Manual de API

Facturacion Electronica via HTTP

Element Tech ERP

Element Tech S.R.L.

Índice

1. Introduccion **2**
1.1. Caracteristicas 2

2. Autenticacion **2**
2.1. Datos de conexión 2
2.2. La API Key 2
2.3. Proceso de Autenticacion 3

3. Endpoints **3**
3.1. Resumen 3
3.2. Health Check 3

4. Crear Factura **4**
4.1. Campos del Request 4
4.2. Campos de api_partner_data 5
4.3. Campo tipo_operacion 6
4.4. Campos tipo_transaccion y presencia 7
4.5. Campos de Linea 8
4.6. Campos de api_product_data 8
4.6.1. Que es uom_id y por que es obligatorio 8
4.7. Ejemplo: Factura al Contado 9
4.8. Ejemplo: Factura a Credito 10
4.9. Creacion de Clientes 11

5. Clientes del Exterior **12**
5.1. Datos del Cliente 12
5.2. Ejemplo: Factura en Dólares 12
5.3. Moneda Extranjera y Tipo de Cambio 13
5.4. Exportación de Mercaderías 13
5.5. Actualizar un Cliente del Exterior 14
5.6. Notas de Crédito 15
5.7. Errores 15

6. Crear Nota de Credito **15**
6.1. Campos del Request . . . 16
6.2. Motivos Disponibles . . . 16
6.3. Ejemplo . . . 16

7. Buscar Factura **17**

8. Cancelar Factura **18**

9. Registrar Pago **18**

10. Crear Nota de Remisión Electrónica **19**
10.1. Campos del Request . . . 19
10.2. Catálogo: motivo_nota_remision . . . 20
10.3. Catálogo: tipo_operacion . . . 21
10.4. Catálogos auxiliares . . . 21
10.5. Campos de carrier_data . . . 21
10.6. Ejemplo: NRE básica . . . 22
10.7. Buscar Nota de Remisión . . . 24
10.8. Cancelar Nota de Remisión . . . 24
10.9. Enviar NRE por Email . . . 24

11. Descargar KUDE PDF **25**
11.1. Descargar KUDE de Factura o Nota de Crédito . . . 25
11.2. Descargar KUDE de Nota de Remisión . . . 26

12. Gestión de Clientes **26**
12.1. Buscar o Crear Cliente por RUC . . . 26
12.2. Actualizar Cliente por RUC, CI o Documento . . . 27

13. Respuestas de la API **31**
13.1. Respuesta Exitosa . . . 31
13.2. Respuesta de Error . . . 32
13.3. Codigos de Error Comunes . . . 34

14. Ejemplo Completo **35**

15. Visualizacion de Resultados **36**
15.1. Paso 1: Acceso al Modulo de Facturacion . . . 36
15.2. Paso 2: Localizacion y Acceso a la Factura . . . 37
15.3. Paso 3: Visualizacion del Codigo QR y XML . . . 37
15.3.1. Visualizacion del Codigo QR . . . 37
15.3.2. Acceso al XML Generado . . . 37
15.4. Verificacion de Integridad . . . 38

1. Introduccion

Esta API HTTP permite crear facturas electronicas, notas de credito y gestionar documentos electronicos de forma simple. Utiliza autenticacion via API Key de Element.

1.1. Caracteristicas

- Crear facturas al contado y a credito
- Crear notas de credito
- Cancelar documentos electronicos
- Consultar facturas por numero
- Registro automatico de pagos
- Auto-creacion de clientes y productos

2. Autenticacion

Toda petición se autentica con la API Key que se generó al aprovisionar su base de datos. No hay que configurar nada: la base ya viene lista para recibir peticiones.

2.1. Datos de conexión

Para empezar a usar la API solo necesita tres valores:

- **URL base** — el subdominio de su base, por ejemplo `https://80200001.elementtech.com.py`
- **Nombre de la base de datos** — por ejemplo `80200001_test`
- **API Key** — la clave de 40 caracteres que autentica cada petición

Los tres llegan en el correo que se envía cuando la base queda lista, y quedan siempre disponibles en su panel de configuración, en la tabla de credenciales de la base.

Prueba y producción

Las dos bases comparten la misma URL base y cada una tiene su propia API Key. Lo que distingue el entorno es el nombre de la base de datos: el de pruebas termina en `_test` y el de producción en `_prod`.

2.2. La API Key

La clave se genera automáticamente durante el aprovisionamiento y pertenece a un usuario de su propia base de datos, que ya tiene asignados los permisos necesarios para operar la API. No hay ningún paso previo de su parte: la clave que figura en su panel de configuración funciona desde el primer momento.

Importante

Trate la API Key como una contraseña: otorga acceso a la facturación de su empresa. Guárdela en un lugar seguro de su sistema y no la incluya en código que se publique. Si la extravía, puede volver a consultarla en su panel de configuración.

2.3. Proceso de Autenticacion

```python
import requests

# Los tres valores estan en su panel de configuracion.
URL = "https://80200001.elementtech.com.py"  # URL base de su instancia
API_KEY = "tu_api_key"                       # API Key de la base
DB = "80200001_test"                         # Base de datos: _test o
    _prod

# Paso 1: Iniciar sesion HTTP contra el servidor.
# Esto es necesario porque Element requiere una cookie de sesion
# activa vinculada a la base de datos. El parametro "db"
# le indica al servidor cual base de datos utilizar.
session = requests.Session()
session.get(f"{URL}/web", params={"db": DB})

# Paso 2: Definir headers para todas las peticiones autenticadas.
# - "Authorization: Bearer <key>": El servidor valida esta clave
#   contra las API Keys registradas en res.users.apikeys.
#   Si la clave es valida, la peticion se ejecuta con los
#   permisos del usuario propietario de la clave.
# - "Content-Type": Indica que el cuerpo de la peticion es JSON.
headers = {
    "Authorization": f"Bearer {API_KEY}",
    "Content-Type": "application/json"
}
```

Listing 1: Obtener sesion y autenticar

3. Endpoints

3.1. Resumen

3.2. Health Check

Verifica que el servicio esta disponible. No requiere autenticacion.

```python
# Este endpoint NO requiere autenticacion (ni API Key ni sesion).
# Sirve para verificar que el modulo esta instalado y el
# servicio responde correctamente antes de enviar peticiones.
resp = session.get(f"{URL}/api/fe/health")

# Respuesta esperada:
# {
#   "success": true,
#   "status": "ok",
#   "module": "eterp_de_api",
#   "version": "16.0.1.0.0"
# }
```

Listing 2: GET /api/fe/health

Metodo	Endpoint	Descripcion
GET	/api/fe/health	Estado del servicio
POST	/api/fe/invoice	Crear factura
POST	/api/fe/credit-note	Crear nota de credito
GET	/api/fe/invoice/{numero}	Buscar factura
POST	/api/fe/invoice/{numero}/cancel	Cancelar factura
POST	/api/fe/invoice/{id}/payment	Registrar pago
POST	/api/fe/remision	Crear nota de remisión
GET	/api/fe/remision/{numero}	Buscar nota de remisión
POST	/api/fe/remision/{numero}/cancel	Cancelar nota de remisión
POST	/api/fe/remision/{id}/send-email	Enviar nota de remisión por email
GET	/api/fe/invoice/{de_id}/kude-pdf	Descargar KUDE PDF de factura/NC
GET	/api/fe/remision/{de_id}/kude-pdf	Descargar KUDE PDF de nota de remisión
POST	/api/fe/partner/by-ruc/{ruc}	Buscar o crear cliente por RUC
POST	/api/fe/partner/by-ruc/{ruc}/update	Actualizar cliente por RUC
POST	/api/fe/partner/by-ci/{ci}/update	Actualizar cliente por CI
POST	/api/fe/partner/by-document/{numero}/update	Actualizar cliente por documento

4. Crear Factura

POST /api/fe/invoice

4.1. Campos del Request

Campo	Tipo	Requerido	Descripcion
partner_id	string	Si	ID externo del cliente
api_ruc	string	No	RUC para consulta SIFEN
api_partner_data	object	No	Datos del cliente
currency	string	No	Codigo de moneda (default: PYG)
payment_term_id	int	No	ID termino de pago
journal_id	int	No	ID de diario especifico
tipo_operacion	string	No	Tipo de operacion DE (1-4)
tipo_cambio	number	Cond.	Cotización del día (sección 5.3)
tipo_transaccion	string	No	Tipo de transacción (1-13)
presencia	string	No	Indicador de presencia (1-6)
exportacion	object	No	Datos de exportación (sección 5.4)
lines	array	Si	Lineas de factura

4.2. Campos de api_partner_data

El objeto `api_partner_data` contiene los datos del cliente. El campo `name` siempre es requerido, junto con uno de `ruc`, `ci` o `tipo_documento`. El resto de los campos son opcionales.

