

BOTYCONNECT DEVELOPERS

BC

API DE CARTERA

v1.1

Guía de integración para ERP y sistemas de
crédito

CONTRATO DE PRODUCCIÓN

<https://api.botyconnect.com/api/v1/botyconnect>

OpenAPI 3.1: developers.botyconnect.com/openapi.yaml

Versión 1.1 | 17 de agosto de 2026

soporte@botyconnect.com



Contenido

Esta guía contiene el contrato operativo que debe implementar el ERP para sincronizar cartera y conciliar pagos con Botyconnect.

- 1. Objetivo y modelo de integración
- 2. Autenticación y permisos
- 3. Idempotencia obligatoria
- 4. Formato de respuestas
- 5. Estados y reglas generales
- 6. Crear o actualizar una deuda
- 7. Sincronizar deudas por lote
- 8. Consultar una deuda
- 9. Registrar un pago recibido fuera de Boty
- 10. Resolver una deuda
- 11. Webhook de pagos originados en Boty
- 12. Errores relevantes
- 13. Estrategia de sincronización
- 14. Checklist de salida a producción
- 15. Soporte y trazabilidad

1. Objetivo y modelo de integración

La API permite que el ERP del cliente sincronice deudas, cuotas y pagos con Botyconnect para ejecutar cobranza por WhatsApp.

- El ERP del cliente continúa siendo la fuente de verdad de la cartera.
- Botyconnect mantiene una réplica operativa para cobranza y conciliación.
- El ERP asigna un `external_id` estable a cada deuda, cuota y pago.
- Botyconnect informa por webhook los pagos originados dentro de Boty.
- La API nunca expone IDs internos de base de datos.

Flujo recomendado:

1. Obtener una API key con scopes `cartera:read` y `cartera:write`.
2. Consultar el `account_id` del número de WhatsApp que realizará la cobranza.
3. Ejecutar la carga inicial mediante `POST /cartera/deudas/lote`.
4. Enviar cambios incrementales mediante `POST /cartera/deudas`.
5. Reflejar pagos recibidos por el ERP mediante `POST /cartera/pagos`.
6. Configurar el webhook `debt.payment_received` para pagos recibidos en Boty.
7. Conciliar una muestra periódica con `GET /cartera/deudas/{external_id}`.

2. Autenticación y permisos

Todas las solicitudes usan una API key de Botyconnect como Bearer token:

HTTP

```
Authorization: Bearer bc_live_bck_XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
Accept: application/json
```

La API key pertenece a una empresa o tenant. Una key no puede consultar deudas ni usar números de WhatsApp de otra empresa.

Scope	Permite
<code>cartera:read</code>	Consultar deudas, cuotas y pagos sincronizados.
<code>cartera:write</code>	Sincronizar deudas, registrar pagos externos y resolver cartera.
<code>accounts:read</code>	Consultar los números disponibles y obtener su <code>account_id</code> .
<code>webhooks:write</code>	Crear y administrar el endpoint que recibe pagos originados en Boty.

Formato real de la key: `bc_live_{public_id}.{secret}`. El secreto completo solo se muestra al crear o rotar la key y debe guardarse en un gestor de secretos.

3. Idempotencia obligatoria

Toda operación POST de cartera exige:

HTTP

Idempotency-Key: 39f350eb-f2a2-4dad-9574-91f0bd6df7bd

La key identifica una operación lógica, no una conexión HTTP.

- Mismo Idempotency-Key y mismo body: devuelve la respuesta original sin repetir la operación.
- Mismo Idempotency-Key con body distinto: 409 idempotency_key_conflict.
- Header ausente: 422 idempotency_key_required.
- Las keys se conservan durante 48 horas.
- Si ocurre un timeout, se debe reintentar con la misma key y el mismo body.

Además de la idempotencia HTTP, la deuda y el pago tienen identidad funcional:

- Deuda: combinación tenant + external_id.
- Pago: combinación tenant + external_id.
- Rotar la API key no cambia estas identidades ni crea duplicados.

