

InterfazWEB de Sphinx

La interfazWEB: cómo un sistema externo —una tienda en línea, un marketplace, un sistema de despacho o un tablero de gestión— lee el catálogo de Sphinx y le registra ventas.

Catálogo y maestros

Productos, precio y stock

Ventas y pagos

Boletas y despachos

Reportes de venta y caja

```
$ curl -H "access_token: $CLAVE" \  
    https://erp.miempresa.cl/interfazWEB/miempresa/Marcas  
  
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "Marcas",  
  "resultado": { "marcas": [ { "id_marca": 1, "marca": "Andina" } ] } }
```

Contenido

1 Introducción

1.1 Qué es y para quién

1.2 Qué cambió desde la versión 4.1

2 Cómo conectarse

2.1 La dirección de un servicio

2.3 Parámetros y cuerpo

2.2 La clave

2.4 Límites

3 La respuesta

3.1 El sobre

3.3 Códigos de respuesta

3.2 Códigos HTTP

3.4 Formatos

4 Catálogo y maestros

4.1 Check

4.5 Marcas

4.2 Sucursales

4.6 DivisionPolitica

4.3 Listas

4.7 Familias

4.4 Formas

4.8 WEB_Datos

5 Productos

5.1 Productos

5.3 ProductoSucursal

5.2 Producto

5.4 Las fotos: URL o base64

6 Ventas

6.1 Documento

6.3 Documento_Anular

6.2 Documento_Voucher

7 Consultas para el cliente final

7.1 Boleta

7.3 OrdenServicio

7.2 Despacho

8 Reportes

8.1 Estadísticas

8.3 Cuadratura

8.2 Usuarios

8.4 ReporteDiario

9 Procesos recomendados

9.1 Sincronizar el catálogo

9.3 Antes de salir a producción

9.2 Registrar una venta

1 Introducción

La interfazWEB va a quedar obsoleta. La reemplaza la **API de Sphinx v1** (`/api/v1/...`), con claves por sistema, códigos HTTP de verdad y el producto por partes, y tiene su propio manual en **sphinx.cl/manual**. Las integraciones nuevas deben hacerse con ella; las que usan la interfazWEB siguen funcionando mientras se cambian, y la empresa les avisará la fecha en que se apaga.

1.1 Qué es y para quién

La **interfazWEB** es la API REST con que un sistema externo conversa con el ERP Sphinx de una empresa. Con ella se puede:

- **leer el catálogo** —sucursales, marcas, familias, formas de pago y de entrega, productos con su precio, su stock por sucursal y sus fotos— para publicarlo en una tienda en línea o en otro canal;
- **registrar ventas** —crear el pedido o el documento, informar el pago y anularlo si el pago no llega—, de modo que el stock se reserve en el mismo momento y la boleta o factura se emita sola;
- **consultar** una boleta electrónica, el estado de un despacho o las órdenes de servicio de un cliente, y **leer reportes** de venta y de caja.

Este manual está escrito para quien programa esa integración. Cada servicio se documenta con su dirección, sus parámetros, los campos de su respuesta y un ejemplo tomado de un Sphinx real.

Una API, un ERP. Cada empresa tiene su propio Sphinx, en su propio servidor y con su propia clave. Todo lo que aquí se describe se hace contra el Sphinx de esa empresa: la dirección y la clave las entrega Sphinx al habilitar la integración (ver 2.2).

1.2 Qué cambió desde la versión 4.1

Los servicios, sus rutas y la forma de la respuesta son los mismos del documento anterior: una integración que funciona con la versión 4.1 sigue funcionando. Lo que cambia es que ahora el manual dice lo que el sistema efectivamente hace:

- **La base va en la ruta** (`/interfazWEB/<base>/<Servicio>`) y la clave es la de esa base. En un servidor con varias empresas la forma corta atiende siempre la primera (2.1).
- **Tope de peticiones:** 30 por segundo por IP, con ráfagas de hasta 60; lo que pasa de ahí se contesta con HTTP 429 (2.4).
- **Códigos de respuesta** completos y al día, incluidos los HTTP (3). El servicio que no existe contesta código 7, no 1.
- **Formas:** `id_formapago` es un número (el documento anterior mostraba una sigla); cada forma de pago dice qué precio le corresponde y cada forma de entrega si es despacho.
- **Productos con fecha** devuelve una lista simple de códigos, sin la fecha de modificación.
- **Producto** trae bastante más: los precios neto, con débito, con tarjeta y de oferta; el stock por bodega; las variantes; los productos adicionales; las fichas técnicas; y las fotos en dos tamaños más las de la descripción. Además se aclara el tope de 100 productos, 30 segundos o 16 MB por llamada (5.2).

- **WEB_Datos** ya no trae la sucursal, la lista ni la tasa de IVA; trae la imagen de «no disponible», avisos, tipos de destacado, motivos de contacto y las páginas de aterrizaje.
- **Documento:** se documentan las reglas con que Sphinx valida los totales, qué campos son obligatorios, cómo se evitan las ventas duplicadas con `nroOrden` y los mensajes de error. El parámetro `termica` del documento anterior nunca tuvo efecto y se retiró del manual. En el `detalle`, el producto se identifica por `idProducto`.
- **Documento_Voucher** y **Documento_Anular:** qué se puede pagar y qué se puede anular, y qué pasa si el pago llega dos veces.
- **Estadísticas** acepta un rango de hasta 120 días y devuelve 100 filas por página.

2 Cómo conectarse

2.1 La dirección de un servicio

```
https://<servidor>/interfazWEB/<base>/<Servicio>[/<param1>/<param2>/...] [?parámetro=valor]
```

Parte	Qué es
<servidor>	El nombre del Sphinx de la empresa, el mismo con que sus usuarios entran al sistema. Siempre por HTTPS : por HTTP el servidor redirige y la clave viajaría en claro.
<base>	El nombre de la base de datos de la empresa en ese servidor. Lo entrega Sphinx junto con la clave.
<Servicio>	El nombre del servicio, respetando mayúsculas : Producto , no producto .
/param1/...	Los parámetros de ruta que pida el servicio, en el orden que indica cada uno.

La forma corta, sin la base (/interfazWEB/<Servicio>), sigue funcionando y atiende la **primera** base del servidor. Sirve sólo donde el servidor tiene una única empresa; como no se puede saber si mañana tendrá otra, se recomienda escribir siempre la base.

2.2 La clave

Cada base tiene su propia clave de la API. La emite Sphinx cuando la empresa contrata la integración y se pide a **soporte@sphinx.cl**. La clave de una base no sirve para otra, aunque estén en el mismo servidor. Se envía de una de estas dos formas:

Forma	Cómo
Encabezado access_token (recomendada)	access_token: <clave>
Autenticación básica HTTP	Usuario sphinx y la clave como contraseña: Authorization: Basic base64("sphinx:<clave>")

```
curl -H "access_token: $CLAVE" https://erp.miempresa.cl/interfazWEB/miempresa/Check
curl -u "sphinx:$CLAVE" https://erp.miempresa.cl/interfazWEB/miempresa/Check
```

Sin clave, o con una que no corresponde, la respuesta es **HTTP 401** con el sobre de siempre y código 7 (ver 3.2). Si la base no tiene la integración habilitada el mensaje es Sphinx no habilitado para Interfaz WEB.

La clave es de la empresa, no del programa. Debe vivir en el servidor de la integración y nunca en el navegador ni en una aplicación que se instale en el teléfono del cliente: quien la tenga puede leer el catálogo completo y crear ventas. Si se filtra, pida una nueva a soporte@sphinx.cl.

2.3 Parámetros y cuerpo

- **Método.** Los servicios de lectura se llaman con **GET**. Los que reciben datos (Documento , Documento_Voucher , Cuadratura , ReporteDiario) exigen **POST**, y con otro método contestan código 3. Los demás que reciben un JSON (Producto , Estadísticas , Documento_Anular) lo leen del cuerpo del POST.
- **Cuerpo.** JSON en UTF-8, con Content-Type: application/json. Los nombres de campo del cuerpo van en *camel/Case* (fechaDesde , idSucursal); los de la respuesta, en su mayoría, con guion bajo (id_sucursal).
- **Empresa.** Si la base tiene más de una empresa, el parámetro ?id_empresa=<n> elige cuál; sin él se usa la primera.
- **Codificación.** Un parámetro de ruta que lleve espacios o símbolos se codifica como en cualquier URL: 2026-09-01%2000:00:00.

2.4 Límites

Límite	Valor y qué pasa al superarlo
Peticiones por IP	30 por segundo, con ráfagas de hasta 60. Lo que pase de ahí recibe HTTP 429 con código 9 (Demasiadas peticiones, intente nuevamente en unos segundos). No es una cuota comercial sino una red contra un programa en bucle: una integración normal no llega nunca.
Tiempo de una petición	300 segundos. Una respuesta que tarde más se corta.
Productos por llamada	100 productos, 30 segundos de proceso o 16 MB de respuesta, lo que ocurra primero. Lo que no alcanzó se informa en faltaron (ver 5.2).
Reportes	Estadísticas , Cuadratura y ReporteDiario no se ejecutan dos veces a la vez para la misma empresa: la segunda llamada contesta código 8 y hay que reintentarla cuando termine la primera.