Cuadro 1: api_partner_data — identificación del cliente

Campo	Tipo	Req.	Descripción
name	string	Sí	Nombre o razón social
ruc	string	Cond.	RUC del contribuyente, con o sin dígito verificador
ci	string	Cond.	Número de cédula de identidad
tipo_documento	string	Cond.	Código del tipo de documento: `1` cédula paraguaya, `2` pasaporte, `3` cédula extranjera, `4` carnet de residencia, `5` innominado, `6` tarjeta diplomática, `9` otro
documento_numero	string	Cond.	Número del documento cuando `tipo_documento` no es `1`. Máximo 20 caracteres
tipo_documento_descripcion	string	Cond.	Descripción del documento cuando `tipo_documento` es `9`. Mínimo 9 y máximo 41 caracteres
company_type	int/string	No	`1` o `"person"` para personas físicas, `2` o `"company"` para empresas
nf	string	No	Nombre de fantasía
country_id	int/string	Cond.	País: código ISO de 2 o 3 letras (`BR` o `BRA`), nombre o ID. Requerido para clientes del exterior

Enviar `ruc` marca al cliente como contribuyente; enviar `ci` lo marca como persona física con cédula paraguaya. Para un cliente contribuyente, `company_type` determina si el XML lo informa como persona física o jurídica. Un cliente del exterior se envía con `country_id`, `tipo_documento` y `documento_numero`, sin `ruc` (sección 5).

Cuadro 2: api_partner_data — contacto y dirección

Campo	Tipo	Req.	Descripción
email	string	No	Correo electrónico del cliente
phone	string	No	Teléfono. 6 a 15 caracteres; si el cliente es de Paraguay, con el prefijo de la ciudad en vez del código de país (ej. `021445566`)
mobile	string	No	Celular. 10 a 20 caracteres
street	string	No	Calle y número
numero_casa	string	No	Número de casa. Máximo 6 caracteres
departamento_id	int/string	No	Departamento: ID o nombre
distrito_id	int/string	No	Distrito: ID o nombre
ciudad_id	int/string	No	Ciudad: ID o nombre
barrio_id	int/string	No	Barrio: ID o nombre

Los cuatro campos geográficos aceptan tanto el ID numérico como el nombre de la ubicación en texto, y se resuelven siguiendo la jerarquía departamento → distrito → ciudad → barrio. Como existen nombres de distrito y de ciudad repetidos entre departamentos distintos, conviene enviar siempre la jerarquía completa: el distrito se busca dentro del departamento indicado, y la ciudad dentro del distrito. Si un nombre resulta ambiguo, la API devuelve `400 INVALID_FIELD` con la lista de IDs que coinciden.

Dirección en el XML del Documento Electrónico

Para que la dirección del cliente se incluya en el XML del DE se deben enviar los cinco campos `street`, `numero_casa`, `departamento_id`, `distrito_id` y `ciudad_id`. Si falta alguno de ellos, el documento se genera igual pero **sin el bloque de dirección**, y no se devuelve ningún error. El campo `barrio_id` es opcional y se incluye en el XML cuando se envía.

Para un cliente del exterior la dirección se compone solo de `street` y `numero_casa`; los campos geográficos no aplican (sección 5).

El campo `email` se incluye en el XML del DE y es el destinatario del envío automático del documento por correo electrónico. Para configurar copias por tipo de documento existe el campo `email_cc`, descrito en la sección 12.

4.3. Campo tipo_operacion

El campo `tipo_operacion` define el tipo de operacion del Documento Electronico segun la normativa DNIT. Se serializa al XML del DE como `cliente.tipoOperacion`.

Valor	Codigo	Descripcion
1	B2B	Empresa a Empresa
2	B2C	Empresa a Consumidor Final (default)
3	B2G	Empresa a Entidad del Gobierno
4	B2F	Empresa a Empresa o Persona del Exterior

Si el campo se omite, se usa el default `2` (B2C). En la interfaz de Element ERP el valor se auto-ajusta al cambiar el cliente (empresa → B2B, persona → B2C), pero en la creacion via API ese ajuste automatico no ocurre, por lo que el cliente externo debe enviar el valor explicitamente cuando necesite B2B o B2G. Valores fuera de `1`–`4` devuelven `400 INVALID_FIELD`.

Para un cliente con país distinto de Paraguay no hace falta enviarlo: la operación se informa como `4` (B2F), el único valor que admite un cliente del exterior. A la inversa, `4` solo se admite con un cliente del exterior. Cualquier otra combinación devuelve `400 TIPO_OPERACION_INCOMPATIBLE`.

4.4. Campos tipo_transaccion y presencia

Ambos son opcionales y se informan en la factura electrónica. Si se omiten, se usan `3` (venta de mercaderías y servicios) y `1` (operación presencial).

Cuadro 3: tipo_transaccion

Valor	Descripción
1	Venta de mercaderías
2	Prestación de servicios
3	Venta de mercaderías y servicios
4	Venta de activo fijo
5	Venta de divisas
6	Compra de divisas
7	Promoción o entrega de muestras
8	Donación
9	Anticipo
10	Compra de productos
11	Compra de servicios
12	Venta de crédito fiscal
13	Muestras médicas

Cuadro 4: presencia

Valor	Descripción
1	Operación presencial
2	Operación electrónica
3	Operación telemarketing
4	Venta a domicilio
5	Operación bancaria
6	Operación cíclica

Un valor fuera de estas tablas devuelve `400 INVALID_FIELD`.

Campo	Tipo	Requerido	Descripcion
product_id	string/int	Si	ID externo o ID Element
api_product_data	object	No	Datos para crear producto
name	string	No	Descripcion personalizada
quantity	number	Si	Cantidad
price_unit	number	Si	Precio unitario
tax_ids	array	No	IDs de impuestos (override)
discount	number	No	Porcentaje de descuento (0-100)
discount_amount	number	No	Monto fijo de descuento por unidad

4.5. Campos de Linea

Los campos `discount` y `discount_amount` son mutuamente excluyentes. `discount` es un porcentaje (0-100); `discount_amount` es un monto fijo de descuento por unidad, en la misma denominacion que `price_unit`. Enviar ambos en la misma linea devuelve `400 INVALID_FIELD`.

4.6. Campos de api_product_data

Cuando se utiliza `api_product_data` para crear un producto nuevo automaticamente, se deben proporcionar los siguientes campos obligatorios:

Campo	Tipo	Requerido	Descripcion
name	string	Si	Nombre del producto
default_code	string	Si	Codigo interno de referencia
uom_id	int	Si	ID de unidad de medida en Element
tax_id	int	Cond.	ID de impuesto en Element
iva	string	Cond.	Tasa de IVA: `"10"`, `"5"` o `"exento"`

Se envía `tax_id` o `iva`, no ambos. Con `iva` se usa el impuesto de venta de esa tasa; si la empresa tiene más de un impuesto con la misma tasa, la API devuelve `400 INVALID_FIELD` y hay que indicar `tax_id`.

4.6.1. Que es uom_id y por que es obligatorio

El campo `uom_id` (Unit of Measure ID) corresponde al identificador de la **unidad de medida** del producto en Element ERP. Toda factura electronica en Paraguay requiere especificar la unidad de medida del bien o servicio facturado (por ejemplo: unidad, kilogramo, litro, hora, metro, etc.), ya que este dato se incluye en el XML del Documento Electronico (DE) enviado a DNIT.

En Element ERP, las unidades de medida se gestionan en el modelo `uom.uom` y cada una tiene un ID numerico unico. Al crear un producto via la API, el sistema necesita saber a que unidad de medida asociarlo. Por eso `uom_id` es un campo obligatorio dentro de `api_product_data`: el sistema valida que el ID proporcionado exista en la base de datos antes de crear el producto.

Como obtener el uom_id

Para consultar las unidades de medida disponibles en su instancia, puede:

1. Navegar a **Inventario** → **Configuracion** → **Unidades de Medida**

Cuadro 5: Unidades de medida mas comunes

ID tipico	Nombre	Uso comun
1	Unidad(es)	Productos generales, servicios
2	Docena(s)	Productos vendidos por docena
3	kg	Productos pesados
4	Litro(s)	Liquidos

Importante

Los IDs de unidades de medida pueden variar entre instalaciones de Element ERP. El valor `1` (Unidad) es el mas comun, pero **siempre verifique** los IDs disponibles en su instancia antes de integrar. Si el ID proporcionado no existe, la API retornara un error de validacion.

4.7. Ejemplo: Factura al Contado

```
resp = session.post(
    f"{URL}/api/fe/invoice",
    headers=headers,
    json={
        # --- Identificacion del cliente ---
        # "partner_id" es el ID externo del cliente en su sistema.
        # Se usa como clave para buscar o crear el partner en Element.
        # Si el cliente ya fue creado previamente con este mismo
        # partner_id, el sistema lo reutiliza automaticamente.
        "partner_id": "48213",

        # "api_ruc" activa la consulta automatica al servicio
        # SIFEN de la DNIT. El sistema consulta el RUC, obtiene
        # nombre, tipo de contribuyente y estado, y crea el
        # partner en Element con esos datos. Formato: "numero-dv".
        "api_ruc": "80000001-1",

        # --- Lineas de la factura ---
        # Cada elemento del array representa una linea.
        # "api_product_data" es OBLIGATORIO en cada linea.
        "lines": [{
            # "product_id" es el ID externo del producto.
            # Si un producto con el mismo "default_code" ya existe
            # en Element, se reutiliza. Si no, se crea uno nuevo.
            "product_id": "20458",

            # Datos para crear el producto si no existe:
            "api_product_data": {
                "name": "Servicio de Consultoria",
                "default_code": "78465674",  # Codigo interno (unico)
```

```
                "uom_id": 1,   # ID de unidad de medida en Element
                "tax_id": 1    # ID de impuesto en Element (ej: IVA 10%)
            },
            "quantity": 1,
            "price_unit": 150000   # Precio en la moneda de la factura
        }],

        # --- Opciones de procesamiento ---
        # tipo_operacion: tipo de operacion DE.
        # "1"=B2B, "2"=B2C (default), "3"=B2G, "4"=B2F.
        # El onchange automatico de la UI no se dispara al crear
        # por API, asi que para operaciones con empresas (B2B)
        # u otros tipos se debe enviar explicitamente.
        "tipo_operacion": "1",

    }
)
# La respuesta incluye el ID, numero, estado, montos y
# datos del documento electronico (DE_estado, DE_id, etc.)
print(resp.json())
```

