



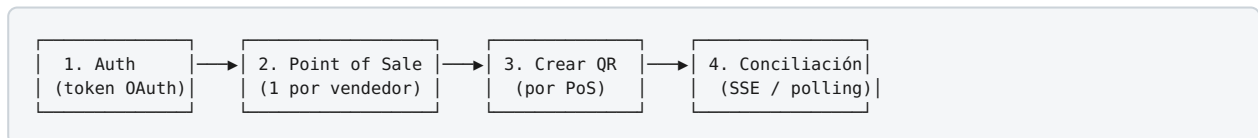
Documentación Técnica — Flujo de Cobro con QR (API v4)

Versión: 1.1 | Fecha: Julio 2026 | Audiencia: Equipos técnicos que integran el flujo de cobro por QR sobre la plataforma Sipago.

1. Descripción general

Este documento describe el flujo completo de integración para generar códigos QR de cobro (dinámicos y estáticos), escuchar la confirmación del pago en tiempo real y conciliar las operaciones.

El flujo consta de cuatro etapas:



- 1. Autenticación:** obtener un `access_token` OAuth 2.0 con las credenciales provistas por Sipago.
- 2. Points of Sale:** crear los puntos de venta necesarios (por ejemplo, uno por vendedor).
- 3. Generación de QR:** crear QRs dinámicos (con monto) o estáticos (sin monto) asociados a un point of sale. Se obtiene un `qr_id`.
- 4. Conciliación:** con el `qr_id` se consulta el estado del pago, ya sea por suscripción SSE (tiempo real) o por consulta directa (polling), y se obtiene el detalle de la autorización.

2. Prerequisitos

2.1. Credenciales

Antes de comenzar, el integrador debe solicitar credenciales al equipo de Sipago:

Credencial	Descripción
<code>client_id</code>	Identificador del cliente OAuth. Es particular de cada cliente/integrador y debe solicitarse al equipo de Sipago.
<code>client_secret</code>	Secreto del cliente OAuth. Es particular de cada cliente/integrador y debe solicitarse al equipo de Sipago.
<code>TENANT_IDENTIFIER</code>	Identificador del tenant (UUID del realm), necesario para el auth y la suscripción SSE. Valor fijo: <code>5662e892-59f2-4a54-aea8-cfbf77942843</code>

Importante — TENANT_IDENTIFIER: el valor `5662e892-59f2-4a54-aea8-cfbf77942843` es un identificador **único y compartido**: lo utilizan **todos los clientes/integradores** por igual, tanto en el path del endpoint de autenticación (sección 3) como en el tópico de la suscripción SSE (sección 6.1). No es necesario solicitarlo por cliente ni varía entre integraciones — a diferencia de `client_id` y `client_secret`, que sí son específicos de cada cliente.

⚠ Las credenciales son sensibles. Deben almacenarse de forma segura (vault / secret manager) y nunca versionarse en repositorios ni exponerse en clientes front-end.

2.2. Información del comercio

Vas a necesitar información de tu comercio:

- `SUBSIDIARY_ID`
- `MERCHANT_ID`

Esta información podés obtenerla logueándote en <https://portal.sipago.coop/>. En tu perfil vas a encontrar la información necesaria.



Información del Comercio



Comercio: **Martin Pereira1**

#

ID Sucursal: **12345**

#

ID Merchant: **67890**

2.3. Ambientes y hosts

Servicio	Host
Identity Provider (Keycloak)	https://idp.prd.geopagos.io
API principal	https://api.prd.geopagos.io
API mPOS (generación de QR)	https://api-mpos-cabal.prd.geopagos.io

3. Etapa 1 — Autenticación

Se utiliza el flujo OAuth 2.0 `client_credentials` contra el Identity Provider.