3 La respuesta

3.1 El sobre

Todos los servicios contestan el mismo sobre JSON:

```
{
  "codigo": 0,
  "mensaje": "Exitoso",
  "servicio": "Marcas",
  "resultado": { ... }
}
```

Campo	Tipo	Descripción
codigo	Integer	0 es éxito ; cualquier otro valor es un error (ver 3.3). Es lo primero que hay que mirar.
mensaje	String	Exitoso , o el motivo del error en palabras que se le pueden mostrar al usuario.
servicio	String	El servicio que contestó.
resultado	Object / Array	Los datos. No viene cuando el servicio no tiene nada que devolver ni cuando hubo un error.
interno	String	Sólo en algunos errores: el detalle técnico (por ejemplo, qué regla del XML no se cumplió). Sirve para diagnosticar, no para mostrar.

3.2 Códigos HTTP

Un error de negocio —un producto que no existe, un monto que no cuadra— se contesta con **HTTP 200** y el error en `codigo`. El código HTTP sólo cambia cuando la petición no llegó a ejecutarse:

HTTP	codigo	Cuándo
200	0 u otro	El servicio se ejecutó. Mirar <code>codigo</code> .
401	7	Falta la clave, no corresponde a la base, o la base no tiene la integración habilitada.
429	9	Se superó el tope de peticiones (2.4). Reintentar tras unos segundos.
502, 503, 504	—	El servidor está reiniciando o actualizándose. Reintentar en un minuto.

3.3 Códigos de respuesta

Los códigos que pueden devolver los servicios de este manual. El `mensaje` puede ser más preciso que el texto de la tabla (por ejemplo, Producto `[NB-001]` no existe con código 1).

Código	Significado	Lo devuelve
--------	-------------	-------------

0	Exitoso	Todos
1	Parámetros incorrectos, o un dato pedido que no existe	Todos los que reciben parámetros
3	El servicio requiere POST	Documento, Documento_Voucher, Cuadratura, ReporteDiario
4	El documento enviado no es válido (JSON o XML mal formado)	Documento, Documento_Voucher
6	Sphinx no está configurado para la venta web: falta la Sucursal WEB, o una sucursal tiene más de una bodega web	Check, Producto
7	Clave inválida (HTTP 401) o servicio que no existe (HTTP 200)	Todos
8	El servicio ya se está ejecutando para esta empresa	Estadísticas, Cuadratura, ReporteDiario
9	Demasiadas peticiones (HTTP 429)	Todos
10	El documento no se pudo grabar	Documento
12	El documento está anulado	Documento_Voucher
15	Se superó el máximo de cotizaciones del día	Documento
20	El pago no se pudo registrar	Documento_Voucher
25	El documento no se puede pagar	Documento_Voucher
42	El cliente está bloqueado por la empresa	Documento, Despacho, OrdenServicio
45	RUT no válido	Despacho, OrdenServicio
46	El cliente no existe	Despacho, OrdenServicio
60, 61, 62	Folio, fecha o monto de la boleta inválido	Boleta
69	La boleta no existe	Boleta
80, 81	Tipo o folio de documento inválido	Despacho
82	No hay despacho asociado	Despacho
99	Error interno. Si el mensaje es el genérico (Estimado, nuestro sistema presenta problemas...) el error quedó registrado y se puede reintentar; si trae un texto propio, es una validación y reintentar no cambia nada.	Todos
100	Rechazo de negocio, con el motivo en mensaje (ver cada servicio)	Documento, Documento_Voucher

3.4 Formatos

Dato	Formato
Fecha	AAAA-MM-DD — 2026-09-27
Fecha y hora	AAAA-MM-DD HH:MM:SS , en la hora de Chile — 2026-09-27 21:48:47

Marca de tiempo	Milisegundos desde 1970 (sólo en las fotos) — 1759012127000
Montos	Número decimal en pesos, IVA incluido salvo que el campo diga <code>neto</code> .
Indicadores	Entero <code>1</code> (sí) o <code>0</code> (no).
RUT	En la ruta, con dígito verificador y sin puntos: 12608204-5 . En el JSON, el número sin dígito (12608204) salvo donde se indique.
Nulos	Un campo sin valor viaja como <code>null</code> ; no se omite.

4 Catálogo y maestros

Los maestros que una tienda necesita tener antes de publicar productos. Todos son **GET**, no reciben parámetros y devuelven la tabla completa: lo que no venga ya no existe en Sphinx (o dejó de publicarse) y hay que sacarlo del lado de la integración.

4.1 Check

Verifica que la conexión, la clave y la configuración de venta web estén bien. Es la primera llamada de cualquier integración y la que conviene usar como chequeo de salud.

Ruta **GET** /interfazWEB/<base>/Check

Contesta código 0 sin resultado, o código 6 con lo que falta configurar (Falta Sucursal WEB, o una sucursal con más de una bodega para la venta web).

```
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "Check" }
```

4.2 Sucursales

Las sucursales que la empresa publica en la web y la bodega de cada una de la que se vende en línea.

Ruta **GET** /interfazWEB/<base>/Sucursales

Campo	Tipo	Descripción
sucursales	Array	Las sucursales activas y marcadas para la web.
id_sucursal	Integer	Código de la sucursal. Es el que usan el stock de Producto y el idSucursal de un documento.
sucursal	String	Nombre.
direccion	String	Calle, número y complemento en un solo texto.
id_comuna , comuna , ciudad	Int, String	Ubicación (códigos de DivisionPolitica).
fono	String	Teléfono.
latitud , longitud	String	Coordenadas, para un mapa.
horario	String	Horario de atención, en texto libre.
maps_embed	String	HTML para incrustar un mapa, tal como se pegó en Sphinx.
web	Integer	Siempre 1 (sólo vienen las publicadas).
descripcion , tipo_tienda	String	Textos libres para describir la tienda.
orden	Integer	Orden en que mostrarlas.

imagen, path_imagen	String	Foto de la sucursal en base64 y su nombre de archivo. Sólo si tiene foto; el nombre cambia cada vez que se cambia la foto.
bodegas	Array	La bodega de venta web de cada sucursal.
id_bodega, id_sucursal, bodega, web	Int, Int, String, Int	Código, sucursal a la que pertenece, nombre, y 1.

```
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "Sucursales",
  "resultado": {
    "sucursales": [ { "id_sucursal": 1, "sucursal": "Matriz", "direccion": "La calle 123 #123",
      "id_comuna": 307, "comuna": "SANTIAGO", "ciudad": "Santiago", "fono": "+56
9 1234 1234",
      "latitud": "-33.4708517", "longitud": "-70.636261",
      "horario": "Lunes a Sábado - 10:00 a 14:00 y 15:00 a 18:30",
      "maps_embed": "<iframe ...></iframe>", "web": 1, "descripcion": "",
      "tipo_tienda": "", "orden": null } ],
    "bodegas": [ { "id_bodega": 1, "id_sucursal": 1, "bodega": "Bodega", "web": 1 } ] } }
```

4.3 Listas

Las listas de precios activas y publicadas en la web. Un producto trae un precio por cada una.

Ruta **GET** /interfazWEB/<base>/Listas

Campo	Tipo	Descripción
listas	Array	
id_lista	Integer	Código de la lista.
lista	String	Nombre (Publico , Mayorista ...).

```
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "Listas",
  "resultado": { "listas": [ { "id_lista": 1, "lista": "Publico" } ] } }
```

4.4 Formas

Las formas de pago, las formas de entrega y los transportes que la empresa ofrece en la web. Sus códigos son los que se envían al crear un Documento.

