BANCO PLAZA

Tú cuentas

Referencia de Uso: API Banco Plaza

Pago P2P/P2C

Enero 2026

Versión 4.0.0

Tabla de Contenido

Datos de Dominio

Todos los recursos descritos en el presente documento son relativos al siguiente dominio:

URL para consumo API

Ambiente	Protocolo	Dominio	Puerto
Prueba	HTTPS	openapiqa.bancoplaza.com	N/A
Productivo	HTTPS	openapi.bancoplaza.com	N/A

URL para generación de token de autenticación API

Ambiente	Protocolo	Dominio	Puerto
Prueba	HTTPS	portalapiqa.bancoplaza.com/oauth2/token	N/A
Productivo	HTTPS	portalapi.bancoplaza.com/oauth2/token	N/A

Procedimiento para la autenticación de APIs con Oauth 2.0

OAuth 2.0 es un marco de autorización (Authorization Framework) basado en principios arquitectónicos y estándares de seguridad globales. Su objetivo principal es delegar el acceso a recursos protegidos sin la necesidad de compartir las credenciales directas del usuario. En el ecosistema de WSO2, el endpoint /oauth2/token actúa como la puerta central para la gestión de acceso. Su función es realizar el intercambio de credenciales de autorización (como códigos de autorización o credenciales de aplicación) por un Access Token válido.

Requisitos Obligatorios de la Petición

Para garantizar la integridad y seguridad del proceso, todas las solicitudes al endpoint /oauth2/token deben contemplar los valores del client_id y client_secret emitidos por el WSO2 APIM

Definición: CLIENT_ID

Es el identificador único que permite al servidor reconocer qué aplicación específica está solicitando el acceso.

Ejemplo: client_id

VgV9jtzb6ozPr_mqwlckJFR5C1oa

Definición: CLIENT_SECRET

Es la clave secreta o contraseña de la aplicación que verifica su identidad y asegura que solo entidades autorizadas puedan interactuar con la API.

Ejemplo: client_secret

FJ2lciJ3xXW7LhFa2f7oZ_HKNZsa

Definición: URL TOKEN

Es el recurso de red (punto de acceso) dedicado exclusivamente a la emisión de tokens de acceso. En el modelo de OAuth 2.0, este endpoint actúa como el servidor de autorización.

Características Principales:

- Método Obligatorio: Es estrictamente de tipo POST. Esto es así porque el cuerpo de la petición contiene credenciales sensibles y porque la operación no es "idempotente" (estás creando una sesión o recurso nuevo).
- Seguridad (SSL/TLS): Al manejar credenciales, este endpoint solo debe estar disponible a través de HTTPS.
- Procesamiento: Cuando la URL recibe la petición, el Key Manager de WSO2 realiza tres pasos internos:
- Valida la autenticidad del client_id y client_secret.
- Verifica que la aplicación esté activa.
- Genera un token firmado con un tiempo de expiración definido.

Procedimiento para la solicitud de token

Antes de consumir el endpoint, necesitas las credenciales emitidas por Banco Plaza.

- client_id: Tu identificador único
- client_secret: Tu contraseña de aplicación.

La solicitud se envía mediante un método POST a la URL del consumo del token. Este paso es el intercambio formal de identidad por acceso a continuación se describen la información de acuerdo con el ambiente.

URL de consumo del token

Ambiente	Protocolo	Dominio	Puerto
Prueba	HTTPS	portalapiqa.bancoplaza.com/oauth2/token	N/A
Productivo	HTTPS	portalapi.bancoplaza.com/oauth2/token	N/A

Para realizar el proceso de consumo de las APIs se requiere la emisión del token, para ello se ha incorporado un script basado en el concepto de encadenamiento de peticiones, su funcionalidad es obtener un token y posteriormente enviarlo como parámetro al header: Authorization.

