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