Listing 3: Factura con consulta RUC en SIFEN

4.8. Ejemplo: Factura a Credito

```
resp = session.post(
    f"{URL}/api/fe/invoice",
    headers=headers,
    json={
        # --- Cliente con datos manuales (sin consulta SIFEN) ---
        # En lugar de api_ruc, se usa api_partner_data para
        # proporcionar los datos directamente. El sistema primero
        # busca si ya existe un partner con ese RUC en Element;
        # si lo encuentra, lo reutiliza sin crear uno nuevo.
        "partner_id": "298463811",
        "api_partner_data": {
            "name": "Empresa Cliente SA",
            "ruc": "80012345-6",  # RUC con digito verificador
            # Al incluir "ruc" el cliente queda marcado como
            # contribuyente. "company_type" define si el XML lo
            # informa como persona fisica (1) o juridica (2).
            "company_type": 2,

            # Correo del cliente. Se incluye en el XML del DE y
            # es el destinatario del envio automatico del documento.
            "email": "facturacion@empresacliente.com.py",
            "phone": "021445566",

            # Direccion del cliente. Los cinco campos siguientes
            # deben enviarse juntos para que la direccion aparezca
            # en el XML del DE.
            "street": "Avda. Mariscal Lopez 1234",
            "numero_casa": "1234",
            # Los campos geograficos aceptan ID numerico o nombre.
            "departamento_id": "Central",
            "distrito_id": "Luque",
            "ciudad_id": "Luque",
```

```
            "barrio_id": "Centro"   # Opcional
        },

        # Termino de pago: ID del registro payment.term en Element.
        # Si el termino tiene dias > 0, la factura se considera
        # "a credito" y el sistema NO registra pago automatico.
        "payment_term_id": 2,  # Ej: "30 Dias" en Element

        "lines": [{
            # Cuando product_id es un entero, se interpreta como
            # el ID interno de Element. El sistema verifica que el
            # producto exista; si no, lanza un error.
            # NOTA: api_product_data sigue siendo obligatorio
            # para lineas con ID externo (string).
            "product_id": 1,
            "quantity": 2,
            "price_unit": 250000
        }]
    }
)
```

Listing 4: Factura a credito (30 dias)

4.9. Creacion de Clientes

Cuatro formas de especificar el cliente:

Opciones de Partner

1. **api_ruc**: Consulta automatica en SIFEN por RUC
2. **api_partner_data con ruc**: Datos manuales con RUC
3. **api_partner_data con ci**: Datos manuales con cedula
4. **api_partner_data con country_id**: Cliente del exterior (sección 5)

Para dar de alta o actualizar un cliente sin emitir un documento, ver la sección 12.

```
# Para personas fisicas (no contribuyentes), se usa "ci"
# en lugar de "ruc". El partner se crea con:
#   - contribuyente = False
#   - company_type = "person"
#   - tipo_documento = segun el valor proporcionado
"api_partner_data": {
    "name": "Juan Perez",
    "ci": "1234567",          # Numero de cedula de identidad
    "tipo_documento": "1",    # "1" = Cedula de identidad paraguaya.
                              # Para otros documentos (pasaporte,
                              # cedula extranjera, etc.) se envian
                              # "tipo_documento" y "documento_numero"
                              # en lugar de "ci".
    "email": "juan.perez@gmail.com"
}
```

Listing 5: Ejemplo con cedula

5. Clientes del Exterior

Para facturar a una empresa o persona del exterior se envían en `api_partner_data` el país del cliente, su dirección y su documento de identidad. La operación se informa a la DNIT como B2F, por lo que no hace falta enviar `tipo_operacion`.

5.1. Datos del Cliente

Cuadro 6: api_partner_data para clientes del exterior

Campo	Tipo	Req.	Descripción
name	string	Sí	Nombre o razón social. Mínimo 4 caracteres
country_id	string/int	Sí	País: código ISO de 2 o 3 letras (`BR`, `BRA`), nombre (`Brasil`) o ID
street	string	Sí	Dirección completa
numero_casa	string	No	Número de casa, solo dígitos. Si se omite se informa `0`
tipo_documento	string	Sí	`2` pasaporte, `3` cédula extranjera, `4` carnet de residencia o `9` otro
documento_numero	string	Sí	Número del documento o identificador fiscal del país del cliente. Máximo 20 caracteres
tipo_documento_descripcion	string	Cond.	Requerido con `tipo_documento 9`: nombre del documento, de 9 a 41 caracteres (ej. `Cadastro Nacional PJ`)
company_type	int/string	No	`1` persona física, `2` empresa
email	string	No	Correo electrónico
phone	string	No	Teléfono, de 6 a 15 caracteres
mobile	string	No	Celular, de 10 a 20 caracteres. Admite el prefijo internacional (ej. `+55 11 98765-4321`)

Un cliente del exterior no lleva `ruc`: su identificador fiscal va en `documento_numero` con `tipo_documento 9`. Los campos `departamento_id`, `distrito_id`, `ciudad_id` y `barrio_id` no aplican. Un número de teléfono internacional suele superar los 15 caracteres de `phone`; en ese caso se envía en `mobile`.

5.2. Ejemplo: Factura en Dólares

```
resp = session.post(
    f"{URL}/api/fe/invoice",
    headers=headers,
    json={
        "partner_id": "731845290",
        "api_partner_data": {
            "name": "Comercial Exemplo Ltda",
```

```
            "company_type": 2,
            "country_id": "BR",
            "street": "Rua das Flores 100",
            "tipo_documento": "9",
            "tipo_documento_descripcion": "Cadastro Nacional PJ",
            "documento_numero": "12345678000190",
            "email": "compras@exemplo.com.br"
        },
        "currency": "USD",
        "tipo_cambio": 7300,        # Gs. por 1 USD (ver mas abajo)
        "tipo_transaccion": "2",    # Prestacion de servicios
        "presencia": "2",           # Operacion electronica
        "lines": [{
            "product_id": "516131942",
            "api_product_data": {
                "name": "Servicio de consultoria",
                "default_code": "78465791",
                "uom_id": 1,
                "iva": "exento"     # o "10" / "5", segun la operacion
            },
            "quantity": 1,
            "price_unit": 1500
        }]
    }
)
```

Listing 6: Factura a una empresa del exterior

```
"api_partner_data": {
    "name": "Martina Fernandez",
    "country_id": "AR",
    "street": "Av. Corrientes 1234",
    "tipo_documento": "2",          # Pasaporte
    "documento_numero": "AAA123456"
}
```

Listing 7: Persona del exterior con pasaporte

5.3. Moneda Extranjera y Tipo de Cambio

Una factura o nota de crédito en moneda distinta del guaraní informa a la DNIT la cotización del día. El campo `tipo_cambio` indica esa cotización en guaraníes por unidad de la moneda (ej. `7300` para 1 USD = 7.300 Gs.). Aplica a cualquier cliente, no solo a los del exterior.

- Si Element ERP ya tiene la cotización del día, `tipo_cambio` es opcional; si se envía, debe coincidir con ella.
- Si no la tiene, `tipo_cambio` es requerido y queda registrado como la cotización de ese día para los documentos siguientes.
- En una factura en guaraníes, enviar `tipo_cambio` devuelve `400 INVALID_FIELD`.

5.4. Exportación de Mercaderías

Para exportar mercaderías se agrega el objeto `exportacion`. La factura se emite en el diario de exportación de la empresa, que debe estar habilitado en Element ERP; los datos se informan a la DNIT junto con el documento.

Cuadro 7: Campos de `exportacion`

Campo	Tipo	Req.	Descripción
condicion_negociacion	string	Sí	Incoterm (ver tabla siguiente)
empresa_fletera	string	No	Empresa fletera o exportador nacional
agente_transporte	string	No	Nombre del agente de transporte
numero_conocimiento_embarque	string	No	Número/s de conocimiento/s de embarque
numero_manifiesto_carga	string	No	Número/s de manifiesto/s internacional/es de carga
numero_barcaza_remolcador	string	No	Número de barcaza o remolcador
instrucciones_pago	string	No	Instrucciones de pago para el cliente

Cuadro 8: condicion_negociacion

Valor	Descripción
`CFR`	Costo y flete
`CIF`	Costo, seguro y flete
`CIP`	Transporte y seguro pagados hasta
`CPT`	Transporte pagado hasta
`DAP`	Entregada en lugar convenido
`DAT`	Entregada en terminal
`DDP`	Entregada derechos pagados
`EXW`	En fábrica
`FAS`	Franco al costado del buque
`FCA`	Franco transportista
`FOB`	Franco a bordo

```
"exportacion": {
    "condicion_negociacion": "FOB",
    "empresa_fletera": "Transportes Ejemplo SA",
    "agente_transporte": "Agencia Maritima Ejemplo",
    "numero_conocimiento_embarque": "241578963",
    "instrucciones_pago": "Transferencia a la cuenta en USD"
}
```

Listing 8: Bloque exportacion dentro del body de la factura

Si se envía `journal_id` junto con `exportacion`, debe ser un diario de exportación. Si la empresa no tiene uno habilitado, la API devuelve `400 NO_EXPORT_JOURNAL`.