4. Formato de respuestas

Respuesta exitosa:

JSON

```
{
  "success": true,
  "data": {}
}
```

Respuesta de error:

JSON

```
{
  "success": false,
  "error": {
    "message": "Descripción legible del error.",
    "code": "codigo_estable",
    "details": {}
  }
}
```

Cada respuesta incluye X-Botyconnect-Request-Id. Debe enviarse ese valor al soporte de Botyconnect cuando se reporte una incidencia.

5. Estados y reglas generales

Los nombres de estados se envían en inglés y en minúscula.

Estado de deuda	Significado
pending	Pendiente, sin pagos aplicados.

partial	Tiene uno o más abonos y saldo pendiente.
paid	Saldada; el saldo debe ser cero.
overdue	Vencida y aún cobrable.
cancelled	Anulada.
written_off	Castigada o dada de baja como incobrable.

Estado de cuota	Significado
pending	Pendiente.
overdue	Vencida.
paid	Pagada; Botyconnect no permite sobrescribirla después.
cancelled	Cancelada.

Reglas de identificadores externos:

- Longitud máxima: 160 caracteres.
- Patrón permitido: letras, números, punto, guion, guion bajo y dos puntos.
- Ejemplos válidos: CRED-000123, erp:credito:981, PAGO_2026.99881.
- El valor debe ser estable e inmutable durante toda la vida del registro.

6. Crear o actualizar una deuda

HTTP

```
POST /cartera/deudas
Scope: cartera:write
Idempotency-Key: debt-CRED-000123-v7
```

Ejemplo cURL:

BASH

```
curl -X POST 'https://api.botyconnect.com/api/v1/botyconnect/cartera/deudas' \
-H 'Authorization: Bearer bc_live_...' \
-H 'Idempotency-Key: debt-CRED-000123-v7' \
-H 'Content-Type: application/json' \
--data @deuda.json
```

Archivo deuda.json:

JSON

```
{
  "account_id": "wacc_x8k2m9p4q7w1n5r3t6y0u2i4",
  "external_id": "CRED-000123",
  "external_version": 7,
  "external_updated_at": "2026-08-17T14:20:00-05:00",
  "deudor": {
    "nombre": "Johani Ramirez",
    "documento": "1033694167",
    "telefono": "573235529764"
  },
}
```

JSON

```

"moneda": "COP",
"referencia": "Credito mueble sala",
"descripcion": "Compra a cuotas - sede Norte",
"monto_original": 1360000,
"saldo_pendiente": 870000,
"tipo": "installments",
"status": "partial",
"auto_send": true,
"cuotas": [
  {
    "external_id": "CU0-0007",
    "numero": 7,
    "monto": 80000,
    "vence": "2026-09-15",
    "estado": "pending"
  },
  {
    "external_id": "CU0-0008",
    "numero": 8,
    "monto": 80000,
    "vence": "2026-10-15",
    "estado": "pending"
  }
]
}

```

6.1 Campos de deuda

Campo	Tipo	Req.	Descripción
account_id	string	Sí	ID público wacc_ del número de WhatsApp que gestionará la cobranza.
external_id	string	Sí	ID estable de la deuda en el ERP.
external_version	integer	No	Versión creciente. Se recomienda enviarla siempre.
external_updated_at	date-time	No	Control de orden alternativo cuando no se usa versión.
deudor.nombre	string	Sí	Nombre para identificación y mensajes.
deudor.documento	string	No	Documento del cliente, único dentro de la empresa.
deudor.telefono	string	Sí	Número internacional; se normalizan espacios y símbolos.
moneda	string	Sí	Código ISO de tres letras, por ejemplo COP.
monto_original	number	Sí	Valor original, mayor que cero.
saldo_pendiente	number	Sí	Saldo actual, entre cero y monto_original.
tipo	string	No	single o installments.
status	string	No	Si se omite, Botyconnect lo infiere desde el saldo.
referencia	string	No	Factura, contrato o referencia visible.
descripcion	string	No	Información adicional, máximo 5.000 caracteres.
auto_send	boolean	No	Habilita la operación automática de cobranza configurada en Boty.
cuotas	array	No	Snapshot del cronograma; máximo 500 cuotas.