Request

POST https://idp.prd.geopagos.io/auth/realms/{TENANT_IDENTIFIER}/protocol/openid-connect/token
Content-Type: `application/x-www-form-urlencoded`

```
curl --location 'https://idp.prd.geopagos.io/auth/realms/{TENANT_IDENTIFIER}/protocol/openid-connect/token' \
--header 'Content-Type: application/x-www-form-urlencoded' \
--data-urlencode 'client_id={CLIENT_ID}' \
--data-urlencode 'client_secret={CLIENT_SECRET}' \
--data-urlencode 'grant_type=client_credentials' \
--data-urlencode 'scope=openid'
```

Response — 200 OK

```
{
  "access_token": "eyJhbGciOiJSUzI1NiIs...\"",
  "expires_in": 900,
  "refresh_expires_in": 0,
  "token_type": "Bearer",
  "id_token": "eyJhbGciOiJSUzI1NiIs...\"",
  "not-before-policy": 0,
  "scope": "openid platform merchant",
  "tid": "{TENANT_IDENTIFIER}"
}
```

Consideraciones

- El token expira a los 900 segundos (15 minutos). No hay refresh token (`refresh_expires_in: 0`): al expirar, se debe solicitar un nuevo token repitiendo este request.
- Se recomienda cachear el token y renovarlo proactivamente antes de su expiración (por ejemplo, a los ~13 minutos) para evitar errores 401 en llamadas concurrentes.
- El token se envía en las siguientes llamadas mediante el header `Authorization: Bearer {access_token}`.

4. Etapa 2 — Points of Sale

Un point of sale (PoS) representa un punto de cobro. La recomendación es crear uno por vendedor (o por caja/dispositivo, según el modelo de negocio). Los QRs siempre se generan asociados a un point of sale.

Para crear un PoS se requiere conocer previamente el `merchant_id` y el `subsidiary_id` (provistos junto con el alta del comercio).

4.1. Crear point of sale

El `SUBSIDIARY_ID` y el `MERCHANT_ID` utilizados en el body de este request se obtienen desde el portal de Sipago, según lo indicado en la sección 2.2 (Información del comercio).

POST `https://api.geopagos.io/api/v4/points-of-sale`

```
curl --location 'https://api.geopagos.io/api/v4/points-of-sale' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'Authorization: Bearer {access_token}' \
--data '{
  "name": "point 3",
  "description": "punto de venta point 3",
  "reference": "subsidiary.{SUBSIDIARY_ID}",
  "merchant_id": "{MERCHANT_ID}"
}'
```

Response — 200 OK

```
{
  "uuid": "1ddddecf-3bfa-4dab-96db-607afe061a5c",
  "name": "point 3",
  "description": "punto de venta point 3",
  "reference": "subsidiary.1739"
}
```

El campo `uuid` es el identificador del point of sale (`point_of_sale_id`). Debe persistirse, ya que se utiliza para generar QRs y filtrar movimientos.

4.2. Actualizar point of sale

PUT `https://api.geopagos.io/api/v4/points-of-sale/{point_of_sale_uuid}`

```
curl --location --request PUT 'https://api.geopagos.io/api/v4/points-of-sale/{point_of_sale_uuid}' \
--header 'Content-Type: application/json' \
--header 'Accept: application/json' \
--header 'Authorization: Bearer {access_token}' \
--data '{
  "name": "point 3 updated"
}'
```

Response — 200 OK

```
{
  "uuid": "1ddddecf-3bfa-4dab-96db-607afe061a5c",
  "name": "point 3 updated",
  "description": "",
  "reference": ""
}
```

⚠ El PUT es de reemplazo: los campos no enviados vuelven vacíos (ver `description` y `reference` en el ejemplo). Enviar siempre el objeto completo si se desea conservar los valores existentes.

4.3. Eliminar point of sale

```
DELETE https://api.geopagos.io/api/v4/points-of-sale/{point_of_sale_uuid}
```

Response — 204 No Content

4.4. Listar points of sale

```
GET https://api.prd.geopagos.io/api/v4/points-of-sale/
```

```
curl --location 'https://api.prd.geopagos.io/api/v4/points-of-sale/' \
--header 'Authorization: Bearer {access_token}' \
--header 'Accept: application/json'
```

Response — 200 OK

```
[
  {
    "uuid": "6fda9b64-d317-4723-9480-cda88579c987",
    "name": "Federico kucher",
    "description": "Point of sale for user kucherfederico@gmail.com",
    "reference": "subsidiary.1739"
  }
]
```

5. Etapa 3 — Generación de QR

Los QRs se generan por point of sale, indicando el `uuid` del PoS en el path. Existen dos tipos:

Tipo	Uso	Monto	Vigencia
Dinámico	Una venta puntual (se incluyen ítems y monto)	Fijo, definido al crearlo	Expira (~1 hora)
Estático	QR impreso/permanente del punto de venta	Lo ingresa el pagador	No expira

