónica.

Anonymous · 05/11/2026

# Boleta de principio a fin

La **boleta electrónica** (`39`) y la **boleta exenta** (`41`) son los documentos para ventas a **consumidores finales**. Siguen los mismos seis pasos que una factura, pero con dos diferencias importantes: algunos datos del documento cambian, y **el envío al SII y la consulta del estado usan otras operaciones**, porque el SII recibe las boletas por un servicio propio.

## Qué cambia respecto de una factura

| Aspecto | Factura (`33`) | Boleta (`39`) |
|---|---|---|
| Precio en el detalle (`PrcItem`) | Neto, sin IVA | **Con IVA incluido** |
| Receptor | Datos completos de la empresa | Opcionales. Si el cliente no da su RUT, se usa `66666666-6`. |
| `GiroRecep`, `Acteco`, `FmaPago` | Se envían | No se usan |
| CAF | De facturas (`33`) | De boletas (`39`, o `41` para la exenta) |
| Sobre (paso 3) | `EnvioDTE` | `EnvioBOLETA` (la API lo crea sola) |
| Enviar al SII (paso 4) | `sii_dte/sendXmlDocument` | **`sii_boleta/sendXmlDocument`** |
| Consultar el estado (paso 5) | `sii_dte/checkXmlDocumentSentStatus` | **`sii_boleta/checkXmlDocumentSentStatus`** |
| Verificar un documento | `/api/v2/sii/dte/emitidos/verificar` | **`sii_boleta/validateDocument`** |
| Copia cedible del PDF | Sí | No |

El resto es igual: `builder/build`, las validaciones, `dispatcher/create` y `renderer/render` reciben los

mismos parámetros. En cada paso, esto es lo que corresponde a las boletas:

| Paso | Dónde está |
|---|---|
| 1. Construir | 1.2. Boletas: los datos y el CAF |
| 2. Validar | Las mismas validaciones |
| 3. Armar el sobre | Un sobre por tipo de documento |
| 4. Enviar al SII | 4.2. Boletas |
| 5. Consultar el estado | 5.2. Boletas |
| 6. Generar el PDF | Copia tributaria |

## Lo que necesitas

| Archivo o dato | Dónde se explica |
|---|---|
| Token de API Gateway | Primeros pasos |
| Certificado digital en PEM: `cert.pem` y `key.pem` | El certificado digital |
| CAF de boletas (`39`) del ambiente de certificación | Folios y CAF |
| Fecha de la resolución en certificación | Resolución de autorización |

**¿Aún no tienes CAF de certificación?**

Puedes probar los pasos 1, 2, 3 y 6 con un CAF y un certificado de prueba. Te explicamos cómo en Probar sin el SII.