Ruta **GET** /interfazWEB/<base>/Formas

Campo	Tipo	Descripción
pagos	Array	Formas de pago publicadas.
id_formapago	Integer	Código. Va en idFormapago del documento.
formapago	String	Nombre.

precio	String	Qué precio del producto corresponde cobrar con esta forma de pago: precio (el normal), precio_cash (contado, con descuento) o precio_tarjeta (con recargo). Es el nombre del campo en los precios de Producto .
imagen, path_imagen	String	Logo en base64 y su nombre de archivo, si tiene.
entregas	Array	Formas de entrega publicadas.
id_formaentrega	Integer	Código. Va en idFormaentrega del documento.
formaentrega	String	Nombre.
despacho	Integer	1 si es despacho a domicilio (el documento debe traer la dirección de despacho); 0 si es retiro.
imagen, path_imagen	String	Logo, si tiene.
transportes	Array	Empresas de transporte.
id_transporte, transporte	Int, String	Código y nombre.

```
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "Formas",
  "resultado": {
    "transportes": [ { "id_transporte": 20, "transporte": "ChileExpress" } ],
    "pagos": [ { "id_formapago": 9, "formapago": "Transferencia Bancaria", "precio": "precio" },
               { "id_formapago": 6, "formapago": "WebPay", "precio": "precio" } ],
    "entregas": [ { "id_formaentrega": 3, "formaentrega": "Despacho a Domicilio", "despacho": 1
                  { "id_formaentrega": 4, "formaentrega": "Retiro en Tienda", "despacho": 0 } ] } ] }
```

4.5 Marcas

Ruta **GET** /interfazWEB/<base>/Marcas

Campo	Tipo	Descripción
marcas	Array	Todas las marcas.
id_marca	Integer	Código; es el id_marca del producto.
marca	String	Nombre (NN si no tiene).
imagen o url_imagen	String	El logo, en base64 o como dirección según cómo publique las fotos la empresa (ver 5.4). Sólo si tiene logo.
path_imagen	String	Nombre de archivo del logo; cambia cuando se cambia el logo.

```
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "Marcas",
  "resultado": { "marcas": [ { "id_marca": 1, "marca": "Andina" }, { "id_marca": 5, "marca": "Austral" } ] } }
```

4.6 DivisionPolitica

Las regiones y comunas de Chile con los códigos que Sphinx usa en las direcciones. El `idComuna` de un cliente o de un despacho tiene que ser uno de éstos.

Ruta **GET** /interfazWEB/<base>/DivisionPolitica

Campo	Tipo	Descripción
regiones	Array	16 regiones.
id_region , region	Int, String	Código y nombre.
comunas	Array	346 comunas.
id_comuna , id_region , comuna	Int, Int, String	Código, región a la que pertenece y nombre.

```
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "DivisionPolitica",  
  "resultado": {  
    "regiones": [ { "id_region": 1, "region": "Tarapaca" }, { "id_region": 13, "region":  
"Metropolitana" }, ... ],  
    "comunas": [ { "id_comuna": 272, "id_region": 11, "comuna": "AISEN" }, { "id_comuna": 322,  
"id_region": 13, "comuna": "NUÑOA" }, ... ] } }
```

4.7 Familias

El árbol de familias (categorías) publicadas en la web. Cada familia puede traer sus subfamilias en `familias`, a cualquier profundidad.

Ruta **GET** /interfazWEB/<base>/Familias

Campo	Tipo	Descripción
familias	Array	Las familias de primer nivel. <code>null</code> si no hay ninguna publicada.
id_familia	Integer	Código; es el <code>id_familia</code> del producto.
familia	String	Nombre.
orden	Integer	Orden en que mostrarlas.
url	String	Una ruta amigable sugerida, armada con el nombre de la familia y el de sus padres (/computacion-notebooks).
id_familia_tabulacion	Integer	La familia cuya ficha técnica usan sus productos (ver <code>tabulacion</code> en <code>Producto</code>), o <code>null</code> .
familias	Array	Subfamilias, con los mismos campos. No viene si no tiene.

```
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "Familias",
  "resultado": { "familias": [
    { "id_familia": 1, "id_familia_tabulacion": null, "familia": "Computacion", "orden": 1,
      "url": "/computacion",
      "familias": [ { "id_familia": 3, "id_familia_tabulacion": null, "familia": "Notebooks",
        "orden": 1,
        "url": "/computacion-notebooks" } ] },
    { "id_familia": 2, "id_familia_tabulacion": null, "familia": "Television", "orden": 2, "url":
      "/television" } ] } }
```

4.8 WEB_Datos

El contenido del sitio que se administra en Sphinx, en la pantalla Web: banners, imbatibles, avisos, destacados, logo del encabezado y páginas de aterrizaje. Sólo tiene sentido para una tienda que tome ese contenido de Sphinx.

Ruta **GET** /interfazWEB/<base>/WEB_Datos

Campo	Tipo	Descripción
url_nodisponible	String	La imagen que se muestra cuando un producto no tiene foto.
tablas	Object	Una lista por tabla:
web_banner	Array	id_banner, id_tipo, id_target, banner (el enlace o los productos), precio, id_grupo, y la imagen de escritorio y de móvil.
web_imbatible	Array	id_imbatible, id_tipo, id_target, producto, precio y las dos imágenes.
web_logoheader	Array	id_logoheader y las dos imágenes.
web_aviso	Array	id_aviso, aviso: los mensajes de la barra de avisos.
web_destacado	Array	id_destacado, id_tipo, id_producto: los productos de cada vitrina.
web_destacado_tipo	Array	id_tipo, tipo, modo: las vitrinas (Ofertas de la Semana, Nuevos...).
web_landing, web_landing_datasource, web_landing_datasource_producto	Array	Las páginas de aterrizaje, sus bloques y los productos de cada bloque.
web_banner_grupo, web_target, web_tipo, web_contacto_motivo	Array	Las tablas de códigos que usan las anteriores: grupos de banner, destino del enlace (Self / Blank), tipo de banner (Link / Productos) y motivos del formulario de contacto.

Cada imagen viaja como imagen / imagen_mobile en base64 o como url_imagen / url_imagen_mobile, según cómo publique las fotos la empresa (5.4), más su nombre de archivo en path_imagen / path_imagen_mobile.

5 Productos

Los productos se sincronizan en dos pasos: **Productos** dice cuáles hay y cuándo cambió cada uno, y **Producto** entrega el detalle de los que hay que actualizar (ver 9.1).

Un producto se publica cuando está marcado para la web y su familia también lo está.

5.1 Productos

Los códigos de los productos publicados, con la fecha de su última modificación, del más reciente al más antiguo.

Ruta	GET /interfazWEB/<base>/Productos
Con fecha	GET /interfazWEB/<base>/Productos/<AAAA-MM-DD HH:MM:SS>

Sin fecha devuelve todos los productos publicados, cada uno con su `codigo` y su `fecha_modificacion`.

Con fecha devuelve sólo los modificados desde ese momento, y como una lista simple de códigos.

Campo	Tipo	Descripción
productos	Array	Sin fecha: objetos con los dos campos de abajo. Con fecha: códigos (String).
codigo	String	Código del producto (SKU). Es lo que se le pide a Producto .
fecha_modificacion	String	Fecha y hora de la última modificación de la ficha, el precio o el stock.

```
GET /interfazWEB/miempresa/Productos
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "Productos",
  "resultado": { "productos": [ { "codigo": "NB-001", "fecha_modificacion": "2026-09-27 21:48:47"
                                },
                                { "codigo": "TV-002", "fecha_modificacion": "2026-09-26 10:02:13"
                                },
                                ... ] } } }

GET /interfazWEB/miempresa/Productos/2026-09-27%2000:00:00
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "Productos",
  "resultado": { "productos": [ "NB-001", "COM-010", "TV-002" ] } } }
```

La lista con fecha no avisa de los que se despublicaron. Un producto que se saca de la web deja de aparecer en la lista completa, pero no aparece en la de cambios. Para enterarse, compare de vez en cuando la lista completa con la suya (9.1).

5.2 Producto

El detalle completo de uno o varios productos: ficha, precios, stock y fotos.

Ruta	POST /interfazWEB/<base>/Producto
Cuerpo	Un arreglo JSON con los códigos: ["NB-001", "TV-002"]
Alternativa	GET /interfazWEB/<base>/Producto/NB-001/TV-002 — cómoda para probar; la dirección no debe pasar de unos 1.000 caracteres.
Tope	100 productos, 30 segundos de proceso o 16 MB de respuesta por llamada, lo que ocurra primero.

Si un código no existe, la llamada entera contesta código 1 (Producto [XX-999] no existe) sin detalle de los demás. Si la respuesta se corta por el tope, faltaron dice cuántos quedaron fuera: son los últimos del arreglo enviado y se piden en otra llamada.

Los productos se reconocen por su código (SKU). En las pocas empresas que no usan código, el identificador es el `id_producto`.

La respuesta

Campo	Tipo	Descripción
total	Integer	Productos que vienen en la respuesta.
faltaron	Integer	Productos pedidos que no alcanzaron a venir.
productos	Array	Un objeto por producto, con los campos que siguen.

Identificación y ficha

Campo	Tipo	Descripción
id_producto	Integer	Identificador interno.
codigo	String	Código (SKU).
nombre	String	Nombre.
descripcion	String	Descripción en HTML. Las imágenes que tenga adentro apuntan al sitio de fotos de la empresa (ver <code>fotos_ficha</code>).
texto	String	Palabras para el buscador: código, marca, familia y nombre, en minúsculas.
url	String	Ruta amigable sugerida: /producto/notebook-14-i5-8gb-p17.
meta_title, meta_keywords, meta_description	String	Datos para buscadores (SEO).
id_marca	Integer	Marca (Marcas).
id_familia, id_familia_2, id_familia_3	Integer	Familia principal y hasta dos familias adicionales en que también aparece (Familias).
id_familia_tabulacion	Integer	Familia cuya ficha técnica usa.
partno, upc	String	Número de parte del fabricante y código de barras.

garantia	Integer	Meses de garantía.
dim_peso, dim_alto, dim_ancho, dim_largo	Double	Peso en kilos y medidas en centímetros, para calcular el despacho.
weblink, webvideo	String	Enlace al sitio del fabricante y a un video.
ind_exento	Integer	1 si el producto está exento de IVA.
fecha_creacion, fecha_modificacion	String	Fecha y hora. La de modificación es la misma de Productos : guárdela para saber si su copia está al día.