5.1. Crear QR dinámico

```
POST https://api-mpos-cabal.prd.geopagos.io/api/v4/payments/qr/{point_of_sale_uuid}
```

```
curl --location 'https://api-mpos-cabal.prd.geopagos.io/api/v4/payments/qr/{point_of_sale_uuid}' \
--header 'Authorization: Bearer {access_token}' \
--header 'Content-Type: application/json' \
--data '{
  "source": "mpos_app",
  "type": "GENERIC",
  "items": [
    {
      "name": "Venta",
      "amount": 111.0,
      "quantity": 1
    }
  ],
  "currency": "032"
}'
```

Campo	Descripción
<code>source</code>	Origen de la operación (ej.: <code>mpos_app</code>)
<code>type</code>	Tipo de QR (<code>GENERIC</code>)
<code>items[]</code>	Ítems de la venta: <code>name</code> , <code>amount</code> , <code>quantity</code>
<code>currency</code>	Código ISO 4217 numérico (032 = Peso argentino)

Response — 200 OK

```
{
  "id": "8cf1ce0f-8969-4029-b8ba-87406971d54d",
  "hash": "8cf1ce0f-8969-4029-b8ba-87406971d54d",
  "qr": "00020101021230150011coop.sipago...",
  "expires_at": "2026-06-16T15:41:19.1549225Z",
  "expires_in_seconds": 3599
}
```

Campo	Descripción
id / hash	qr_id : identificador del QR / intent de pago. Se usa para consultar estado y conciliar. Debe persistirse asociado a la venta.
qr	Payload EMVCo del QR. Es el contenido a renderizar como imagen QR.
expires_at / expires_in_seconds	Vencimiento del QR dinámico (~3600 segundos)

El string `qr` se renderiza con cualquier librería estándar de generación de códigos QR (formato EMVCo / QR interoperable).

5.2. Obtener QR estático

Mismo endpoint, sin `items` (no lleva monto):

```
curl --location 'https://api-mpos-cabal.prd.geopagos.io/api/v4/payments/qr/{point_of_sale_uuid}' \
--header 'Authorization: Bearer {access_token}' \
--header 'Content-Type: application/json' \
--data '{
  "source": "mpos_app",
  "type": "GENERIC",
  "currency": "032"
}'
```

Response — 200 OK

```
{
  "id": "7c25437f-5543-4dde-beff-f606ead88cb0",
  "hash": "7c25437f-5543-4dde-beff-f606ead88cb0",
  "qr": "00020101021130150011coop.sipago..."
}
```

⚠ La consulta de estado por intent (sección 6.2) y el detalle de venta por intent (sección 6.3) aplican solo a QR dinámico. Para QR estático, la conciliación se realiza a través de la lista de movimientos filtrada por point of sale (sección 6.5) o de las notificaciones SSE.

6. Etapa 4 — Confirmación de pago y conciliación

Una vez generado el QR y mostrado al pagador, existen dos estrategias para conocer el resultado:

- **Tiempo real (recomendada):** suscripción SSE al tópico del QR.
- **Polling:** consulta periódica del estado del intent.

6.1. Suscripción SSE (tiempo real)

```
GET https://api.prd.geopagos.io/sse/notifications?topic=urn%3A{TENANT_IDENTIFIER}%3Aintent_{QrHash}
Accept: text/event-stream
```

```
curl --location 'https://api.prd.geopagos.io/sse/notifications?topic=urn%3A{TENANT_IDENTIFIER}%3Aintent_{QrHash}' \
--header 'Authorization: Bearer {access_token}' \
--header 'Accept: text/event-stream'
```

Parámetro	Descripción
TENANT_IDENTIFIER	UUID del tenant, valor compartido por todos los clientes: 5662e892-59f2-4a54-aea8-cfbf77942843 (ver sección 2.1)
QrHash	id / hash del QR generado

El tópicó tiene la forma `urn:{TENANT_IDENTIFIER}:intent_{QrHash}` (URL-encoded: `urn%3A...%3Aintent_...`).

Cuando el QR se paga, la notificación SSE incluye el `reference_number` de la autorización, que permite obtener el detalle completo de la venta (sección 6.4).

Consideraciones de implementación: mantener la conexión abierta mientras el QR esté vigente, implementar reconexión ante cortes y contemplar la expiración del token (15 min) para streams de larga duración.