A continuación, se mencionan los parámetros a considerar remplazar su valor para consumo de las APIS en un código de ejemplo en Javascript y la explicación de este

Los valores que se deben de cambiar son:

tokenUrl = Url del token

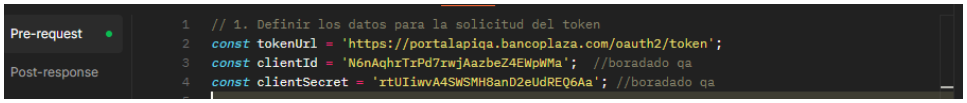
Ejemplo: https://portalapiqa.bancoplaza.com/oauth2/token

clientId = valor del client_id

Ejemplo: N6nAqhrTrPd7rwjAazbeZ4EWpWMa

clientSecret = valor del client_secret

Ejemplo: rtUIiwvA4SWSMH8anD2eUdREQ6Aa



Nota: cada uno de los recursos se deberá hacer el remplazo de la información que anteriormente se menciona

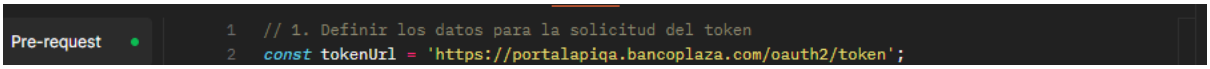
La función principal del script es mostrar el proceso a seguir para automatizar la obtención de un token de acceso (OAuth 2.0) antes de ejecutar la petición principal de la colección.

A continuación, se muestra el desglose paso a paso:

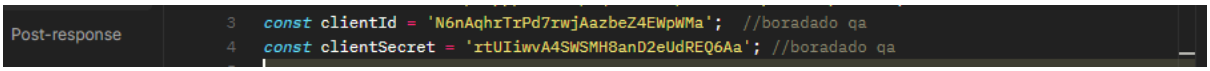
1. Configuración de Credenciales

   Se definen las variables necesarias para identificarse ante el servidor de Banco Plaza:

   - **URL del Token:** La dirección del servidor de autorización (tokenUrl).



   - **Credenciales:** Define el clientId y el clientSecret.



   - **Codificación:** Convierte estas credenciales a un formato **Base64** usando la función btoa(). Esto es un requisito estándar para el encabezado de "Authorization: Basic".

2. Definición del Objeto de Petición (requestOptions)

   Se construye la estructura de la solicitud HTTP que se enviará:

   - Método: POST.
   - Headers: Configura el tipo de contenido como application/x-www-form-urlencoded e incluye el token de autorización básica que se generó en el paso anterior.
   - Cuerpo (Body): Define que el flujo de autenticación es de tipo client_credentials, que es el usado para comunicación de servidor a servidor (sin intervención de un usuario final).

```
const requestOptions = {
    url: tokenUrl,
    method: 'POST',
    header: {
        'Authorization': 'Basic ' + base64Auth,
        'Content-Type': 'application/x-www-form-urlencoded'
    },
    body: {
        mode: 'urlencoded',
        urlencoded: [
            { key: 'grant_type', value: 'client_credentials' }
        ]
    }
};
```

3. Ejecución de la Petición (pm.sendRequest)

   El script envía la solicitud de forma asíncrona:

   - Manejo de Errores: Si algo sale mal (problemas de red, URL incorrecta), imprime el error en la consola.
   - Procesamiento de Respuesta: Si la petición es exitosa, convierte la respuesta del servidor en un objeto JSON para poder leerla.

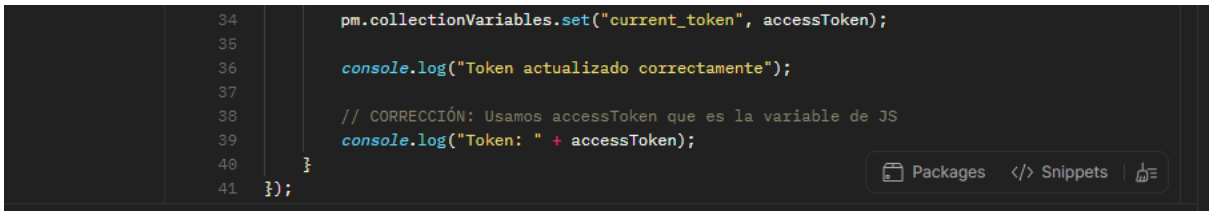
```
pm.sendRequest(requestOptions, (err, response) => {
    if (err) {
        console.log("Error obteniendo el token:", err);
    } else {
        const jsonResponse = response.json();
        const accessToken = jsonResponse.access_token;

        // 3. Guardar el token en una variable de colección
        // "current_token" es el NOMBRE de la etiqueta en Postman
        pm.collectionVariables.set("current_token", accessToken);

        console.log("Token actualizado correctamente");

        // CORRECCIÓN: Usamos accessToken que es la variable de JS
        console.log("Token: " + accessToken);
    }
});
```

4. Extracción y Almacenamiento del Token

   Esta es la parte más importante para el flujo de trabajo:

   - Extracción: Toma el valor del campo access_token que devolvió el banco.
   - Persistencia: Guarda ese valor en una variable de colección llamada current_token.