6.2 Semántica del snapshot de cuotas

La presencia del campo cuotas es significativa:

- Campo ausente: conserva el cronograma existente y solo actualiza la deuda.
- "cuotas": []: cancela cuotas abiertas y conserva las pagadas.
- Arreglo con elementos: actualiza o crea las cuotas recibidas y cancela cuotas abiertas que ya no estén en el snapshot.
- Una cuota pagada nunca se sobrescribe ni se elimina.
- external_id y numero deben ser únicos dentro de cada deuda.

6.3 Control de eventos atrasados

Se recomienda enviar external_version como un entero creciente por deuda. Si Botyconnect ya aplicó la versión 7 y recibe nuevamente la versión 7 o una versión menor, responde action: ignored_stale y no modifica la deuda.

Cuando no se informa versión, puede usarse external_updated_at; un timestamp igual o anterior también se ignora.

Respuesta al crear, 201:

JSON

```
{
  "success": true,
  "data": {
    "action": "created",
    "deuda": {
      "external_id": "CRED-000123",
      "account_id": "wacc_x8k2m9p4q7w1n5r3t6y0u2i4",
      "deudor": {
        "nombre": "Johani Ramirez",
        "documento": "1033694167",
        "telefono": "573235529764"
      },
      "tipo": "installments",
      "moneda": "COP",
      "monto_original": 1360000,
      "saldo_pendiente": 870000,
      "status": "partial",
      "external_version": 7,
      "cuotas": [],
      "pagos": []
    }
  }
}
```

En una actualización responde 200 con action: updated. Para un evento obsoleto responde 200 con action: ignored_stale.

7. Sincronizar deudas por lote

HTTP

POST /cartera/deudas/lote
Scope: cartera:write

El endpoint acepta entre 1 y 500 deudas:

JSON

```
{
  "deudas": [
    {
      "account_id": "wacc_x8k2m9p4q7w1n5r3t6y0u2i4",
      "external_id": "CRED-000123",
      "external_version": 7,
      "deudor": {
        "nombre": "Johani Ramirez",
        "documento": "1033694167",
        "telefono": "573235529764"
      },
      "moneda": "COP",
      "monto_original": 1360000,
      "saldo_pendiente": 870000
    }
  ]
}
```

Respuesta 200:

JSON

```
{
  "success": true,
  "data": {
    "summary": {
      "created": 1,
      "updated": 0,
      "ignored_stale": 0,
      "errors": 0
    },
    "results": [
      {
        "external_id": "CRED-000123",
        "success": true,
        "action": "created",
        "deuda": {}
      }
    ]
  }
}
```

Un error de negocio de un elemento no revierte los elementos válidos. Un error estructural del body, como más de 500 deudas o IDs duplicados dentro del lote, rechaza todo el request con 422.

Para reintentar únicamente los errores, se deben enviar en un lote nuevo y utilizar otro Idempotency-Key.

8. Consultar una deuda

HTTP

```
GET /cartera/deudas/{external_id}
Scope: cartera:read
```

BASH

```
curl 'https://api.botyconnect.com/api/v1/botyconnect/cartera/deudas/CRED-000123' \
-H 'Authorization: Bearer bc_live_...' \
-H 'Accept: application/json'
```

La respuesta contiene la deuda, el deudor, todas las cuotas y los pagos registrados. Es el endpoint recomendado para conciliación puntual.

Si no existe en el tenant autenticado responde 404 debt_not_found.

9. Registrar un pago recibido fuera de Boty

Este endpoint se usa cuando el cliente recibe dinero por caja, transferencia o una pasarela propia.

HTTP

```
POST /cartera/pagos
Scope: carterita:write
Idempotency-Key: payment-PAGO-99881
```

JSON

```
{
  "external_id": "PAGO-99881",
  "deuda_external_id": "CRED-000123",
  "monto": 120000,
  "moneda": "COP",
  "metodo": "transferencia",
  "pagado_en": "2026-08-17T14:30:00-05:00",
  "metadata": {
    "caja": "Sede Norte"
  }
}
```

Métodos aceptados: efectivo, transferencia, pasarela, otro, manual y transfer.