6.2. Consultar estado de un QR (solo QR dinámico)

GET `https://api.prd.geopagos.io/api/v4/payments/intents/{qr_id}/sse`

```
curl --location 'https://api.prd.geopagos.io/api/v4/payments/intents/{qr_id}/sse' \
--header 'Accept: application/json' \
--header 'Authorization: Bearer {access_token}'
```

Semántica de la respuesta:

Código	Significado
204 No Content	El QR no fue pagado aún
200 OK	El QR fue pagado (se devuelve la autorización)

Response — 200 OK (pagado)

```
{
  "id": "3f59b249-59f3-42f7-aa12-42a99dad92b",
  "type": "SALE",
  "status": "APPROVED",
  "refNumber": "09841023-b6da-4384-a4d4-a852dc4e923e",
  "entryMode": "QR",
  "totalAmount": 85.00
}
```

El campo `refNumber` es el `reference_number` de la autorización, insumo para obtener el detalle completo (sección 6.4).

6.3. Obtener detalle del intent (solo QR dinámico)

Devuelve el estado y composición del intent de pago (venta), aún si no fue pagado.

GET `https://api.prd.geopagos.io/api/v4/payments/intents/{qr_id}`

Response — 200 OK

```
{
  "uuid": "8cf1ce0f-8969-4029-b8ba-87406971d54d",
  "state": "PENDING",
  "amount": 111.0,
  "tip_amount": 0.0,
  "currency": "032",
  "origin": "mpos_app",
  "items": [
    { "amount": 111.0, "quantity": 1, "name": "Venta" }
  ],
  "taxes": [],
  "authorizations": [],
  "metadata": {
    "can_refund": false,
    "refund_without_card": false
  }
}
```

```

},
"point_of_sale_id": "6fda9b64-d317-4723-9480-cda88579c987"
}

```

Estados observados del intent: `PENDING` (a la espera de pago) → `SALE_CONFIRMED` (venta confirmada). Cuando el intent se paga, el array `authorizations[]` contiene la autorización con su `reference_number`.

6.4. Obtener detalle de venta por número de referencia

El `reference_number` se obtiene cuando un QR se paga exitosamente, desde cualquiera de estas fuentes:

- La notificación SSE (sección 6.1)
- La consulta de estado del QR (sección 6.2, campo `refNumber`)
- La lista de movimientos (sección 6.5, campo `tracking.reference_number`)

GET https://api.prd.geopagos.io/api/v4/payments/authorizations/{reference_number}

Response — 200 OK (resumen de campos relevantes)

```

{
  "reference_number": "2147792a-a875-4651-bd01-223f803243f8",
  "type": "SALE",
  "status": "APPROVED",
  "amount": 111.00,
  "date_time": "2026-06-16T15:08:09Z",
  "currency": "032",
  "entry_mode": {
    "source": "QR",
    "qr": { "content": "0002010102...", "id": "561f4def-3874-4e67-8ab3-ac2ea65996fc" }
  },
  "payment_method": { "type": "TRANSFER" },
  "merchant": {
    "identification": { "type": "CUIT", "number": "20338975938" },
    "legal_name": "federico kucher",
    "trade_name": "federico kucher"
  },
  "processor_result": {
    "date": "2026-06-16T15:08:09Z",
    "authorization_code": "86VRPQ2GMLY40QKG2GLY0M",
    "processor_reference_number": "2147792a-a875-4651-bd01-223f803243f8"
  },
  "payment_plan": { "installment_quantity": 1 },
  "payment_intent": {
    "uuid": "561f4def-3874-4e67-8ab3-ac2ea65996fc",
    "state": "SALE_CONFIRMED",
    "point_of_sale_id": "6fda9b64-d317-4723-9480-cda88579c987"
  }
}

```

Campos clave para conciliación:

Campo	Descripción
<code>reference_number</code>	Identificador único de la autorización
<code>status</code>	Estado de la operación (<code>APPROVED</code>)
<code>entry_mode.qr.id</code> / <code>payment_intent.uuid</code>	<code>qr_id</code> del QR que originó el pago (vínculo con la venta local)
<code>processor_result.authorization_code</code>	Código de autorización del procesador
<code>payment_method.type</code>	Medio de pago utilizado (ej.: <code>TRANSFER</code>)
<code>payment_intent.point_of_sale_id</code>	Point of sale (vendedor) asociado

6.5. Obtener lista de movimientos

Permite conciliación batch de todas las operaciones del comercio. La respuesta está paginada.