5.5. Actualizar un Cliente del Exterior

Para corregir los datos de un cliente del exterior ya registrado se usa este endpoint, con el mismo body que la actualización por RUC o CI (sección 12.2):

POST /api/fe/partner/by-document/{numero}/update

```
resp = session.post(
    f"{URL}/api/fe/partner/by-document/12345678000190/update",
    headers=headers,
    json={"street": "Rua Nova 200", "country_id": "BR"}
)
```

Listing 9: Corregir la direccion de un cliente del exterior

5.6. Notas de Crédito

Una nota de crédito sobre una factura a un cliente del exterior se crea igual que cualquier otra (sección 6). Sin `lines`, toma las líneas, la moneda y el tipo de operación de la factura original.

5.7. Errores

Código	Causa
TIPO_OPERACION_INCOMPATIBLE	Cliente del exterior con `tipo_operacion` distinto de `4`, o `4` con un cliente de Paraguay
RECEPTOR_EXTERIOR_INCOMPLETO	Faltan datos del cliente o tienen un formato inválido. El mensaje enumera cada problema
TIPO_CAMBIO_FALTANTE	No hay cotización del día y no se envió `tipo_cambio`
TIPO_CAMBIO_DISTINTO	`tipo_cambio` no coincide con la cotización del día ya registrada
NO_EXPORT_JOURNAL	Se envió `exportacion` y la empresa no tiene diario de exportación
INVALID_FIELD	País no reconocido, `ruc` en un cliente del exterior u otro campo inválido

```
{
    "success": false,
    "error": {
        "code": "RECEPTOR_EXTERIOR_INCOMPLETO",
        "message": "Faltan datos del cliente del exterior Comercial
   Exemplo Ltda: falta 'street': la direccion es obligatoria en B2F;
   falta 'documento_numero'"
    }
}
```

Listing 10: Ejemplo de error por datos faltantes

6. Crear Nota de Credito

POST /api/fe/credit-note

6.1. Campos del Request

Campo	Tipo	Requerido	Descripcion
original_invoice_number	string	No	Numero de factura original
original_invoice_id	int	No	ID de factura original
motivo	string	No	Codigo de motivo (1-8)
partner_id	string	Condicional	Si no hay factura original
lines	array	Condicional	Si no hay factura original
tipo_operacion	string	No	Tipo de operacion DE (1-4)
amount_factor	float	No	Factor de monto (default: 1.0)
currency	string	No	Moneda, cuando se envían lines
tipo_cambio	number	Cond.	Cotización del día (sección 5.3)

Sin `lines`, la nota de crédito toma las líneas, la moneda y el tipo de operación de la factura original; `amount_factor` permite una nota de crédito parcial. Con `lines`, la moneda se indica en `currency`, igual que en una factura.

6.2. Motivos Disponibles

Codigo	Descripcion
1	Devolucion y Ajuste de precios
2	Devolucion
3	Descuento
4	Bonificacion
5	Credito incobrable
6	Recupero de costo
7	Recupero de gasto
8	Ajuste de precio

6.3. Ejemplo

```
resp = session.post(
    f"{URL}/api/fe/credit-note",
    headers=headers,
    json={
        # Numero de la factura original a la que aplica la NC.
        # El sistema busca la factura y vincula la NC mediante
        # "reversed_entry_id". Alternativa: usar
        # "original_invoice_id" con el ID numerico de Element.
        "original_invoice_number": "001-001-0004661",

        # Codigo de motivo segun normativa SIFEN/DNIT.
        # Se almacena en "motivo_nota_credito_debito".
        # Ver tabla de motivos disponibles mas arriba.
        "motivo": "1",  # 1 = Devolucion y Ajuste de precios

        # Lineas personalizadas para la NC. Si se omite
```

```
        # "lines", el sistema copia las lineas de la factura
        # original. Se puede usar "amount_factor" (ej: 0.5)
        # para crear una NC parcial.
        "lines": [{
            "product_id": "516131774",
            "api_product_data": {
                "name": "Servicio a devolver",
                "default_code": "78465711",
                "uom_id": 1,
                "tax_id": 1
            },
            "quantity": 1,
            "price_unit": 100000
        }],

    }
)
```

Listing 11: Nota de credito basada en factura

7. Buscar Factura

GET /api/fe/invoice/{numero}?move_type={tipo}

Parámetro de query opcional

Valor	Descripción
`invoice`	Busca solo facturas
`credit_note`	Busca solo notas de crédito
(omitido)	Busca en ambos tipos

Códigos de respuesta

HTTP	Código	Descripción
200	—	Documento encontrado
404	`NOT_FOUND`	No existe documento con ese número
409	`AMBIGUOUS_NUMBER`	El número corresponde a una factura y una NC; usar `?move_type=` para

```
resp = session.get(
    f"{URL}/api/fe/invoice/001-001-0004707",
    headers=headers
)

# Con filtro de tipo:
resp = session.get(
    f"{URL}/api/fe/invoice/001-001-0004707?move_type=invoice",
    headers=headers
```

```
)
```

Listing 12: Buscar factura por numero

8. Cancelar Factura

POST /api/fe/invoice/{numero}/cancel

Atencion

La cancelacion es **irreversible**. Solo documentos con DE_estado = 'aprobado'.

Campos del request

Campo	Tipo	Requerido	Descripción
motivo	string	Sí	Motivo de cancelación (5–500 caracteres). Queda registrado perman
move_type	string	No	`invoice` o `credit_note`. Requerido si el número es ambiguo.

```
resp = session.post(
    f"{URL}/api/fe/invoice/001-001-0004707/cancel",
    headers=headers,
    json={
        "motivo": "Solicitud de anulacion del cliente",
        # "move_type": "invoice"  # incluir si el numero es ambiguo
    }
)
```

Listing 13: Cancelar factura electronica

9. Registrar Pago

POST /api/fe/invoice/{id}/payment

Campo	Tipo	Requerido	Descripcion
journal_type	string	No	`cash` o `bank` (default: cash)
payment_journal_id	int	No	ID especifico de diario
amount	number	No	Monto a pagar (default: total pendiente)

```
# Registra un pago para una factura ya publicada (posted).
# Utiliza el wizard "account.payment.register" de Element.
# REQUISITOS:
# - La factura debe estar en state="posted"
# - La factura NO debe estar ya pagada (payment_state != "paid")
# Retorna error INVALID_STATE o ALREADY_PAID si no cumple.
resp = session.post(
    # El {id} en la URL es el ID numerico de la factura en Element
    # (campo "id" retornado al crear o buscar la factura).
```

```
    f"{URL}/api/fe/invoice/123/payment",
    headers=headers,
    json={
        # Tipo de diario de pago:
        # - "cash": diario de caja (efectivo) - valor por defecto
        # - "bank": diario de banco (transferencia)
        # Alternativamente, usar "payment_journal_id" con el ID
        # especifico del diario deseado.
        "journal_type": "cash",

        # Monto del pago. Si se omite, se paga el total
        # pendiente (amount_residual de la factura).
        # Permite pagos parciales especificando un monto menor.
        "amount": 50000
    }
)
# Respuesta: payment_id, payment_name, invoice_payment_state
```

Listing 14: Registrar pago manualmente

10. Crear Nota de Remisión Electrónica

POST /api/fe/remision

Una nota de remisión electrónica (NRE) ampara el traslado de mercaderías entre dos puntos. La API crea el `stock.picking`, asigna la secuencia, llena los datos del DE de remisión y dispara la generación del XML automáticamente al validar el picking.

Para crear una NRE el cliente externo debe enviar:

- Un **partner** (cliente destinatario): `partner_id` y `api_partner_data`.
- Un **punto de salida**: `warehouse_data` (datos del almacén de salida).
- Una **secuencia NRE**: `secuencia_id` (id) o `prefijo` (string, ej: `"002-001-"`).
- Un **transportista**: `carrier_data` (objeto con datos del vehículo, chofer y empresa).
- Los **datos del traslado**: motivo, fechas, distancia, tipo y modalidad de transporte.
- Las **líneas** de productos a trasladar.

10.1. Campos del Request

Las claves de `api_partner_data` son las mismas descritas en la sección 4.2. En la nota de remisión el destinatario debe tener `departamento_id`, `distrito_id` y `ciudad_id`: si falta alguno de los tres, la generación del DE de remisión devuelve un error. La calle y el número de casa también son obligatorios en el XML y pueden enviarse con `partner_street` y `partner_numero_casa`.

Tres campos obligatorios

1. **Punto de salida**: `warehouse_data`.
2. **Secuencia**: `prefijo`.
3. **Transportista**: `carrier_data`.

Si falta uno de estos tres campos, la API retorna un error 400.