## El programa

Instala `requests` (`pip install requests`) y define la variable de entorno `API_TOKEN`. Deja en la misma carpeta los archivos `cert.pem`, `key.pem` y `caf_39.xml`.

```
"""Emite una boleta electrónica en el ambiente de certificación del SII."""
import base64
import os
import time

import requests

API_URL = 'https://app.apigateway.cl/api/v2/libredte'
HEADERS = {
    'Authorization': f'Token {os.environ["API_TOKEN"]}',
    'Content-Type': 'application/json',
}
```

```
AMBIENTE = 1  # 1: certificación, 0: producción.
SII_RUT = '60803000-K'

# Estados del envío que ya no cambian (ver el paso 5). Los demás significan
# que el SII sigue procesando, o que la consulta falló y hay que repetirla.
ESTADOS_FINALES = {'EPR', 'LOK', 'RPR', 'RLV', 'RCH', 'RCO', 'RFR', 'RSC', 'RCT',
'RPT', 'VOF'}
ESTADOS_ACEPTADOS = {'EPR', 'LOK', 'RPR', 'RLV'}


def libredte(operacion: str, parameters: dict) -> dict | list | None:
    """Ejecuta una operación de LibreDTE y entrega el contenido de `data`."""
    response = requests.post(
        f'{API_URL}/{operacion}',
        json={'parameters': parameters},
        headers=HEADERS,
        timeout=120,
    )
    if response.status_code != 200:
        try:
            detalle = response.json().get('detail', response.text)
        except ValueError:  # La respuesta de error no viene en JSON.
            detalle = response.text
        raise RuntimeError(f'{operacion}: HTTP {response.status_code}: {detalle}')
    return response.json()['data']


def leer_texto(ruta: str) -> str:
    """Lee un archivo de texto, como el certificado o la llave en PEM."""
    with open(ruta, encoding='ascii') as archivo:
        return archivo.read()


def leer_base64(ruta: str) -> str:
    """Lee un archivo y lo entrega codificado en base64."""
    with open(ruta, 'rb') as archivo:
        return base64.b64encode(archivo.read()).decode()


def guardar_base64(ruta: str, contenido: str) -> None:
    """Decodifica un base64 y lo guarda como archivo, sin tocar sus bytes."""
    with open(ruta, 'wb') as archivo:
        archivo.write(base64.b64decode(contenido))


# Datos que normalmente vienen de tu base de datos.
CERTIFICADO = {
    'certificate': leer_texto('cert.pem'),
    'privateKey': leer_texto('key.pem'),
}
CAF = leer_base64('caf_39.xml')
```

```
EMISOR = {
    'rut': '88888888-8',
    'razon_social': 'Mi Empresa SpA',
    'autorizacion_dte': {
        'fecha_resolucion': '2025-11-20',  # Fecha de autorización en certificación.
        'numero_resolucion': 0,            # En certificación siempre es 0.
    },
}
FOLIO = 1  # El siguiente folio libre del CAF de boletas.

# Paso 1: construir la boleta. Los precios van con IVA incluido.
datos = {
    'Encabezado': {
        'IdDoc': {'TipoDTE': 39, 'Folio': FOLIO},
        'Emisor': {
            'RUTEmisor': EMISOR['rut'],
            'RznSoc': EMISOR['razon_social'],
            'GiroEmis': 'Venta de artículos de computación',
            'DirOrigen': 'Av. Siempre Viva 742',
            'CmnaOrigen': 'Providencia',
        },
        'Receptor': {
            'RUTRecep': '66666666-6',  # Cliente sin RUT.
            'RznSocRecep': 'Cliente final',
            'DirRecep': 'Santiago',
            'CmnaRecep': 'Santiago',
        },
    },
    'Detalle': [
        {'NmbItem': 'Mouse inalámbrico', 'QtyItem': 2, 'PrcItem': 11900},
    ],
}

boleta = libredte('billing/document/builder/build', {
    'bag': {'parsedData': datos, 'caf': CAF, 'certificate': CERTIFICADO},
})
xml_boleta = boleta['document_xml']
guardar_base64(f'boleta_39_{FOLIO}.xml', xml_boleta)
totales = boleta['document_normalized']['Encabezado']['Totales']
print('Boleta construida:', boleta['document_id'], 'total', totales['MntTotal'])

# Paso 2: validar el esquema antes de enviar.
libredte('billing/document/validator/validateSchema', {'source': xml_boleta})

# Paso 3: armar el sobre dirigido al SII. La API crea un EnvioBOLETA.
sobre = libredte('billing/document/dispatcher/create', {
    'bag': {
        'xmlDocument': xml_boleta,
        'certificate': CERTIFICADO,
        'emisor': EMISOR,
        'receptor': {'rut': SII_RUT},
```

```
    },
})
print('Sobre:', sobre['tag'])  # EnvioBOLETA
guardar_base64(f'sobre_sii_39_{FOLIO}.xml', sobre['xml'])

# Paso 4: enviar al SII con el servicio de boletas.
request_sii = {'certificate': CERTIFICADO, 'options': {'environment': AMBIENTE}}
envio = libredte('billing/integration/sii_boleta/sendXmlDocument', {
    'request': request_sii,
    'doc': sobre['xml'],
    'company': EMISOR['rut'],
})
track_id = envio['track_id']
print('Enviado al SII, Track ID:', track_id)  # ¡Guárdalo en tu base de datos!

# Paso 5: consultar el estado hasta tener un resultado final. La primera
# espera respeta los segundos que pidió el SII al recibir el envío.
estado = None
esperas = [max(30, envio['retry_after'] or 0), 60, 120, 300, 600]
for espera in esperas:
    time.sleep(espera)
    try:
        estado = libredte('billing/integration/sii_boleta/checkXmlDocumentSentStatus',
{
            'request': request_sii,
            'trackId': track_id,
            'company': EMISOR['rut'],
        })
    except RuntimeError as error:
        # La consulta falló (por ejemplo, el SII no respondió): se reintenta.
        print('Falló la consulta, se reintenta:', error)
        continue
    print('Estado:', estado['status'], estado['description'])
    if estado['status'] in ESTADOS_FINALES:
        break

resumen = estado['resume'] if estado else {}
if not estado or estado['status'] not in ESTADOS_FINALES:
    print('Sin estado final todavía. Vuelve a consultar más tarde con el Track ID',
track_id)
elif estado['status'] == 'RPT':
    print('El SII ya había recibido este sobre: consulta el estado con el Track ID del
primer envío.')
elif estado['status'] == 'VOF':
    print('El SII no encontró el archivo: envía de nuevo el mismo sobre (no lo
construyas de nuevo).')
elif estado['status'] in ESTADOS_ACEPTADOS and resumen.get('rejected', 0) == 0:
    resultado = 'aceptada con reparos' if resumen.get('repairs') else 'aceptada'
    print('Resultado:', resultado)

    # Paso 6: generar el PDF (las boletas solo tienen copia tributaria).
```

```
    pdf = libredte('billing/document/renderer/render', {
        'bag': {
            'xmlDocument': xml_boleta,
            'emisor': EMISOR,  # Con la resolución, que se imprime en el PDF.
            'options': {'renderer': {'format': 'pdf'}},
        },
    })
    for archivo in pdf['renderings']:
        guardar_base64(archivo['filename'], archivo['content'])
        print('PDF generado:', archivo['filename'])
else:
    print('La boleta fue rechazada:', estado['status'], estado['description'])
    for detalle in estado['details']:
        print('Folio', detalle['folio'], detalle['status'], detalle['description'])
```

## Probar sin el SII

Si todavía no tienes un CAF y un certificado reales del ambiente de certificación, puedes probar los pasos 1, 2, 3 y 6 con datos de prueba. Los pasos 4 y 5 no funcionan con ellos: el SII rechaza la autenticación y el programa se detiene en el envío.