GET <https://api.prd.geopagos.io/api/v4/payments/authorizations>

Filtrado por point of sale (por ejemplo, para conciliar por vendedor):

GET https://api.prn.geopagos.io/api/v4/payments/authorizations?point_of_sale={point_of_sale_uuid}

Response — 200 OK (resumen)

```
{
  "data": [
    {
      "date_time": "2026-06-16T15:08:09Z",
      "type": "SALE",
      "status": "APPROVED",
      "origin": "mpos_app",
      "amount": 111.0,
      "payment_method": { "type": "TRANSFER" },
      "payment_plan": { "installment_quantity": 1, "financial_rate": 0.0 },
      "tracking": {
        "authorization_code": "86VRPQ2GMLY40QKG2GLY0M",
        "reference_number": "2147792a-a875-4651-bd01-223f803243f8",
        "payment_intent_uuid": "561f4def-3874-4e67-8ab3-ac2ea65996fc"
      },
      "entry_mode": { "source": "QR" },
      "billing": { "currency": "032" },
      "point_of_sale": "6fda9b64-d317-4723-9480-cda88579c987"
    }
  ],
  "pagination": {
    "items_per_page": 10,
    "current_page": 1,
    "pages": 2
  }
}
```

Clave de conciliación: el campo `tracking.payment_intent_uuid` de cada movimiento es el `qr_id` del QR generado. Esto permite matchear cada pago recibido con la venta registrada localmente.

7. Flujo completo (secuencia)



8. Recomendaciones de implementación

1. **Gestión del token:** implementar un cache con renovación anticipada (el token dura 15 minutos y no hay refresh token). Ante un 401, reintentar con token nuevo.

2. **Persistencia de identificadores:** guardar `point_of_sale_id` (por vendedor) y `qr_id` (por venta). El `qr_id` es la clave que vincula la venta local con el pago recibido (`tracking.payment_intent_uuid`).
3. **Estrategia de confirmación:** priorizar SSE para experiencia en tiempo real; usar el endpoint de estado (`/intents/{qr_id}/sse`) como fallback ante desconexiones, y la lista de movimientos como mecanismo de conciliación batch (por ejemplo, un proceso de cierre diario por point of sale).
4. **Expiración de QR dinámicos:** controlar `expires_at` ; si el QR expira sin pagarse, generar uno nuevo para reintentar el cobro.
5. **Idempotencia y reintegros:** verificar `metadata.can_refund` en el intent antes de habilitar devoluciones desde la aplicación.
6. **Seguridad:** los tokens y credenciales nunca deben llegar al front-end del comercio; todas las llamadas a GeoPagos deben realizarse desde el backend.

9. Referencia rápida de endpoints

#	Operación	Método	Endpoint
1	Obtener token	POST	<code>idp.prd.geopagos.io/auth/realms/{tenant}/protocol/openid-connect/token</code>
2	Crear point of sale	POST	<code>api.geopagos.io/api/v4/points-of-sale</code>
3	Actualizar point of sale	PUT	<code>api.geopagos.io/api/v4/points-of-sale/{uuid}</code>
4	Eliminar point of sale	DELETE	<code>api.geopagos.io/api/v4/points-of-sale/{uuid}</code>
5	Listar points of sale	GET	<code>api.prd.geopagos.io/api/v4/points-of-sale/</code>
6	Crear QR (dinámico/estático)	POST	<code>api-mpos-cabal.prd.geopagos.io/api/v4/payments/qr/{pos_uuid}</code>
7	Suscripción SSE	GET	<code>api.prd.geopagos.io/sse/notifications?topic=urn:{tenant}:intent_{qr_id}</code>
8	Estado de QR (dinámico)	GET	<code>api.prd.geopagos.io/api/v4/payments/intents/{qr_id}/sse</code>
9	Detalle de intent (dinámico)	GET	<code>api.prd.geopagos.io/api/v4/payments/intents/{qr_id}</code>
10	Detalle de venta	GET	<code>api.prd.geopagos.io/api/v4/payments/authorizations/{reference_number}</code>
11	Lista de movimientos	GET	<code>api.prd.geopagos.io/api/v4/payments/authorizations[?point_of_sale={uuid}]</code>