Reglas:

- Un pago se aplica una sola vez por su external_id.
- Repetir el mismo ID, deuda y monto responde action: duplicate sin aplicar dinero de nuevo.
- Reutilizar el ID con datos distintos responde 409 payment_external_id_conflict.
- La moneda debe coincidir con la deuda.
- No se aceptan pagos nuevos sobre deudas paid, cancelled o written_off.
- El pago se asigna por FIFO: primero a la cuota abierta con vencimiento más antiguo.
- Un pago puede completar una cuota y abonar la siguiente.
- Un sobrepago se informa en monto_sin_asignar.
- Este pago no genera un webhook de vuelta al ERP, evitando un eco circular.

Respuesta 201:

JSON

```
{
  "success": true,
  "data": {
    "action": "created",
    "pago": {
```

JSON

```
{
  "external_id": "PAGO-99881",
  "deuda_external_id": "CRED-000123",
  "monto": 120000,
  "moneda": "COP",
  "metodo": "transferencia",
  "pagado_en": "2026-08-17T19:30:00.000000Z",
  "asignaciones": [
    {
      "cuota_external_id": "CU0-0007",
      "numero": 7,
      "monto": 80000
    },
    {
      "cuota_external_id": "CU0-0008",
      "numero": 8,
      "monto": 40000
    }
  ]
},
{
  "saldo_pendiente": 750000,
  "monto_sin_asignar": 0
}
}
```

10. Resolver una deuda

HTTP

```
POST /cartera/deudas/{external_id}/resolver
Scope: cartera:write
```

JSON

```
{
  "status": "written_off",
  "reason": "Castigo aprobado por el ERP"
}
```

Estados permitidos:

- paid: fija el saldo en cero.
- cancelled: anula la deuda y conserva el saldo como dato histórico.
- written_off: marca la cartera como castigada y conserva el saldo histórico.

En los tres casos Botyconnect cancela cuotas abiertas y recordatorios de cobranza pendientes.

11. Webhook de pagos originados en Boty

Evento disponible: debt.payment_received.

Se emite cuando una deuda sincronizada recibe:

- Un pago aprobado por Wompi dentro de Boty.
- Un pago manual registrado por un gestor dentro de Boty.

No se emite para pagos recibidos mediante POST /cartera/pagos.

Headers:

HTTP

```
X-Botyconnect-Event-Id: evt_x8k2m9p4q7w1n5r3t6y0u2i4
X-Botyconnect-Event: debt.payment_received
X-Botyconnect-Timestamp: 1786971731
X-Botyconnect-Signature: sha256=<hmac>
```

Payload:

JSON

```
{
  "id": "evt_x8k2m9p4q7w1n5r3t6y0u2i4",
  "event": "debt.payment_received",
  "created_at": "2026-08-17T19:02:11.000000Z",
  "data": {
    "deuda_external_id": "CRED-000123",
    "pago": {
      "boty_payment_id": "bpay_981",
      "external_id": null,
      "transaction_id": "wompi-transaction-123",
      "monto": 80000,
      "moneda": "COP",
      "metodo": "wompi",
      "pagado_en": "2026-08-17T19:02:10.000000Z"
    },
    "saldo_pendiente": 790000,
    "status": "partial"
  }
}
```

11.1 Validación de firma

Botyconnect calcula:

TEXT

```
sha256=HMAC_SHA256(signing_secret, timestamp + "." + raw_body)
```

Ejemplo PHP:

PHP

```
$timestamp = $_SERVER['HTTP_X_BOTYCONNECT_TIMESTAMP'] ?? '';
$received = $_SERVER['HTTP_X_BOTYCONNECT_SIGNATURE'] ?? '';
$rawBody = file_get_contents('php://input');
$expected = 'sha256=' . hash_hmac(
    'sha256',
    $timestamp . '.' . $rawBody,
    $signingSecret
);

if (!hash_equals($expected, $received)) {
    http_response_code(401);
    exit;
}
```

Recomendaciones:

- Calcular la firma sobre el body crudo antes de decodificar JSON.
- Comparar en tiempo constante.
- Rechazar timestamps con más de cinco minutos de diferencia.