Campo	Tipo	Requerido	Descripción
partner_id	string	Sí	ID externo del cliente
api_partner_data	object	Cond.	Datos del partner
api_ruc	string	Cond.	Consulta SIFEN
warehouse_data	object	Sí	Datos del almacén de salida
prefijo	string	Sí	Prefijo de la secuencia (ej: `002-001-`)
tipo_operacion	string	Sí	B2B / B2C / B2G / B2F
motivo_nota_remision	string	Sí	Código del motivo de traslado (ver catálogo)
descripcion_motivo_remision	string	Cond.	Si motivo=`99`, 5-60 caracteres
responsable_emision	string	Sí	Responsable de emisión (ver catálogo)
distancia_estimada	int	Sí	km, rango 1-99999
tipo_transporte	string	Sí	`1` Propio, `2` Tercero
modalidad_transporte	string	Sí	Terrestre / Fluvial / Aéreo / Multimodal
responsable_costo_flete	string	Sí	Responsable del flete (ver catálogo)
inicio_estimado_traslado	date	Sí	Formato YYYY-MM-DD
fin_estimado_traslado	date	Sí	$\geq$ inicio
carrier_data	object	Sí	Datos del transportista
lines	array	Sí	$\geq$1 línea con quantity > 0
partner_street	string	No	Actualiza la calle del partner
partner_numero_casa	string	No	Actualiza el número de casa

10.2. Catálogo: motivo_nota_remision

Código	Descripción
1	Traslado por venta
2	Traslado por consignación
3	Exportación
4	Traslado por compra
5	Importación
6	Traslado por devolución
7	Traslado entre locales de la empresa
8	Traslado de bienes por transformación
9	Traslado de bienes por reparación
10	Traslado por emisor móvil
11	Exhibición o demostración
12	Participación en ferias
13	Traslado de encomienda
14	Decomiso
99	Otro (requiere descripcion_motivo_remision)

10.3. Catálogo: tipo_operacion

Valor	Código	Descripción
1	B2B	Empresa a Empresa
2	B2C	Empresa a Consumidor Final (default)
3	B2G	Empresa a Entidad del Gobierno
4	B2F	Empresa a Empresa o Persona del Exterior

Si se omite, se usa el default 2 (B2C). Valores fuera de 1–4 devuelven `400 INVALID_FIELD`.

10.4. Catálogos auxiliares

Cuadro 9: responsable_emision

Código	Descripción
1	Emisor de la factura
2	Poseedor de la factura y bienes
3	Empresa transportista
4	Despachante de Aduanas
5	Agente de transporte o intermediario

Cuadro 10: responsable_costo_flete

Código	Descripción
1	Emisor de la Factura Electrónica
2	Receptor de la Factura Electrónica
3	Tercero
4	Agente intermediario del transporte
5	Transporte propio

Cuadro 11: tipo_transporte y modalidad_transporte

Código	tipo_transporte	modalidad_transporte
1	Propio	Terrestre
2	Tercero	Fluvial
3	—	Aéreo
4	—	Multimodal

10.5. Campos de carrier_data

El campo `carrier_data` permite enviar los datos del transportista sin preconfiguración previa en el sistema. El sistema busca el vehículo por matrícula, el chofer por tipo y número de documento, y la empresa transportista por RUC o documento; si alguno no existe, lo crea automáticamente.

Cuadro 12: Campos del vehículo y chofer

Campo	Tipo	Descripción
nombre_transportista	string	Nombre de la empresa transportista
marca_vehiculo	string	Marca del vehículo
categoria_vehiculo	string	Categoría
matricula_vehiculo	string	Matrícula
tipo_documento_vehiculo	string	1=N° identificación, 2=N° matrícula
nombre_chofer	string	Nombre completo del chofer
tipo_documento_chofer	string	1=CI, 2=Pasaporte, 3=Cédula extranjera, 4=Carnet residencia
numero_documento_chofer	string	Número de documento del chofer
street_transportista	string	Dirección de la empresa (obligatoria en el XML)

Cuadro 13: Campos según contribuyente_transportista

Campo	Si contribuyente=true	Si contribuyente=false
ruc_transportista	Requerido	—
tipo_documento_transportista	—	Requerido (1–4)
numero_documento_transportista	—	Requerido

10.6. Ejemplo: NRE básica

```
resp = session.post(
    f"{URL}/api/fe/remision",
    headers=headers,
    json={
        # --- Cliente destinatario ---
        # ID externo. Si ya existe el partner mapeado, se reutiliza.
        # En caso contrario, se crea con los datos de api_partner_data.
        "partner_id": "48214",
        "api_partner_data": {
            "name": "Distribuidora ABC S.A.",
            "ruc": "80123456-7",       # RUC con digito verificador
            "company_type": 2,         # 2=empresa, 1=persona
            "email": "logistica@distribuidoraabc.com.py",
            # Datos geograficos: aceptan ID numerico o nombre en texto.
            # La API resuelve los IDs segun la jerarquia SIFEN.
            "departamento_id": "Central",
            "distrito_id": "Luque",
            "ciudad_id": "Luque"
        },
        # Calle y numero de casa del destinatario.
        # Obligatorios en el XML del DE de remision.
        "partner_street": "Av. Mariscal Lopez 1234",
        "partner_numero_casa": "1234",

        # --- Punto de salida ---
        # warehouse_data crea o reutiliza el almacen por nombre (no
   genera duplicados).
        "warehouse_data": {
            "name": "Almacen Principal",
            "code": "MAIN",
            "partner_street": "Av. Eusebio Ayala 1234",
```

```
        "partner_numero_casa": "1234",
        "partner_departamento": "Central",
        "partner_distrito": "Luque",
        "partner_ciudad": "Luque",
        "partner_barrio": "Centro",
        "location_direccion": "Av. Eusebio Ayala 1234",
        "location_numero_casa": "1234",
        "location_departamento": "Central",
        "location_distrito": "Luque",
        "location_ciudad": "Luque",
    },

    # --- Secuencia NRE ---
    # El prefijo identifica la secuencia DE de remision.
    # Configurada en: Facturacion > Configuracion > DE > Secuencias.
    # Alternativa: secuencia_id (ID numerico).
    "prefijo": "002-001-",

    # --- Datos del traslado ---
    "tipo_operacion": "1",            # 1=B2B
    "motivo_nota_remision": "1",      # 1=Traslado por venta
    "responsable_emision": "1",       # 1=Emisor de la factura
    "distancia_estimada": 25,         # km estimados
    "tipo_transporte": "1",           # 1=Propio
    "modalidad_transporte": "1",      # 1=Terrestre
    "responsable_costo_flete": "1",   # 1=Emisor FE
    "inicio_estimado_traslado": "2026-05-05",
    "fin_estimado_traslado": "2026-05-06",

    # --- Transportista ---
    # El sistema crea o reutiliza vehiculo, chofer y
    # empresa transportista segun los datos provistos.
    "carrier_data": {
        "nombre_transportista": "Transportes XYZ S.A.",
        "marca_vehiculo": "Toyota",
        "categoria_vehiculo": "Camion",
        "matricula_vehiculo": "ABC-123",
        "tipo_documento_vehiculo": "2",    # 2=Nro de matricula
        "nombre_chofer": "Juan Perez",
        "tipo_documento_chofer": "1",       # 1=CI paraguaya
        "numero_documento_chofer": "4567890",
        "street_transportista": "Av. Espana 999",
        # Empresa transportista contribuyente:
        "contribuyente_transportista": True,
        "ruc_transportista": "80999888-7"
    },

    # --- Lineas a trasladar ---
    "lines": [{
        "product_id": "20458",
        "api_product_data": {
            "name": "Mercaderia general",
            "default_code": "516131737",
            "uom_id": 1,        # Unidad de medida
            "tax_id": 1
        },
        "quantity": 5
    }]
```

```
    }
)
# La respuesta incluye id, name (numero NRE), DE_estado,
# DE_id (CDC), DE_xml y DE_url cuando fue aprobada por DNIT.
print(resp.json())
```

Listing 15: Crear NRE con partner manual y carrier_data

Recomendación: usar api_partner_data en lugar de api_ruc

La consulta SIFEN (`api_ruc`) puede fallar si el RUC no está registrado en el servicio de la DNIT. Para producción, prefiera `api_partner_data` con los datos manuales: el ejemplo anterior siempre funciona, mientras que `api_ruc` depende de la disponibilidad del servicio externo.

10.7. Buscar Nota de Remisión

GET /api/fe/remision/{numero}

```
# Busca una NRE por su numero (campo "name" del stock.picking).
# Retorna 404 NOT_FOUND si no existe.
resp = session.get(
    f"{URL}/api/fe/remision/002-001-0000123",
    headers=headers
)
data = resp.json()["data"]
print(data["DE_estado"])   # pendiente, aprobado, rechazado, cancelado
print(data["DE_id"])       # CDC de 44 digitos
```

Listing 16: Buscar NRE por numero

10.8. Cancelar Nota de Remisión

POST /api/fe/remision/{numero}/cancel

```
# Cancela una NRE ya aprobada por DNIT.
# REQUISITOS:
# - DE_estado debe ser "aprobado"
# - El motivo debe tener al menos 5 caracteres (recomendado: 10-200)
resp = session.post(
    f"{URL}/api/fe/remision/002-001-0000123/cancel",
    headers=headers,
    json={"motivo": "El cliente cancelo el pedido antes del despacho"}
)
# Respuesta: DE_estado pasa a "cancelado".
```

Listing 17: Cancelar una NRE aprobada

10.9. Enviar NRE por Email

POST /api/fe/remision/{id}/send-email

Envía la NRE por correo electrónico con el XML y el KuDE en PDF adjuntos. El parámetro {id} es el ID numérico interno del picking (no el número de NRE).

```
# El picking_id es el campo "id" retornado al crear o
# buscar la NRE. NOTA: se usa el ID numerico, no el numero.
picking_id = 456
resp = session.post(
    f"{URL}/api/fe/remision/{picking_id}/send-email",
    headers=headers,
    json={"email": "destinatario@cliente.com.py"}
)
# La respuesta confirma email_to, remision_name y mail_state.
# Requiere servidor de correo configurado en los ajustes DE.
```

Listing 18: Enviar NRE por email

11. Descargar KUDE PDF

El **KUDE** es la representación gráfica oficial del Documento Electrónico en formato PDF. La API permite descargarlo directamente mediante el CDC del documento.