Publicación y disponibilidad

Campo	Tipo	Descripción
lista	Integer	1 si el producto se puede mostrar y vender; 0 si no. Es 0 cuando el producto o alguna de sus familias no está publicada, o cuando el stock no alcanza el mínimo que la empresa exige para vender en la web (salvo que venga en camino o esté en preventa, si la empresa lo permite).
id_disponibilidad	Integer	0 Stock, 1 Kit, 2 Servicio, 3 Kit fijo, 99 A pedido. Sólo 0, 3 y 99 descuentan stock.
nuevo, liq, out, preventa	Integer	Marcas comerciales (1/0): producto nuevo, en liquidación, <i>outlet</i> y en preventa.
en_camino	Integer	1 si hay una orden de compra pendiente de recibir con este producto.
api	Integer	1 si lo que se publica (stock y precio) es la oferta de un proveedor mayorista y no stock propio.
id_mayorista	Integer	El mayorista de esa oferta, cuando api es 1.

Precios

`precios` trae un objeto por cada [lista de precios](#) publicada. Todos los montos son en pesos, con IVA salvo los `_neto`; los que la empresa no usa vienen en `null`.

Campo	Tipo	Descripción
id_lista	Integer	Lista de precios.
precio / precio_netto	Double	Precio normal.
precio_cash / precio_cash_netto	Double	Precio contado (transferencia, efectivo).
precio_debito / precio_debito_netto	Double	Precio con tarjeta de débito.
precio_tarjeta / precio_tarjeta_netto	Double	Precio con tarjeta de crédito.
precio_oferta / precio_oferta_netto	Double	Precio de oferta vigente, si hay.
precio_debito_oferta, precio_tarjeta_oferta (y sus <code>_netto</code>)	Double	La oferta con débito y con tarjeta.
desde, hasta	String	Vigencia de la oferta.

precio_dolar	Double	El precio neto en dólares, al tipo de cambio del día.
---------------------	--------	---

Qué precio cobrar según la forma de pago lo dice el campo `precio` de cada forma de pago en [Formas](#).

Stock

Campo	Tipo	Descripción
stock	Array	Por sucursal publicada: <code>id_sucursal</code> , <code>stock</code> (Double).
stock_bodega	Array	Por bodega de venta web: <code>id_bodega</code> , <code>stock</code> .

Variantes, relacionados y ficha técnica

Campo	Tipo	Descripción
id_grupo	Integer	El grupo de variantes al que pertenece (una polera en varias tallas y colores), o <code>null</code> .
grupo	Object	Sólo si tiene grupo: <code>id_grupo</code> , <code>nombre</code> y <code>etiqueta1</code> ... <code>etiqueta3</code> , los nombres de los ejes (<code>Talla</code> , <code>Color</code>).
opcion1, opcion2, opcion3	String	El valor de este producto en cada eje (<code>M</code> , <code>Rojo</code>).
relacionados	Array	<code>id_producto</code> , <code>codigo</code> : productos sugeridos junto a éste.
adicionales	Array	<code>id_producto</code> , <code>codigo</code> : productos que se ofrecen como agregado (una garantía extendida, una instalación).
tabulacion, tabulacion_opc	Array	La ficha técnica: <code>id_detalle</code> , <code>id_familia</code> , <code>detalle</code> (la característica) y <code>detalle_valor</code> (su valor). La segunda son características opcionales.

Fotos

Campo	Tipo	Descripción
fotos	Array	Las fotos del producto en tamaño completo, en el orden en que se muestran.
fotos_200	Array	Las mismas en miniatura (200 px), para listados.
fotos_ficha	Array	Las imágenes que aparecen dentro de la <code>descripcion</code> .
id	String	Identificador de la foto.
tipo	String	Extensión: <code>jpg</code> , <code>png</code> , <code>webp</code> .
timestamp	Long	Cuándo se subió. Si no cambió, no hace falta volver a bajarla.
url o base64	String	La foto, según cómo la publique la empresa (5.4).

Ejemplo

```
POST /interfazWEB/miempresa/Producto
["NB-001"]

{ "codigo": 0, "mensaje": "Exitoso", "servicio": "Producto",
  "resultado": { "total": 1, "faltaron": 0, "productos": [ {
    "id_producto": 17, "codigo": "NB-001", "nombre": "Notebook 14 i5 8GB", "descripcion": "<p>...
</p>",
    "lista": 1, "nuevo": 1, "preventa": 0, "liq": 0, "out": 0, "id_disponibilidad": 0,
    "id_familia": 3, "id_familia_2": null, "id_familia_3": null, "id_familia_tabulacion": null,
    "id_marca": 1,
    "dim_peso": 1.6, "dim_alto": 3.0, "dim_ancho": 33.0, "dim_largo": 23.0, "garantia": 12,
    "partno": null, "upc": null, "texto": "nb 001 notebooks andina notebook 14 i5 8gb",
    "meta_title": null, "meta_keywords": null, "meta_description": null, "weblink": null,
    "webvideo": null,
    "ind_exento": null, "url": "/producto/notebook-14-i5-8gb-p17",
    "fecha_creacion": "2026-09-27 21:45:33", "fecha_modificacion": "2026-09-27 21:48:47",
    "id_grupo": null, "opcion1": null, "opcion2": null, "opcion3": null,
    "api": 0, "id_mayorista": null, "en_camino": 0,
    "stock": [ { "id_sucursal": 1, "stock": 125.0 } ],
    "stock_bodega": [ { "id_bodega": 1, "stock": 125.0 } ],
    "precios": [ { "id_lista": 1, "precio": 618800.0, "precio_netto": 520000.0,
      "precio_cash": 599990.0, "precio_cash_netto": 504193.0,
      "precio_debito": null, "precio_debito_netto": null, "precio_tarjeta": null,
    "precio_tarjeta_netto": null,
      "precio_oferta": null, "precio_oferta_netto": null, "desde": null, "hasta":
    null,
      "precio_debito_oferta": null, "precio_debito_oferta_netto": null,
      "precio_tarjeta_oferta": null, "precio_tarjeta_oferta_netto": null,
    "precio_dolar": 567.33 } ],
    "relacionados": [ { "id_producto": 18, "codigo": "NB-002" } ], "adicionales": [],
    "tabulacion": [ { "id_detalle": 4, "id_familia": 3, "detalle": "Memoria", "detalle_valor": "8
GB" } ],
    "tabulacion_opc": [],
    "fotos": [ { "id": "1", "tipo": "webp", "timestamp": 1759012127000,
      "url": "https://static.miempresa.cl/fotos/17/1/1759012127000.webp" } ],
    "fotos_200": [ ... ], "fotos_ficha": [ ] } ] } }
```

5.3 ProductoSucursal

Los productos publicados de una familia (y opcionalmente de una marca), con el mismo detalle de [Producto](#) y paginados. Pensado para mostrar una categoría de una sucursal sin tener el catálogo copiado.

Ruta **GET** /interfazWEB/<base>/ProductoSucursal/<id_sucursal>/<id_familia>[/<id_marca>][?page=<n>]

La sucursal y la familia son obligatorias (sin ellas, código 1). Sólo vienen los productos que se pueden vender (lista = 1).

Campo	Tipo	Descripción
producto	Array	Los productos, con los campos de Producto .
paginacion	Object	registros (total), pagina (la actual) y paginas (cuántas hay).

```
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "ProductoSucursal",  
  "resultado": { "producto": [ { "id_producto": 20, "codigo": "NB-004", "nombre": "Notebook 13  
Ultraliviano", ... } ],  
    "paginacion": { "registros": 5, "pagina": 1, "paginas": 1 } } }
```

5.4 Las fotos: URL o base64

Cada empresa elige en Sphinx (Web > Parámetros, grupo Fotos) cómo entrega sus imágenes, y la elección vale para todos los servicios:

Forma	Qué llega
Por URL (recomendada)	El campo <code>url</code> (o <code>url_imagen</code>) con una dirección pública y definitiva de la foto. La integración puede enlazarla directamente o bajarla una vez. La dirección cambia cada vez que la foto cambia, así que se puede guardar en caché sin límite.
En base64	El campo <code>base64</code> (o <code>imagen</code>) con el archivo completo codificado. Hace las respuestas mucho más pesadas —un producto con varias fotos puede pesar varios MB— y es la razón del tope de 16 MB de Producto .

Las fotos de sucursales y de formas de pago y entrega viajan siempre en base64. Para no bajar la misma foto dos veces, compare el `timestamp` (productos) o el `path_imagen` (el resto) con lo que ya tiene.