1. **El certificado:** genera uno de prueba (ver Un certificado de prueba) y guarda `cert` como `cert.pem` y `pkey` como `key.pem`.
2. **El CAF:** genera uno de prueba y guárdalo como `caf_39.xml`. Ejecuta esto una vez, con las funciones `libredte()` y `guardar_base64()` del programa. El CAF viene en base64 en el campo `xml`, y `guardar_base64()` lo decodifica al guardarlo:

```
caf = libredte('billing/identifier/caf_faker/create', {
    'emisor': {'rut': '88888888-8', 'razon_social': 'Mi Empresa SpA'},
    'codigoDocumento': 39,
    'folioDesde': 1,
    'folioHasta': 100,
})
guardar_base64('caf_39.xml', caf['xml'])
```

3. **El programa:** borra todo lo que sigue a `# Paso 4: enviar al SII con el servicio de boletas.` y pon en su lugar solo el paso 6, sin la condición de que el SII haya aceptado:

```
# Paso 6: generar el PDF, sin pasar por el SII.
pdf = libredte('billing/document/renderer/render', {
    'bag': {
        'xmlDocument': xml_boleta,
        'emisor': EMISOR,
        'options': {'renderer': {'format': 'pdf'}},
    },
})
for archivo in pdf['renderings']:
```

```
    guardar_base64(archivo['filename'], archivo['content'])
    print('PDF generado:', archivo['filename'])
```

La boleta que obtienes no tiene validez tributaria: el CAF es ficticio. Sirve para revisar los datos, los totales y el PDF.

## Qué hace, en orden

| Paso | Llamada | Qué guarda |
|---|---|---|
| 1 | `builder/build` | `boleta_39_1.xml`: la boleta timbrada y firmada |
| 2 | `validator/validateSchema` | (nada; falla si el XML no es válido) |
| 3 | `dispatcher/create` con `receptor` del SII | `sobre_sii_39_1.xml`, un `EnvioBOLETA` |
| 4 | `sii_boleta/sendXmlDocument` | El Track ID |
| 5 | `sii_boleta/checkXmlDocumentSentStatus` | El estado final |
| 6 | `renderer/render` | El PDF (copia tributaria) |

Para llevarlo a tu sistema, sepáralo igual que la factura: emitir y enviar en línea, consultar el estado en segundo plano y generar el PDF cuando la boleta es aceptada (ver Factura de principio a fin).

## Boleta exenta

La boleta exenta (`41`) se construye igual, con su propio CAF (`41`). Marca cada ítem como exento con **`IndExe: 1`**:

```
{
    "Encabezado": {
        "IdDoc": { "TipoDTE": 41, "Folio": 1 },
        "Emisor": {
            "RUTEmisor": "88888888-8",
            "RznSoc": "Mi Empresa SpA",
            "GiroEmis": "Venta de libros",
            "DirOrigen": "Av. Siempre Viva 742",
            "CmnaOrigen": "Providencia"
        },
        "Receptor": {
            "RUTRecep": "66666666-6",
            "RznSocRecep": "Cliente final",
            "DirRecep": "Santiago",
            "CmnaRecep": "Santiago"
        }
    },
    "Detalle": [
```

```
        { "IndExe": 1, "NmbItem": "Libro", "QtyItem": 1, "PrcItem": 15000 }
    ]
}
```

La API calcula `MntExe` $15.000 y `MntTotal` $15.000, sin IVA. El sobre, el envío y la consulta del estado son los mismos de la boleta afecta.

## Reglas al agrupar boletas

- Un sobre `EnvioBOLETA` lleva solo boletas (`39` y `41`). Las facturas, notas y guías van en un `EnvioDTE` aparte, que se envía con `sii_dte`.
- Puedes enviar muchas boletas en un sobre con `dispatcher/createMany` (ver Varios documentos en un sobre). Si el SII rechaza alguna, el detalle de la consulta del estado indica su folio.

## Nota de crédito de una boleta

Una boleta se anula o corrige con una nota de crédito (`61`), con referencia `TpoDocRef: 39`. La nota de crédito **no es una boleta**: va en un sobre `EnvioDTE` y se envía con `sii_dte`, como una factura (ver Nota de crédito).

---

Última actualización el 05/11/2026