El identificador en el path es el **CDC** (`DE_id`), el código de 44 dígitos asignado por SIFEN a cada documento electrónico aprobado. Para obtener el `DE_id` de un documento, usar `GET /api/fe/invoice/{numero}`.

11.1. Descargar KUDE de Factura o Nota de Crédito

`GET /api/fe/invoice/{de_id}/kude-pdf`

Descarga el KUDE PDF de una factura o nota de crédito. Devuelve directamente los bytes del PDF (`Content-Type: application/pdf`, `Content-Disposition: attachment`).

Parámetro	Tipo	Descripción
`de_id`	string	CDC de 44 dígitos del documento (`DE_id`)

Ejemplo (dos pasos: primero obtener el CDC, luego descargar el PDF):

```
# Paso 1: obtener el CDC buscando la factura por numero
curl -b cookies.txt -X GET \
  "https://ejemplo.com/api/fe/invoice/001-001-0004707" \
  -H "Authorization: Bearer TU_API_KEY" \
  -H "X-Database: nombre_db"
# La respuesta incluye: "DE_id":
   "01800000011001001000000012024010112345678901"

# Paso 2: descargar el KUDE en PDF
curl -b cookies.txt -X GET \
  "https://ejemplo.com/api/fe/invoice
   /01800000011001001000000012024010112345678901/kude-pdf" \
  -H "Authorization: Bearer TU_API_KEY" \
  -H "X-Database: nombre_db" \
  --output kude.pdf
```

HTTP	Código	Descripción
200	—	PDF descargado correctamente
404	NOT_FOUND	No existe documento con ese CDC
400	DE_NO_XML	El documento existe pero aún no tiene XML generado
500	PDF_ERROR	Error al renderizar el PDF
401	UNAUTHORIZED	API Key inválida o ausente

11.2. Descargar KUDE de Nota de Remisión

`GET /api/fe/remision/{de_id}/kude-pdf`

Idéntico al anterior, pero para notas de remisión electrónicas. El `de_id` se obtiene desde `GET /api/fe/remision/{numero}`.

```
curl -b cookies.txt -X GET \
  "https://ejemplo.com/api/fe/remision
   /04800000011001001000000012024010112345678901/kude-pdf" \
  -H "Authorization: Bearer TU_API_KEY" \
  -H "X-Database: nombre_db" \
  --output kude_remision.pdf
```

Cookie de sesion en instancias multi-base-de-datos

En instalaciones con multiples bases de datos, establecer la cookie de sesion antes de llamar al endpoint:

```
curl -c cookies.txt "https://ejemplo.com/web?db=nombre_db"
```

Usar esa cookie (`-b cookies.txt`) en todas las llamadas siguientes. No requiere credenciales.

12. Gestión de Clientes

Estos endpoints permiten dar de alta o actualizar un cliente en Element ERP de forma independiente a la creación de un documento electrónico.

12.1. Buscar o Crear Cliente por RUC

POST /api/fe/partner/by-ruc/{ruc}

Consulta el RUC en SIFEN y devuelve el contacto correspondiente en Element ERP. No requiere body. El RUC en la URL puede enviarse con o sin dígito verificador (`80000001` o `80000001-1`).

```
resp = session.post(
    f"{URL}/api/fe/partner/by-ruc/80000001-1",
    headers=headers
)
result = resp.json()
print(result["created"])       # true si se creo un contacto nuevo
print(result["data"]["id"])    # ID de Element ERP para usar en otras
   llamadas
```

Listing 19: Buscar o crear cliente por RUC

```
{
    "success": true,
    "created": true,
    "data": {
        "id": 123,
        "name": "EMPRESA EJEMPLO S.A.",
        "ruc": "80000001-1",
        "ruc_state": "ACT",
        "contribuyente": true,
        "company_type": "company"
    }
}
```

Listing 20: Respuesta exitosa

`created` indica si el contacto fue creado (`true`) o ya existía (`false`). `ruc_state` es el estado del RUC informado por SIFEN (por ejemplo `"ACT"`).

HTTP	Código	Descripción
200	—	Contacto encontrado o creado
400	`INVALID_FIELD`	RUC no encontrado en SIFEN o con formato inválido
400	`CREATE_ERROR`	Error al crear el contacto

12.2. Actualizar Cliente por RUC, CI o Documento

POST /api/fe/partner/by-ruc/{ruc}/update
POST /api/fe/partner/by-ci/{ci}/update
POST /api/fe/partner/by-document/{numero}/update

El último ubica al contacto por `documento_numero`: pasaporte, cédula extranjera o identificador fiscal de un cliente del exterior.

Actualiza uno o más contactos existentes. Solo se modifican los campos incluidos en el body; los omitidos no se tocan. Enviar `null` vacía el campo explícitamente (excepto `name`, que no puede quedar vacío).

Coincidencias multiples

Si el RUC, la CI o el documento coincide con más de un contacto, la actualización se aplica a todos. El campo `matched_count` en la respuesta indica cuántos contactos fueron afectados.

Cuadro 14: Campos estándar y de dirección

Campo	Tipo	Req.	Descripción
name	string	No	Nombre o razón social
ref	string	No	Referencia interna
company_type	int/string	No	`1` o `"person"` para persona física, `2` o `"company"` para empresa
lang	string	No	Código de idioma
comment	string	No	Notas internas
website	string	No	URL del sitio web
phone	string	No	Teléfono. 6 a 15 caracteres; si el cliente es de Paraguay, con el prefijo de la ciudad en vez del código de país (ej. `021445566`)
mobile	string	No	Celular
email	string	No	Correo electrónico, destinatario principal del envío automático del DE
vat	string	No	Identificación fiscal
street	string	No	Dirección (línea 1)
street2	string	No	Dirección (línea 2)
city	string	No	Ciudad (texto libre)
zip	string	No	Código postal
state_id	int/string	No	Estado o provincia: ID o nombre
country_id	int/string	No	País: código ISO de 2 o 3 letras, nombre o ID

Cuadro 15: Campos de facturación electrónica

Campo	Tipo	Req.	Descripción
ruc	string	No	RUC del contribuyente; se reformatea con dígito verificador automáticamente
ci	string	No	Número de cédula de identidad
tipo_documento	string	No	Código del tipo de documento (ver tabla en la sección 4.2)
tipo_documento_descripcion	string	No	Descripción del documento cuando `tipo_documento` es `9`. Mínimo 9 y máximo 41 caracteres
documento_numero	string	No	Número del documento. Máximo 20 caracteres
contribuyente	bool	No	Es contribuyente ante la DNIT
departamento_id	int/string	No	Departamento: ID o nombre
distrito_id	int/string	No	Distrito: ID o nombre
ciudad_id	int/string	No	Ciudad: ID o nombre
barrio_id	int/string	No	Barrio: ID o nombre
numero_casa	string	No	Número de casa. Máximo 6 caracteres
id_impositivo	string	No	Identificador impositivo interno. Para clientes del exterior, el que se informa a la DNIT es `documento_numero`
tax_id	string	No	Identificador fiscal adicional
cuit	string	No	CUIT (Argentina)

Los campos geográficos aceptan ID numérico o nombre, con la misma jerarquía descrita en la sección 4.2.

Cuadro 16: Campos especiales

Campo	Descripción
api_ruc	Consulta el RUC en SIFEN y sobrescribe `name`, `ruc`, `contribuyente` y `ruc_state` con los datos oficiales
external_id	Vincula el contacto al mapeo de IDs externos de la API, para reutilizarlo luego en la creación de facturas o notas de remisión. Solo puede usarse cuando la búsqueda encuentra un único contacto
email_cc	Configura los correos en copia (CC) al enviar cada tipo de documento electrónico del cliente. Ver detalle a continuación

El `ruc` o `ci` de la URL solo sirve para ubicar el contacto a actualizar. `api_ruc` es un campo aparte, del body: si se envía, dispara una nueva consulta a SIFEN y sobrescribe `name`, `ruc`, `contribuyente` y `ruc_state` con los datos oficiales devueltos, incluso si el

body ya trae esos mismos campos con otro valor. En la práctica se envía el mismo RUC que identifica al contacto en la URL, para refrescar sus datos oficiales sin tocar el resto:

```
resp = session.post(
    f"{URL}/api/fe/partner/by-ruc/80000001-1/update",
    headers=headers,
    json={"api_ruc": "80000001-1"}
)
```

Listing 21: Refrescar datos desde SIFEN

`external_id` vincula el contacto al sistema de mapeo de IDs externos de la API: a partir de esa llamada, el mismo valor puede usarse como `partner_id` en `/api/fe/invoice`, `/api/fe/credit-note` o `/api/fe/remision`, sin volver a enviar `api_ruc` ni `api_partner_data`. Si el RUC o CI de la URL coincide con más de un contacto, enviar `external_id` cancela toda la actualización (ningún campo se modifica) y devuelve `400 INVALID_FIELD`.

```
resp = session.post(
    f"{URL}/api/fe/partner/by-ruc/80000001-1/update",
    headers=headers,
    json={"external_id": "48213"}
)
```

Listing 22: Vincular external_id

Desde entonces, `"partner_id": "48213"` identifica a este contacto en cualquier otro endpoint de la API.