6 Ventas

Una venta en línea se registra en Sphinx en dos o tres llamadas: **Documento** al confirmar el carro (reserva el stock), **Documento_Voucher** cuando el medio de pago aprueba, y **Documento_Anular** si el pago no llega. El recorrido completo está en 9.2.

6.1 Documento

Crea el pedido o el documento de venta. Desde este momento el stock queda reservado.

Ruta	POST /interfazWEB/<base>/Documento[?correo=1]
Cuerpo	El documento en JSON, con Content-Type: application/json.
correo	Si viene, y en esta misma llamada se emitió la boleta o factura, se le envía al cliente por correo.

Qué documento se crea

Lo decide `tipo` junto con la configuración de la empresa. Casi todas las empresas trabajan con **Pedido Web** (es lo que viene configurado):

tipo	Con Pedido Web (<i>lo habitual</i>)	Sin Pedido Web
PEDIDO	Pedido Web.	Pedido Web.
BOLETA	Pedido Web, que la empresa convertirá en boleta.	Boleta.
FACTURA	Pedido Web, que la empresa convertirá en factura.	Factura (exige los datos de facturación del cliente).
COTIZACION	Cotización. No reserva stock. Un mismo RUT o IP puede pedir hasta 15 al día (código 15).	

El Pedido Web reserva el stock y queda esperando el pago; la boleta o la factura la emite la empresa desde Sphinx cuando lo despacha. Si el pedido no se paga dentro de su plazo de vigencia —unos 20 minutos con un medio de pago en línea, dos días hábiles con transferencia—, Sphinx lo anula solo y libera el stock.

El cuerpo

```
{
  "tipo": "BOLETA",
  "nroOrden": 51259,
  "idSucursal": 1,
  "idFormapago": 6,
  "idFormaentrega": 3,
  "rut": 21614514,
  "observaciones": "Dejar en conserjería",
  "ipAddress": "190.45.205.61",
  "neto": 338639,
  "descuento": 0,
  "afecto": 338639,
  "iva": 64341,
  "total": 402980,
  "despacho": 5990,
  "cliente": {
    "nombre": "María Pérez Soto",
    "email": "maria@correo.cl",
    "fono": "+56 9 8765 4321",
    "direccion": "Los Aromos", "numero": "123", "adicional": "Depto 101",
    "idComuna": 322, "ciudad": "Santiago"
  },
  "despachoDomicilio": {
    "nombre": "María Pérez Soto", "fono": "+56 9 8765 4321",
    "direccion": "Los Aromos", "numero": "123", "adicional": "Depto 101",
    "idComuna": 322, "ciudad": "Santiago", "idTransporte": 20,
    "nota": "Dejar en conserjería"
  },
  "detalle": [
    { "idProducto": 17, "cantidad": 1, "precio": 396990, "descuento": 0,
      "descripcion": "Notebook 14 i5 8GB" }
  ]
}
```

Encabezado

Campo	Tipo	Oblig.	Descripción
tipo	String	Sí	PEDIDO, BOLETA, FACTURA o COTIZACION (ver arriba).
nroOrden	Long	Sí	El número de la orden en su sistema. Es lo que evita las ventas duplicadas: si se reenvía la misma orden con el mismo total, Sphinx contesta éxito con el documento que ya existe, sin crear otro (ver abajo). En un Pedido Web, es además su folio.
idSucursal	Integer	Sí*	Sucursal que vende (Sucursales). *Con tipo PEDIDO puede omitirse y se usa la sucursal web.
idBodega	Integer	No	Bodega de la que sale el stock. Por omisión, la bodega web de la sucursal.
idFormapago	Integer	Sí	Forma de pago (Formas). Si es un medio en línea (WebPay, MercadoPago), la respuesta trae id_pago.

idFormaentrega	Integer	No	Forma de entrega (Formas). Si es despacho, envíe <code>despachoDomicilio</code> .
idLista	Integer	No	Lista de precios contra la que se validan precios y descuentos. Por omisión, la de la web.
rut	Integer	Ver nota	RUT del comprador, sin dígito verificador . Una boleta sin RUT se emite a «Cliente sin nombre»; una factura lo exige.
rutRetiro	Integer	No	RUT de quien retira, si es otra persona (su nombre va en <code>cliente.nombreRetiro</code>).
neto, descuento, afecto, iva, total	Double	Sí	Los totales, en pesos enteros. Se validan, no se recalculan (ver abajo).
despacho	Double	No	Costo del despacho que se le cobra al cliente, con IVA .
observaciones	String	No	Nota del pedido (hasta 2.048 caracteres).
ordenCompra	String	No	Orden de compra del cliente, si la trae.
ipAddress	String	No	IP del comprador.
idWeb	String	No	Un identificador propio de la orden (hasta 128 caracteres). Funciona como <code>nroOrden</code> para evitar duplicados; si no se envía, Sphinx genera uno y lo devuelve.
urlRetorno	String	No	Dónde volver después de pagar en línea. Si viene, se envía el correo aunque no se pida <code>? correo</code> .
origen	String	No	Por omisión <code>WEB</code> , que activa las validaciones de una venta en línea (stock, correo del cliente, descuentos máximos). No lo cambie.

Cliente (`cliente`)

Con `rut`, Sphinx crea el cliente o actualiza el que ya tiene con estos datos.

Campo	Tipo	Descripción
nombre	String	Nombre o razón social. Obligatorio si viene <code>rut</code> .
email	String	Correo. Obligatorio si viene <code>rut</code> .
fono	String	Teléfono.
giro	String	Giro. Obligatorio para factura.
direccion, numero, adicional	String	Calle, número y complemento. Obligatorios para factura (salvo <code>adicional</code>).
idComuna, ciudad	Int, String	Comuna (<code>DivisionPolitica</code>) y ciudad.
contacto, emailContacto	String	Persona de contacto, en una empresa.
nombreRetiro	String	Nombre de quien retira (con <code>rutRetiro</code>).

mailing	Integer	1 si acepta recibir publicidad.
----------------	---------	---------------------------------

Despacho (despachoDomicilio)

Sólo cuando la forma de entrega es despacho: **nombre**, **fono**, **direccion**, **numero**, **adicional**, **idComuna**, **ciudad**, **idTransporte** (*Formas*) y **nota** para el repartidor.

Productos (detalle)

Campo	Tipo	Oblig.	Descripción
idProducto	Integer	Sí*	El id_producto de Producto —no el código—. Es lo que descuenta stock. *Una línea sin producto (un cargo, un servicio) va sin él, y sólo con descripcion .
cantidad	Double	Sí	Mayor que cero.
precio	Double	Sí	Precio unitario con IVA , antes del descuento.
descuento	Double	No	Descuento unitario, con IVA. No puede pasar del máximo que la empresa autoriza para ese producto.
descripcion	String	Sí	El texto que se imprime en el documento (hasta 255 caracteres). No se toma del producto: envíelo siempre.
observacion	String	No	Nota de la línea (hasta 50 caracteres).

Pagos ya cobrados (pagos)

Cuando la venta ya viene cobrada —un punto de venta propio, un marketplace que le paga a la empresa, un pago contra entrega que ya se recibió—, el documento puede traer sus pagos y quedar pagado al crearse, sin llamar después a Documento_Voucher. Así se registra cualquier medio, incluidos los que no son electrónicos:

```
"pagos": [
  { "idTipo": 103, "abono": 300000, "codigo": "040217", "idTarjeta": 15, "cuotas": 3 },
  { "idTipo": 100, "abono": 102980 }
]
```

Campo	Tipo	Descripción
idTipo	Integer	El medio de pago, con los códigos de Sphinx: 100 efectivo, 101 cheque, 102 depósito o transferencia, 103 tarjeta, 104 otro, 105 gift card, 106 medio de pago (una pasarela), 107 Webpay, 108 MercadoPago.
abono	Double	Monto pagado con ese medio.
codigo	String	Código de autorización o número de la operación.
idTarjeta, cuotas	Integer	Con tarjeta: cuál (Visa 10, Mastercard 15, Diners 20, American Express 25, Magna 30, CAR 100, Presto 110, Redcompra 200) y en cuántas cuotas.

folio	Long	Número del cheque o del comprobante, si lo tiene.
--------------	------	---

Sólo se aplica cuando lo que se crea es una **boleto o factura**: en un Pedido Web la lista se ignora y el pago se informa con Documento_Voucher. Sphinx no revisa que los pagos sumen el `total`, así que envíelos completos: si lo cubren, la boleto o factura se emite en la misma llamada y la respuesta trae su `pdf`.