- Deduplicar por X-Botyconnect-Event-Id.
- Guardar el evento antes de ejecutar procesos lentos y responder 2xx rápidamente.
- Botyconnect realiza hasta cinco intentos con backoff: 10, 60, 300, 900 y 1.800 segundos.

12. Errores relevantes

HTTP	Código	Acción recomendada
401	unauthorized	Revisar Bearer token, estado de la key y formato bc_live_.
403	forbidden_scope	Crear o rotar la key con el scope requerido.
404	account_not_found	Verificar que el account_id pertenezca al tenant.
404	debt_not_found	Verificar el external_id de la deuda.
409	idempotency_key_conflict	Usar una key nueva o repetir exactamente el body original.
409	debtor_identity_conflict	Documento y teléfono apuntan a contactos distintos; corregir el ERP.
409	debtor_phone_conflict	El teléfono ya pertenece a otro contacto de la cuenta.
409	payment_external_id_conflict	No reutilizar un ID de pago con otro monto o deuda.
409	payment_currency_mismatch	Enviar la misma moneda de la deuda.
409	debt_not_payable	No aplicar pagos nuevos a una deuda ya resuelta.
422	idempotency_key_required	Agregar el header obligatorio.
422	validation_failed	Corregir los campos informados en error.details.
429	límite excedido	Reintentar con backoff y respetar Retry-After cuando venga.

Límites por defecto:

- API key: 120 requests por minuto, configurable.
- Tenant: 1.000 requests por minuto.
- Lecturas: 600 requests por minuto por endpoint.
- Escrituras: 120 requests por minuto por endpoint.

13. Estrategia de sincronización recomendada

Carga inicial:

1. Dividir la cartera en lotes de máximo 500 deudas.
2. Usar un Idempotency-Key distinto para cada lote.
3. Guardar summary y results para auditoría.
4. Reintentar únicamente los errores en un lote nuevo.

Operación incremental:

1. Aumentar external_version cuando cambie saldo, estado, deudor o cronograma.

2. Enviar el upsert después de confirmar la transacción del ERP.
3. Persistir el request y su Idempotency-Key hasta recibir respuesta.
4. Reintentar timeouts y errores 5xx con la misma key y body.
5. No reintentar automáticamente errores 409 o 422 sin corregir la causa.

Conciliación:

1. Procesar webhooks idempotentemente.
2. Comparar periódicamente una muestra de saldos mediante GET.
3. Si hay una diferencia, el ERP debe enviar una versión nueva de la deuda.
4. Botyconnect nunca modifica una cuota pagada desde un snapshot posterior.

14. Checklist de salida a producción

- ☐ API key almacenada en un gestor de secretos.
- ☐ Key con cartera:read, cartera:write, accounts:read y permisos de webhook necesarios.
- ☐ account_id de producción confirmado.
- ☐ external_id estable definido para deudas, cuotas y pagos.
- ☐ external_version creciente implementado.
- ☐ Idempotency-Key persistido para todos los POST.
- ☐ Carga de prueba ejecutada con un lote pequeño.
- ☐ Pagos parciales y pagos que cubren varias cuotas probados.
- ☐ Resolución paid, cancelled y written_off probada.
- ☐ Webhook HTTPS configurado con debt.payment_received.
- ☐ Firma HMAC y deduplicación por Event-Id verificadas.
- ☐ Reintentos con backoff implementados para timeout, 429 y 5xx.
- ☐ Alertas creadas para errores persistentes y diferencias de conciliación.

15. Soporte y trazabilidad

Al reportar un incidente, incluir:

- Fecha y hora con zona horaria.
- Endpoint y método HTTP.
- external_id involucrado.
- Idempotency-Key sin incluir la API key.
- X-Botyconnect-Request-Id de la respuesta.
- Código HTTP y error.code.

Contacto: soporte@botyconnect.com.