`email_cc` es un objeto: cada clave es un tipo de documento (`factura`, `nota_credito`, `nota_debito` o `nota_remision`) y cada valor es la lista de correos que reciben copia para ese tipo.

```
"email_cc": {
    "factura": ["facturacion@cliente.com.py", "contabilidad@cliente.com.
   py"],
    "nota_credito": ["facturacion@cliente.com.py"],
    "nota_debito": []
}
```

Listing 23: Forma del campo email_cc

La lista enviada para un tipo reemplaza por completo la configuración anterior de ese tipo: en el ejemplo, `nota_debito` queda sin copias porque se envió una lista vacía. Un tipo no incluido en el objeto no se modifica. Enviar `email_cc` no toca el correo principal (`email`) ni viceversa: son dos campos independientes.

Destinatario principal requerido

El envío automático del documento electrónico solo se dispara si el contacto tiene `email` configurado. Sin `email` principal, los destinatarios de `email_cc` tampoco reciben copia.

```
resp = session.post(
    f"{URL}/api/fe/partner/by-ruc/80000001-1/update",
    headers=headers,
    json={
        "email": "facturacion@cliente.com.py",
        "phone": "021445566",
```

```
        "email_cc": {
            "factura": ["facturacion@cliente.com.py", "
    contabilidad@cliente.com.py"]
        }
    }
)
```

Listing 24: Actualizar contacto por RUC

Actualizar por cédula sigue el mismo formato, contra `/api/fe/partner/by-ci/{ci}/update`.

```
{
    "success": true,
    "data": {
        "matched_count": 1,
        "updated": [
            {
                "id": 123,
                "name": "EMPRESA EJEMPLO S.A.",
                "ruc": "80000001-1",
                "email": "facturacion@cliente.com.py"
            }
        ]
    }
}
```

Listing 25: Respuesta exitosa

Atencion

Si se actualiza el `ruc`, la `ci` o el `documento_numero` usados para identificar al contacto en la URL, quedará inalcanzable por esa clave en llamadas futuras.

HTTP	Código	Descripción
200	—	Contacto(s) actualizado(s)
404	`NOT_FOUND`	Ningún contacto con ese RUC, CI o documento
400	`INVALID_JSON`	Body ausente o JSON inválido
400	`INVALID_FIELD`	Campo no permitido o validación fallida
400	`UPDATE_ERROR`	Error genérico durante la actualización

13. Respuestas de la API

13.1. Respuesta Exitosa

```
{
    "success": true,        # Siempre true en respuestas exitosas
    "data": {
        # --- Identificacion ---
        "id": 123,                          # ID interno de Element
        "name": "001-001-0004707",          # Numero de factura

        # --- Estado contable ---
```

```
        "state": "posted",                # "draft", "posted" o "cancel"

        # --- Estado electronico (campos DE del modulo eterp_de) ---
        "DE_estado": "aprobado",          # "pendiente", "aprobado",
                                          # "rechazado" o "cancelado"
        "DE_id": "01800...",              # CDC: Codigo de Control
                                          # (44 digitos, identificador
                                          # unico del DE ante SIFEN)
        "DE_xml": "base64_encoded_xml...",# XML completo del DE en
                                          # Base64. Decodificar para
                                          # obtener el XML original.
                                          # Util para generar tickets.
        "DE_url": "https://ekuatia.set.gov.py/...",
                                          # URL de verificacion publica
        "DE_mensaje": "Aprobado",
                                          # Mensaje de respuesta SIFEN

        # --- Montos ---
        "amount_total": 150000,           # Total con impuestos
        "amount_residual": 0,             # Saldo pendiente de cobro
        "currency": "PYG",                # Codigo de moneda

        # --- Pago ---
        "payment_state": "paid",          # "not_paid", "partial",
                                          # "paid", "in_payment"
        "payment_term_id": 1,             # ID del termino de pago
        "payment_term_name": "Pago inmediato",
        "payment_term_days": 0            # Total de dias del termino.
                                          # 0 = contado, >0 = credito
    }
}
```

Listing 26: Estructura de respuesta exitosa

Descripcion de los campos mas relevantes:

- `id / name`: ID interno de Element y numero de factura
- `state`: Estado contable (`draft`, `posted`, `cancel`)
- `DE_estado`: Estado electronico (`pendiente`, `aprobado`, `rechazado`, `cancelado`)
- `DE_id`: Codigo de Control (CDC) de 44 digitos, identificador unico del DE ante SIFEN
- `DE_xml`: XML completo del DE codificado en Base64, decodificable para generar tickets impresos
- `DE_url`: URL publica de verificacion en el portal ekuatia de la DNIT; solo tiene valor cuando `DE_estado = .ªprobado"`
- `payment_term_days`: 0 = contado, mayor a 0 = credito

13.2. Respuesta de Error

```
{
    "success": false,       # Siempre false en respuestas de error
    "error": {
        # Codigo de error programatico. Usar este campo para
        # manejar errores en el codigo (switch/if).
        # Ver tabla de codigos comunes mas abajo.
        "code": "CREATE_ERROR",
```

```
        # Mensaje legible describiendo el error.
        # Puede contener detalles tecnicos de Element
        # (ej: "El campo 'uom_id' con valor 99 no existe").
        "message": "Descripcion del error"
    }
}
```

Listing 27: Estructura de respuesta de error

13.3. Codigos de Error Comunes

Codigo	HTTP	Descripcion
AUTH_MISSING	401	Falta header Authorization
AUTH_INVALID	401	API Key invalida
API_MODE_DISABLED	403	Modo API no activo
NOT_FOUND	404	Factura o NRE no encontrada
INVALID_JSON	400	Body JSON invalido
INVALID_FIELD	400	Campo o valor invalido en el body
CREATE_ERROR	400	Error al crear documento
UPDATE_ERROR	400	Error al actualizar un contacto
TIPO_OPERACION_INCOMPATIBLE	400	tipo_operacion no coincide con el país del cliente
RECEPTOR_EXTERIOR_INCOMPLETO	400	Faltan datos del cliente del exterior
TIPO_CAMBIO_FALTANTE	400	Falta la cotización del día
TIPO_CAMBIO_DISTINTO	400	tipo_cambio distinto de la cotización registrada
NO_EXPORT_JOURNAL	400	Sin diario de exportación habilitado
INVALID_MOTIVO	400	Motivo NRE inválido o falta descripción
FALTA_DESCRIPCION_MOTIVO	400	motivo=99 sin descripción válida (5-60)
INVALID_DISTANCIA	400	distancia_estimada fuera de [1, 99999]
FECHA_FIN_ANTES_INICIO	400	fin_estimado_traslado < inicio
INVALID_PICKING_TYPE	400	Falta warehouse_data
INVALID_WAREHOUSE	400	Almacén sin tipo de salida configurado
NO_SEQ	400	Falta secuencia_id o prefijo
INVALID_PREFIX	400	Prefijo no encontrado en config
NO_CONFIG	400	No hay config DE activa para la compañía
FALTA_CARRIER	400	Falta campo carrier_data
NO_LINES	400	lines vacío
INVALID_LINE	400	quantity faltante o ≤ 0
NOT_NRE	400	El picking no es una NRE
DE_NO_XML	400	NRE sin XML generado
INVALID_EMAIL	400	Email inválido
EMAIL_CONFIG_ERROR	500	Sin servidor de email configurado
EMAIL_ERROR	500	Error generando PDF o enviando mail
CANCEL_ERROR	400	NRE no aprobada o motivo inválido

14. Ejemplo Completo

```
#!/usr/bin/env python3
"""
Script minimo para crear una factura electronica via API.
Requisitos:
  - pip install requests
  - URL base, nombre de la base de datos y API Key
    (ver seccion Autenticacion)
"""
import requests

# === CONFIGURACION ===
# Los tres valores estan en su panel de configuracion.
URL = "https://80200001.elementtech.com.py"  # URL base de su instancia
API_KEY = "tu_api_key"                       # API Key de la base
DB = "80200001_test"                         # Base de datos: _test o
   _prod

# === PASO 1: Inicializar sesion HTTP ===
# Necesario para que Element vincule la cookie de sesion
# con la base de datos especificada.
s = requests.Session()
s.get(f"{URL}/web", params={"db": DB})

# === PASO 2: Crear factura electronica ===
# El flujo completo es:
#   1. El servidor recibe el JSON
#   2. Valida autenticacion (API Key) y modo API activo
#   3. Resuelve partner: consulta SIFEN por RUC "80000001-1",
#      obtiene nombre y datos, crea el partner en Element
#   4. Resuelve producto: busca por default_code "78465748",
#      si no existe lo crea con los datos proporcionados
#   5. Crea la factura (account.move) con los IDs resueltos
#   6. Publica la factura automaticamente
#   7. Retorna los datos incluyendo numero y estado DE
resp = s.post(
    f"{URL}/api/fe/invoice",
    headers={
        "Authorization": f"Bearer {API_KEY}",
        "Content-Type": "application/json"
    },
    json={
        "partner_id": "48213",   # ID externo del cliente
        "api_ruc": "80000001-1",     # RUC para consulta SIFEN
        "lines": [{
            "product_id": "20458",
            "api_product_data": {
                "name": "Servicio",
                "default_code": "78465748",
                "uom_id": 1,       # Unidad de medida (Element ID)
                "tax_id": 1        # Impuesto (Element ID)
            },
            "quantity": 1,
            "price_unit": 100000   # Precio en guaranies
        }],
    }
)
```

```
# === PASO 3: Verificar resultado ===
result = resp.json()
if result.get("success"):
    data = result["data"]
    print(f"Factura creada: {data['name']}")
    print(f"Estado DE: {data.get('DE_estado', 'N/A')}")
    print(f"Total: {data['amount_total']} {data['currency']}")
else:
    error = result["error"]
    print(f"Error [{error['code']}]: {error['message']}")
```

Listing 28: Script minimo de prueba

15. Visualizacion de Resultados

Una vez ejecutado el script de creacion de facturas a credito, es importante verificar que la factura se haya generado correctamente en el sistema Element ERP. A continuacion se describe el proceso paso a paso para acceder y visualizar la factura creada.