Los totales

Sphinx calcula los totales por su cuenta y los compara con los enviados; si no calzan, rechaza el documento. Con t = tasa de IVA (19):

```

neto    = redondeo( ( despacho + Σ cantidad × (precio - descuento) ) / (1 + t/100) ) ± 10 pesos
afecto  = neto - descuento                                                         exacto
iva     = redondeo( afecto × t/100 )                                               ± 5 pesos
total   = afecto + iva                                                             exacto

```

En el ejemplo: $(5.990 + 396.990) / 1,19 = 338.638,66 \rightarrow$ `neto` 338.639; `afecto` 338.639; `iva` 64.341; `total` 402.980. El `descuento` del encabezado es un descuento neto sobre el total; el de cada producto va en su línea.

Ventas duplicadas

Un reintento —porque se cortó la conexión, porque el cliente apretó dos veces— no crea una segunda venta:

- Si ya existe un documento con el mismo `nroOrden` y el mismo `total`, la respuesta es **éxito** con el `id_doc` de ese documento.
- Si existe con el mismo `nroOrden` y otro total, es un error (código 100, ... del Pedido #51259 ya existe en el Sistema): use otro número de orden.
- Además, al crear un pedido Sphinx anula los pedidos pendientes de pago del mismo RUT, de hoy, con los mismos productos y el mismo total: es el cliente que volvió a intentar la compra.

La respuesta

Campo	Tipo	Descripción
id_doc	Long	Identificador del documento en Sphinx. Guárdelo : es el que piden Documento_Voucher y Documento_Anular.
nro_orden	Long	El número de orden enviado.
id_web	String	El <code>idWeb</code> enviado o el que generó Sphinx.
id_pago	String	Sólo con un medio de pago en línea: el identificador con que Sphinx cobra por su cuenta la orden, si la empresa usa el pago de Sphinx.
documento	String	El documento creado: Pedido WEB 51259, o Boleto Electronica Pendiente mientras no tenga folio.
pdf	String	Sólo si en esta llamada se emitió una boleto o factura electrónica: su PDF en base64.
correo	String	El correo del cliente, si lo tiene.

```
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "Documento",
  "resultado": { "id_doc": 5397, "nro_orden": 51259, "id_web": "1b0c6f0e-4f7e-4c50-9a3d-0f2f1d7c2b11",
    "documento": "Pedido WEB 51259", "correo": "maria@correo.cl" } }
```

Errores

Casi todos los rechazos llevan **código 100** y el motivo en `mensaje`, redactado para mostrarlo tal cual:

Código	Mensaje (ejemplos)
1	Falta el cuerpo.
3	No se llamó con POST.
4	El JSON está mal formado.
10	El documento pasó las validaciones pero no se pudo grabar; el motivo va en <code>mensaje</code> (por ejemplo, <code>Fecha esta cerrada contablemente</code>).
15	<code>Sistema de cotizaciones temporalmente desactivado, favor intente mas tarde.</code>
42	<code>No es posible procesar compras para este RUT. Por favor, póngase en contacto con la tienda. (cliente bloqueado)</code>
100	Datos: Sucursal [9] no valida, Bodega [2] no es de la sucursal [1], Debe especificar Forma de Pago, Forma de Entrega desconocida [8], Lista [4] no valida, Tipo de Documento no valido [X], Comuna no existe. Cliente: Falta Nombre del Cliente, Falta email del Cliente, y para factura Falta Giro..., Falta Direccion..., Falta Comuna..., Falta Fono del Cliente. Productos: Producto [17] no existe, Cantidad invalida..., Precio invalido..., Documento sin Productos. Totales: Total diferente a suma de Productos..., Afecto [...] no coincide con Neto - Descuento..., IVA [...] no es el 19% de..., Total [...] no coincide con Afecto + IVA... Negocio: Falta Stock de los productos : NB-001, Descuento excedido de los productos : ..., Promocion excedida de los productos : ..., ... del Pedido #51259 ya existe en el Sistema.

6.2 Documento_Voucher

Informa que el cliente pagó. Sphinx registra el pago en la caja y, si el documento es una boleta o factura, la emite en ese momento.

Ruta	POST /interfazWEB/<base>/Documento_Voucher/<id_doc>[?correo=1]
Cuerpo	El comprobante del medio de pago, en JSON.
correo	Si viene, y se emitió la boleta o factura, se le envía al cliente.

```
{
  "entidad": "WEBPAY",
  "valido": 1,
  "nroOrden": "51259",
  "monto": 402980,
  "codAutorizacion": "040217",
  "nroTransaccion": "11223344",
  "codigoRespuesta": "0",
  "codigoComercio": "597012345678",
  "tipoPago": "VN",
  "numCuotas": 0,
  "ultDigitos": "4514",
  "fechaTransaccion": "2026-09-27",
  "horaTransaccion": "12:34:55"
}
```

Campo	Tipo	Oblig.	Descripción
entidad	String	Sí	Quién cobró, en hasta 15 caracteres: WEBPAY , MERCADOPAGO , FLOW , KHIPU ... Decide cómo queda el pago en la caja de Sphinx: ver Los medios de pago , más abajo.
valido	Integer	Sí	Debe ser 1 : sólo se informan pagos aprobados.
codAutorizacion	String	Sí	Código de autorización del pago (hasta 10 caracteres).
monto	Double	Sí	Monto pagado. Debe coincidir con el total del documento, con 5 pesos de tolerancia.
nroOrden	String	Sí	El mismo nroOrden con que se creó el documento.
tipoPago	String	No	El tipo de pago de Transbank: VD débito; VN crédito sin cuotas; VC , NC , S1 , S2 crédito en cuotas; VP prepago. Un pago con débito cambia la forma de pago del documento a Débito.
tipoTarjeta, abbrevTarjeta	String	No	Tipo (DB , CR) y marca de la tarjeta; si no se envía el tipo, se deduce de tipoPago .
idBanco	Integer	No	Banco, en un pago que no es Transbank.
fechaTransaccion, horaTransaccion	String	No	AAAA-MM-DD y HH:MM:SS .
<i>otros</i>	String	No	nroTransaccion , codigoRespuesta , codigoComercio , terminalId , numCuotas , ultDigitos , numOperacion , numCuenta , rutCliente , tipoTransaccion , fechaContable : se guardan con el pago, tal como los entregó el medio de pago.

Los medios de pago

Documento_Voucher sirve para **cualquier pasarela o medio de pago**, no sólo para los que Sphinx conoce: lo que cambia de uno a otro es la **entidad** . Con ella Sphinx decide con qué tipo de pago lo registra en la caja, que es lo que después ven la cuadratura, el reporte diario y la contabilidad:

entidad	Tipo de pago en Sphinx	Notas
WEBPAY , ONEPAY , TRANSBANK	Webpay	La tarjeta se reconoce por <code>tipoTarjeta</code> o <code>abrevTarjeta</code> (ver abajo). Un pago con débito cambia la forma de pago del documento a Débito.
MERCADOPAGO	MercadoPago	
Cualquier otro nombre: FLOW , KHIPU , GETNET , LINKIFY , PAYPAL , TRANSFERENCIA ...	Medio Pago	El pago queda con el nombre de la entidad en su glosa, y se puede enviar <code>idBanco</code> .

No importa si la entidad va en mayúsculas o minúsculas, pero **use siempre el mismo nombre** para el mismo medio: con él se buscan, se cuadran y se concilian después los pagos de ese medio.

¿A qué cuenta va la plata? Sphinx deja el pago en la cuenta de la empresa del tipo *Medios Pago* cuya glosa sea igual a la `entidad` , en las cuentas bancarias de la empresa. Antes de integrar un medio nuevo, pídale a la empresa que cree esa cuenta con el mismo nombre que usted va a enviar — FLOW , por ejemplo—: así los pagos de ese medio quedan separados de los demás y se pueden conciliar contra lo que el operador deposite. Sin esa cuenta el pago se registra igual, pero sin cuenta de destino.

Las tarjetas. En un pago de Webpay, `tipoTarjeta` o `abrevTarjeta` se buscan entre las tarjetas de Sphinx por su abreviación; si no se envía ninguna, se deduce de `tipoPago` (VD → débito, crédito en el resto):

Abreviación	Tarjeta	Abreviación	Tarjeta
VI	Visa	MAG	Magna
MC	Mastercard	RP	CAR (Ripley)
DC	Diners Club	TP	Presto
AX	American Express	DB	Redcompra (débito)

La empresa puede tener tarjetas propias además de éstas; si su medio de pago informa otra, pídale la abreviación que le corresponde.

Qué hace Sphinx

- Guarda el comprobante y registra el pago en la caja.
- **Boleta o factura:** si con este pago queda pagada, la emite electrónicamente y la respuesta trae su `pdf` .
- **Pedido Web:** lo deja pagado; la boleta o factura la emite la empresa al despacharlo.
- Un segundo pago igual al primero (misma entidad y monto) se descarta: reenviar el voucher no cobra dos veces.

Campo	Tipo	Descripción
documento	String	El documento pagado: Boleta Electronica 245678 , Pedido WEB 51259 .
pdf	String	El PDF de la boleta o factura emitida, en base64.
correo	String	El correo del cliente.

```
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "Documento_Voucher",
  "resultado": { "documento": "Boleta Electronica 245678", "pdf": "JVBERi0xLjQKJ...", "correo":
    "maria@correo.cl" } }
```

Código	Cuándo
12	El documento está anulado (por ejemplo, venció antes de que llegara el pago). El cliente pagó algo que ya no existe: hay que devolverle el dinero o recrear la venta.
20	No se pudo registrar el pago; el motivo va en <code>mensaje</code> (por ejemplo, <code>Nro de Orden [x] no corresponde [y]</code>).
25	El documento no existe o no es una venta que se pueda pagar.
100	<code>Monto del Voucher \$X no coincide con el Documento \$Y</code> , o <code>Voucher Pago no es valido</code> (falta <code>valido = 1</code> o el código de autorización).

6.3 Documento_Anular

Anula un documento que no se pagó —el cliente abandonó el pago, o el medio de pago lo rechazó— y libera el stock que reservaba.

Ruta	POST <code>/interfazWEB/<base>/Documento_Anular</code>
Cuerpo	<code>{ "idDoc": 5397, "obs": "Pago rechazado por WebPay" }</code>

Campo	Tipo	Oblig.	Descripción
idDoc	Long	Sí	El <code>id_doc</code> que devolvió Documento.
obs	String	Sí	El motivo, que queda en el documento.

Contesta código 0 sin `resultado`. Anular dos veces el mismo documento no es un error.

Sólo se puede anular lo que sigue pendiente: un Pedido Web sin pago, o una boleta o factura que todavía no se emite. Lo demás se rechaza con **código 99** y el motivo en `mensaje`: `... no esta pendiente, no lo anulo` (ya emitido), `Pago asociado` (ya tiene un pago registrado) o `Documento asociado` (la empresa ya lo convirtió en otro documento). Esos casos los resuelve la empresa desde Sphinx, con una nota de crédito si corresponde.

7 Consultas para el cliente final

Servicios pensados para que un sitio le muestre algo a su cliente: su boleta, dónde va su despacho, en qué está su equipo en servicio técnico.

7.1 Boleta

El PDF de una boleta electrónica, buscada por los tres datos que el cliente tiene impresos: folio, fecha y monto total.

Ruta **GET** /interfazWEB/<base>/Boleta/<folio>/<AAAA-MM-DD>/<monto>

Campo	Tipo	Descripción
boleta	String	El PDF de la boleta, en formato térmico (el de la impresora de caja), codificado en base64.

Si algún dato no calza con una boleta emitida, contesta código 69 **Boleta no existe**; con un dato mal escrito, 60 (folio), 61 (fecha) o 62 (monto). Cada consulta queda anotada en la boleta.

```
GET /interfazWEB/miempresa/Boleta/48213/2026-09-20/25990
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "Boleta", "resultado": { "boleta":
  "JVBERi0xLjQKJ..." } }
```

7.2 Despacho

El estado del despacho de una factura, boleta o guía, para que el cliente lo siga con su RUT y el folio.

Ruta **GET** /interfazWEB/<base>/Despacho/<rut>/<tipo>/<folio>

tipo **factura**, **boleta** o **guia**, en minúsculas.

Campo	Tipo	Descripción
tipo	String	Nombre del documento (Factura Electronica).
folio	Long	Folio.
rut, rut_dv	Integer, String	RUT del cliente y su dígito verificador.
fecha	String	Fecha del despacho.
transporte	String	Empresa de transporte.
orden	String	Número de seguimiento del transporte.
bultos, peso	Integer, String	Cantidad de bultos y peso.
destino	String	Ciudad de destino.

Errores: 45 RUT inválido, 46 el cliente no existe, 42 cliente bloqueado, 80 tipo inválido, 81 folio inválido, 82 el documento no tiene despacho.

```
GET /interfazWEB/miempresa/Despacho/12608204-5/factura/1783
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "Despacho",
  "resultado": { "transporte": "ChileExpress", "fecha": "2026-09-22 00:00:00", "orden":
    "99812345",
    "bultos": 1, "peso": "2", "destino": "SANTIAGO",
    "tipo": "Factura Electronica", "folio": 1783, "rut": 12608204, "rut_dv": "5" } }
```

7.3 OrdenServicio

Las órdenes de servicio técnico de un cliente de los últimos tres meses.

Ruta **GET** /interfazWEB/<base>/OrdenServicio/<rut>

Campo	Tipo	Descripción
ordenes	Array	Vacío si el cliente no tiene órdenes en el período.
fecha	String	Fecha de la orden.
folio	Long	Número de la orden.
estado	String	Estado en que está (Pendiente , Terminada ...).

Errores: 45 RUT inválido, 46 el cliente no existe, 42 cliente bloqueado.

```
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "OrdenServicio",
  "resultado": { "ordenes": [ { "fecha": "2026-09-10 00:00:00", "estado": "Pendiente", "folio": 2
    } ] } }
```

8 Reportes

Para tableros de gestión y cuadraturas contables. Leen mucho de la base, así que no se ejecutan dos a la vez para la misma empresa (código 8) y conviene llamarlos con moderación: una vez por hora es más que suficiente para un tablero.

8.1 Estadísticas

Las ventas de un período agrupadas por hasta cuatro criterios, como en la pantalla Estadísticas de Sphinx.

Ruta **POST** /interfazWEB/<base>/Estadisticas

Tope Hasta 120 días entre las dos fechas; 100 filas por página.

```
{ "fechaDesde": "2026-09-01", "fechaHasta": "2026-09-27", "idEstadistica": ["DOC", "VEND"],
  "page": 1 }
```

Parámetro	Tipo	Descripción
fechaDesde, fechaHasta	String	Rango de fechas (AAAA-MM-DD), ambas incluidas.
idEstadistica	Array	Hasta cuatro agrupaciones, en orden: DOC documento, DTD tipo de documento, DOC_NUM documento con su número, DBOD bodega, PROVI origen (de dónde proviene la venta), VEND vendedor, D día, H hora. Un valor vacío se ignora; uno desconocido contesta código 99.
page	Integer	Página a leer, desde 1.

Campo	Tipo	Descripción
estadistica	Array	Una fila por combinación de las agrupaciones.
codigol, descripcion1...	String	El valor de cada agrupación, numerados en el orden en que se pidieron.
cantidad	Double	Documentos.
item	Double	Unidades vendidas.
monto	Double	Venta neta (las notas de crédito restan).
paginacion	Object	page , maxPage y count (total de filas).

```
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "Estadisticas",
  "resultado": {
    "paginacion": { "page": 1, "maxPage": 1, "count": 3 },
    "estadistica": [
      { "codigo1": "BVE", "descripcion1": "Boleta Electronica", "codigo2": "DH", "descripcion2":
"Daniel Hasan",
        "cantidad": 35.0, "item": 84.0, "monto": 39206380.0 },
      { "codigo1": "NCE", "descripcion1": "Nota De Credito Electronica", "codigo2": "DH",
"descripcion2": "Daniel Hasan",
        "cantidad": 4.0, "item": 4.0, "monto": -2956700.0 } ] } }
```

8.2 Usuarios

Los usuarios activos de Sphinx, para cruzarlos con el vendedor de [Estadisticas](#) o el cajero de [ReporteDiario](#).

Ruta **GET** /interfazWEB/<base>/Usuarios

Campo	Tipo	Descripción
(resultado)	Array	Directamente la lista, sin un objeto que la envuelva.
id_user	Integer	Identificador.
username	String	Nombre de usuario; es el <code>codigo</code> del vendedor en Estadisticas.
nombre	String	Nombre completo.
rut	String	RUT con dígito verificador (12345678-9), si lo tiene.
id_sucursal	Integer	Sucursal principal.
sucursales	Array	Todas las sucursales a las que tiene acceso.

```
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "Usuarios",
  "resultado": [ { "id_user": 1, "username": "DH", "rut": "12345678-9", "nombre": "Daniel Hasan",
    "id_sucursal": 1, "sucursales": [1, 2] } ] }
```

8.3 Cuadratura

Lo recaudado en un período, sumado por sucursal y por medio de pago, como en la pantalla Cuadratura de Caja.

Ruta **POST** /interfazWEB/<base>/Cuadratura

```
{ "fechaDesde": "2026-09-01", "fechaHasta": "2026-09-30", "idSucursal": null, "idUser": null,
  "cliente": true }
```

Parámetro	Tipo	Descripción
fechaDesde, fechaHasta	String	Rango de fechas. Si se omiten, el día de hoy.

idSucursal	Integer	Una sucursal, o <code>null</code> para todas.
idUser	Integer	Un cajero (<code>id_user</code> de <code>Usuarios</code>), o <code>null</code> para todos.
cliente	Boolean	<code>true</code> (o sin informar): lo cobrado a clientes. <code>false</code> : lo pagado a proveedores.