15.1. Paso 1: Acceso al Modulo de Facturacion

Para visualizar las facturas generadas mediante la API, siga estos pasos:

1. Inicie sesion en Element ERP con sus credenciales
2. Navegue al menu principal y seleccione el modulo **Facturacion**
3. En el submenu, haga clic en **Clientes** y luego **Facturas**

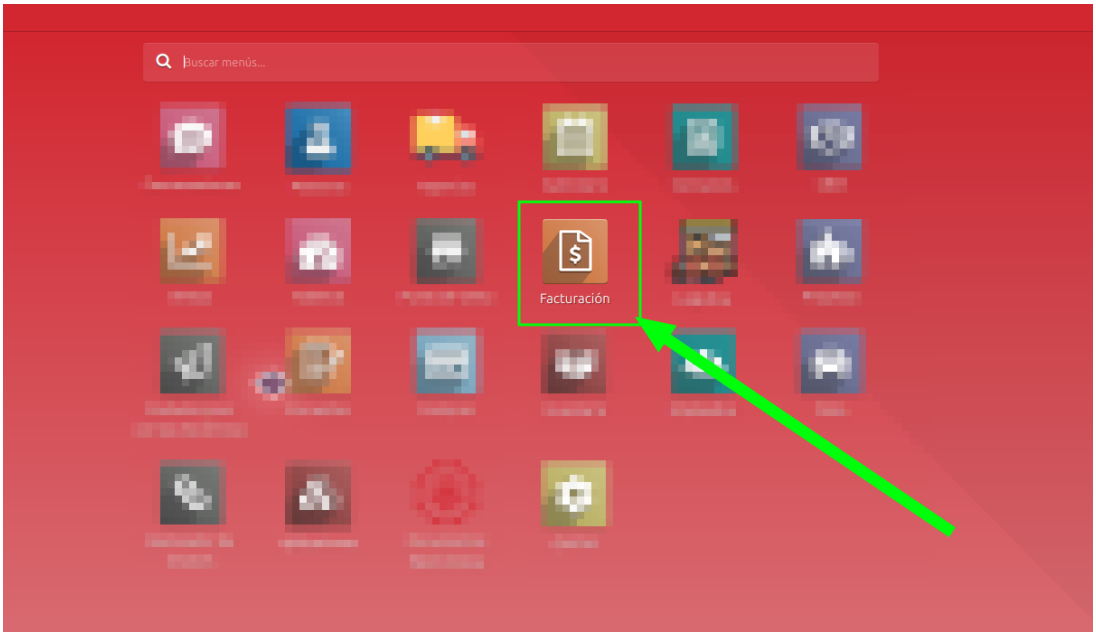

Figura 1: Acceso al modulo de facturacion en Element ERP

15.2. Paso 2: Localizacion y Acceso a la Factura

Una vez en el listado de facturas:

1. Busque la factura recien creada por su numero (retornado por la API)
2. Utilice los filtros para localizar facturas por fecha, cliente o estado
3. Haga clic en la factura para abrirla en modo de visualizacion

Tip de Navegacion

Las facturas generadas via API apareceran con estado **"Publicado"** y el termino de pago configurado sera visible en el campo correspondiente.

Figura 2: Listado de facturas mostrando la factura creada via API

15.3. Paso 3: Visualizacion del Codigo QR y XML

Dentro de la vista detallada de la factura, podra acceder a los elementos de facturacion electronica:

15.3.1. Visualizacion del Codigo QR

1. En la vista de la factura, busque la seccion **"Facturacion Electronica"**
2. El codigo QR se mostrara automaticamente en la factura
3. Puede hacer clic en el QR para ampliar o descargar la imagen

15.3.2. Acceso al XML Generado

1. Localice el campo **"DE_xml"** en la vista de la factura
2. Haga clic en el boton **"Ver XML"** o enlace correspondiente
3. El XML se abrira en una nueva ventana o se descargara automaticamente

XML como base para el ticket

El XML generado por el sistema contiene todos los datos fiscales y comerciales del documento electronico (emisor, receptor, items, totales, CDC, etc.). Este XML puede utilizarse como fuente de datos para generar el **ticket o comprobante impreso** de la factura electronica desde su sistema externo, extrayendo los campos necesarios para armar el formato de impresion deseado (ticket termico, A4, media carta, etc.) sin necesidad de consultar datos adicionales al servidor.
El campo `DE_xml` se retorna codificado en Base64 tanto en la respuesta de creacion de la factura como al consultar una factura existente mediante el endpoint `GET /api/fe/invoice/{numero}`.

DE_url: enlace de verificacion publica

El campo `DE_url` contiene la URL publica del portal ekuatia de la DNIT que permite verificar la autenticidad del documento electronico. Este enlace es unico por documento y puede compartirse directamente con el cliente como comprobante de validez ante la autoridad tributaria.
El campo solo tiene valor cuando `DE_estado = .ªprobado"`. Se retorna tanto en la respuesta de creacion de la factura (`POST /api/fe/invoice`) como al consultar la factura con `GET /api/fe/invoice/{numero}`.

Verificacion de Datos Electronicos

Elementos clave a verificar:

- **Estado**: Factura debe estar en estado "Publicado"
- **QR Code**: Visible y escaneble
- **XML**: Archivo descargable con estructura correcta
- **URL QR**: Campo DE_url debe contener enlace valido

15.4. Verificacion de Integridad

Para asegurar que la factura fue creada correctamente mediante la API, verifique los siguientes elementos:

- **Datos del cliente**: Correctamente asignados segun el partner_id
- **Lineas de factura**: Productos, cantidades y precios coinciden
- **Termino de pago**: Refleja el payment_term_id configurado
- **Totales**: Montos calculados correctamente
- **Estado**: Factura en estado "Publicado"(no borrador)
- **Elementos electronicos**: QR y XML generados exitosamente

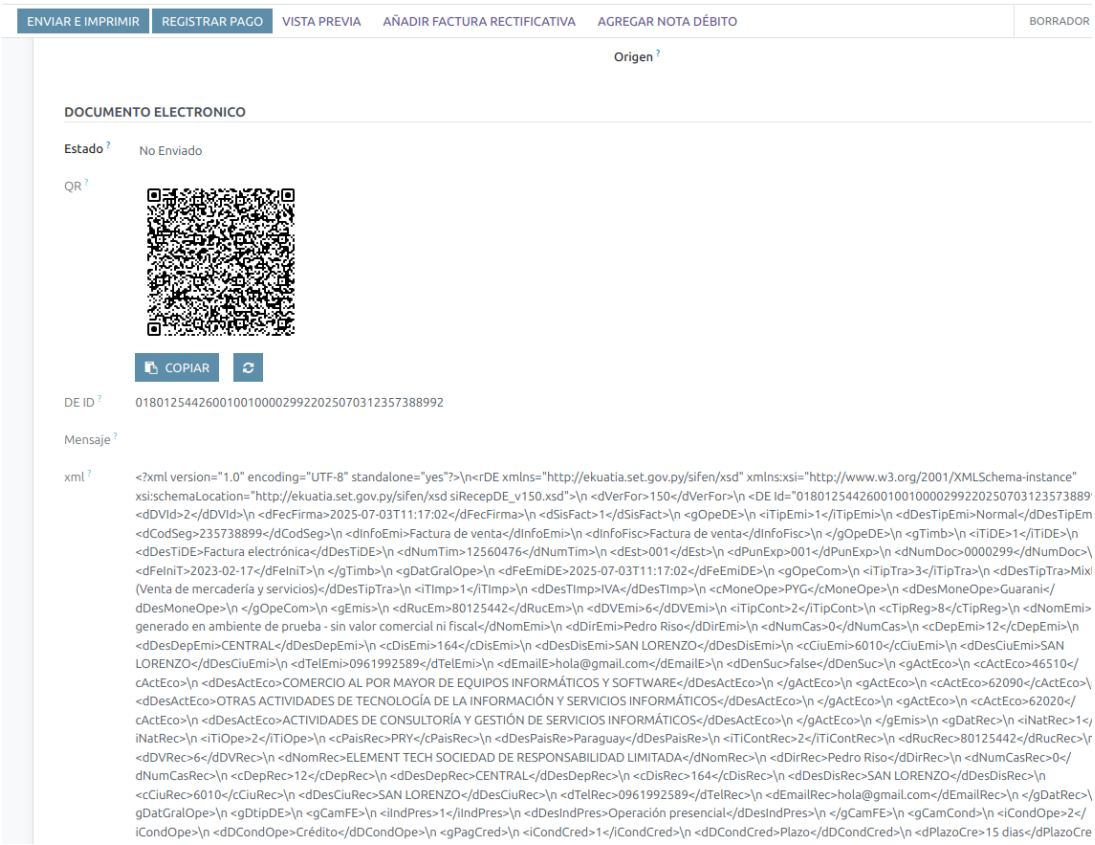

Figura 3: Vista detallada de la factura mostrando QR y acceso al XML