Campo	Tipo	Descripción
<i>(resultado)</i>	Array	Directamente la lista.
sucursal	String	Sucursal.
tipo	String	Medio de pago (<code>CAJA</code> , <code>CHEQUE</code> , <code>TARJETA MASTERCARD</code> , <code>BANCO</code> <cuenta> ...).
cantidad	String	Cantidad de pagos.
monto	String	Monto total, en pesos.

```
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "Cuadratura",
  "resultado": [ { "sucursal": "Matriz", "tipo": "CAJA", "cantidad": "11", "monto": "2436287" },
                  { "sucursal": "Matriz", "tipo": "TARJETA MASTERCARD", "cantidad": "5", "monto":
                    "8852493" } ] }
```

8.4 ReporteDiario

El reporte diario de caja de una sucursal: cada documento pagado ese día con el detalle de cómo se pagó, como en la pantalla Reporte Diario.

Ruta **POST** `/interfazWEB/<base>/ReporteDiario`

```
{ "idSucursal": 1, "fecha": "2026-09-27", "idUser": null, "cliente": true }
```

Parámetro	Tipo	Descripción
idSucursal	Integer	Sucursal.
fecha	String	El día (AAAA-MM-DD).
idUser	Integer	Un cajero, o <code>null</code> para todos.
cliente	Boolean	Igual que en <code>Cuadratura</code> .

A diferencia del resto, **los campos de esta respuesta van en camelCase**. `resultado` trae cinco listas, con las mismas columnas:

Lista	Qué contiene
<code>pagosDia</code>	Documentos del día pagados el mismo día.
<code>pagosOtroDia</code>	Pagos recibidos ese día de documentos de otros días.
<code>pagosCambio</code>	Cambios y devoluciones.

<code>pagosEgreso</code>	Egresos de caja.
<code>pagosGiftCard</code>	Ventas de tarjetas de regalo.

Columna	Tipo	Descripción
<code>idCtacte</code>	Long	Identificador del movimiento.
<code>documento, td, idTipo, folio</code>	String, String, Int, Long	Documento (FVE 663), su sigla, su tipo y su folio.
<code>fecha</code>	String	Fecha del documento.
<code>rut, nombre</code>	String	Cliente (RUT con dígito).
<code>monto, saldo</code>	Double	Monto del documento y lo que queda por pagar.
<code>sucursal, cajero</code>	String	Dónde y quién lo cobró.
<code>formapago, formaentrega</code>	String	Formas del documento.
<code>efectivo, deposito, tarjeta, webpay, chequeDia, chequeFecha, otro, abonos</code>	Array	Los pagos de ese documento por medio de pago, cada uno con <code>tipo</code> , <code>folio</code> , <code>fecha</code> y <code>monto</code> .

```
{ "codigo": 0, "mensaje": "Exitoso", "servicio": "ReporteDiario",
  "resultado": { "pagosDia": [], "pagosCambio": [], "pagosEgreso": [], "pagosGiftCard": [],
    "pagosOtroDia": [ { "idCtacte": 100666, "idTipo": 70, "documento": "FVE 678", "td": "FVE",
      "folio": 678,
        "fecha": "2026-09-22", "rut": "15678901-1", "nombre": "ANA MORALES
CASTRO",
        "monto": 231693.0, "sucursal": "Matriz", "cajero": "DH",
        "efectivo": [ { "tipo": "", "folio": null, "fecha": null, "monto":
231693.0 } ],
        "deposito": [], "tarjeta": [], "webpay": [], "chequeDia": [],
        "chequeFecha": [],
        "otro": [], "abonos": [], "saldo": null, "descuentoP": null,
        "formapago": null, "formaentrega": null } ] } }
```

9 Procesos recomendados

9.1 Sincronizar el catálogo

Los maestros, en este orden y una vez al día (cambian poco): **Sucursales**, **Listas**, **Formas**, **Marcas**, **DivisionPolitica** y **Familias**. Cada uno devuelve la tabla completa: lo que su sistema tenga y no venga, se borra o se despublica.

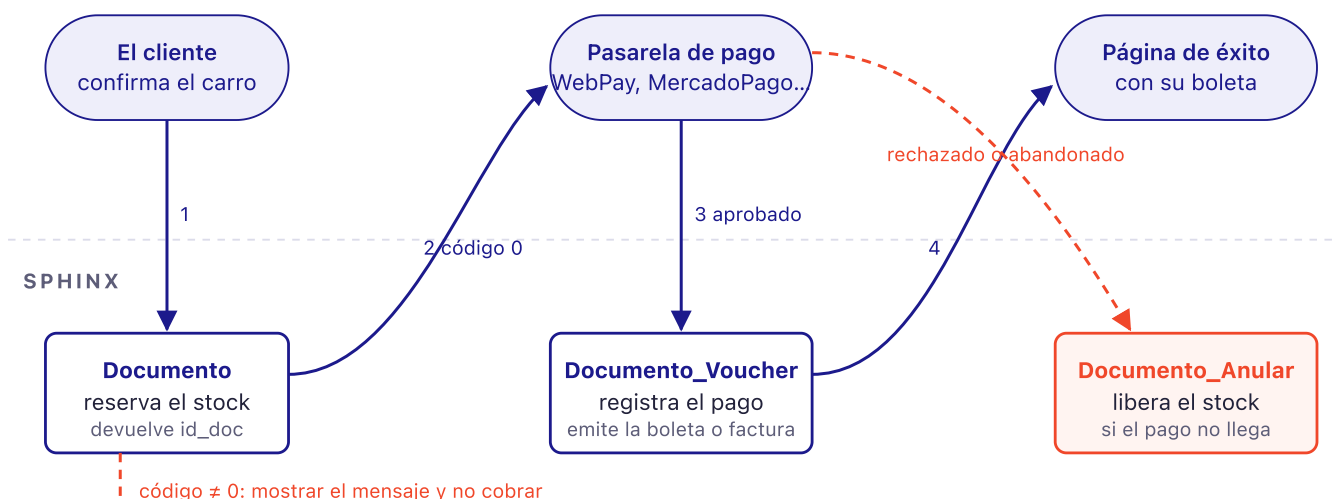
Los productos, en dos pasos:

1. **Una vez al día, la lista completa.** Llame a `Productos` sin fecha. Despublique los que tenga y no vengan, y marque para actualizar los que no tenga o cuya `fecha_modificacion` sea distinta de la que guardó.
2. **Cada 5 a 15 minutos, los cambios.** Llame a `Productos/<fecha>` con la hora de su última consulta menos un par de minutos de margen. Marque para actualizar los códigos que vengan.
3. **Traiga el detalle de los marcados** con `Producto`, en lotes de hasta 100 códigos. Si `faltaron` es mayor que cero, vuelva a pedir los que no llegaron. Guarde la `fecha_modificacion` de cada uno.
4. **Muestre** sólo los que tengan `lista = 1`; los demás se ocultan pero se conservan, porque pueden volver.

Pesa menos con las fotos por URL. En base64 cada llamada a `Producto` trae todas las fotos de nuevo, aunque no hayan cambiado. Si su integración puede bajar imágenes de una dirección, pida a la empresa que publique sus fotos por URL (5.4).

9.2 Registrar una venta

SU TIENDA



1. **Al confirmar el carro**, llame a `Documento`. Si contesta un código distinto de 0, muéstrole el mensaje al cliente (falta de stock, un dato que falta) y no lo mande a pagar. Si contesta 0, guarde el `id_doc`: el stock ya está reservado.
2. **Mande al cliente a pagar** por el monto `total` que envió.

3. **Si el pago se aprueba**, informe el comprobante con `Documento_Voucher` y el `id_doc`. Si la respuesta trae `pdf`, puede mostrárselo al cliente o enviarlo con `?correo=1`.
4. **Si el pago se rechaza o el cliente abandona**, llame a `Documento_Anular` para liberar el stock. Si no lo hace, Sphinx lo anula igual cuando vence la vigencia del pedido, pero ese stock queda bloqueado mientras tanto.

Reintente sin miedo. Si una llamada a Documento no contesta (se cortó la conexión, pasaron los 300 segundos), repítala con el mismo `nroOrden` y el mismo total: si el documento alcanzó a crearse, Sphinx devuelve el mismo `id_doc` en vez de crear otro. Lo mismo con `Documento_Voucher`: un pago repetido no se cobra dos veces.

9.3 Antes de salir a producción

- `Check` contesta código 0 en la base de producción.
- Las formas de pago y de entrega que usa su sistema están en `Formas`, y guardó sus códigos.
- Una venta de prueba completa —Documento, Documento_Voucher— dejó el documento pagado en Sphinx, y otra con Documento_Anular devolvió el stock. Coordine la prueba con la empresa, porque quedan registradas en su sistema.
- Su programa mira siempre el `codigo` de la respuesta y no sólo el HTTP 200.
- La clave está sólo en su servidor.

Las dudas sobre la integración se atienden en soporte@sphinx.cl. Incluya el servicio, la hora de la llamada y el cuerpo que envió: con eso se encuentra en el registro del servidor.