- Log: Imprime el token en la consola para que el desarrollador pueda verificar que se recibió correctamente.



Resumen de Operaciones

Pagos P2P/P2C

¿Requiere	Endpoint	Método HTTP	Permisos autenticación?
Sí	/pagos/v1/v1/pagos/p2p	POST	WRITE
Sí	/pagos/v1/v1/pagos/p2p/{id}	POST	WRITE
No	/pagos/v1/v1/pagos/p2p/bancos	GET	READ

Recurso: /pagos/v1/v1/pagos/p2p

Argumentos de Entrada

	Argumento	Tipo	Descripción
Argumentos Requeridos	banco	String	*Banco*. Se refiere al código del banco beneficiario de la transacción. Su longitud es de cuatro (4) dígitos. Por ejemplo, el código de Banco Plaza es 0138.
	idBeneficia rio	String	*Identificación Beneficiario*. Se refiere al documento de identidad del beneficiario de la transacción. Debe incluir el tipo de persona. Por ejemplo, para persona natural el valor sería V13759368. En contraste, para un comercio sería J00378944781. Longitud de doce (12) caracteres.
	telefono	String	*Teléfono*. Se refiere al número de teléfono del beneficiario de la transacción afiliado al servicio P2P/P2C. El formato que se debe utilizar para este argumento es el siguiente: Código Operadora + Número Telefónico. Por ejemplo: ● 4245378879 ● 4125387761 ● 4169873678 Se debe omitir el cero (0) en el Código de la Operadora.
	monto	Double	*Monto*. Se refiere a la cantidad de dinero que se desea transferir al beneficiario de la transacción. Es un valor numérico decimal. Por ejemplo: 500000.00.
	motivo	String	Motivo. Se refiere a una descripción breve que el ordenante quiera colocarle a la transacción. Por ejemplo: “Compra productos del hogar”. Su longitud es de 35 caracteres.
	canal	String	*Canal*. Código que busca clasificar el origen de la transacción. Los posibles valores son los siguientes: ● "20" para POS

			● "21" para MERCHANT ● "22" para VPOS ● "23" para BOTON DE PAGO ● “24” para BILLETERA DIGITAL ● Entre otros.
	id-externo	String	*ID Externo.* Se refiere a un código o ID que identifica la transacción de forma unívoca del lado del cliente que invoca. La API almacenará este ID una vez que la transacción haya sido completada de forma exitosa y validará contra este repositorio las subsecuentes transacciones para evitar duplicados.
Argumentos Opcionales	cuenta	String	*Cuenta.* Se refiere al número de cuenta de donde se realizará el débito para el pago P2P. Debe pertenecer al ordenante del pago. En el caso de omisión, se usará la cuenta afiliada al servicio de Open Bank. En el caso de que el argumento se encuentre presente, se validará que la cuenta pertenezca al propietario de la API-KEY. En caso de que la cuenta no pertenezca al propietario de la APIKEY, el sistema asumirá la petición de pago como un intento de subrogar al titular de la cuenta. En ese sentido, se validará si el propietario de la APIKEY posee privilegios para subrogar la petición, en cuyo caso, la petición será procesada de forma ordinaria. De lo contrario, será rechazada.
	telefonoAfi liado	String	*Teléfono Afiliado*. Se refiere al número de teléfono del ordenante de la transacción afiliado al servicio P2P/P2C. El formato que se debe utilizar para este argumento es el siguiente: Código Operadora + Número Telefónico. Por ejemplo: ● 4245378879 ● 4125387761 ● 4169873678 Se debe omitir el cero (0) en el Código de la Operadora.

	moneda	String	*Moneda.* Se refiere al código de moneda especificado en el estándar ISO 4217. Por ejemplo, USD para dólares americanos, VES para bolívares soberanos y EUR para euros.
	sucursal	String	*Sucursal.* Se refiere al código con el que el cliente identifica la sucursal de donde proviene la operación C2P.
	cajero	String	*Cajero*. Se refiere a un código que identifica de forma unívoca al cajero dentro del comercio. Por ejemplo: “0032”.
	caja	String	*Caja.* Se refiere al número o código de la caja dentro del comercio y desde la cual se efectuó la operación.
	ipCliente	String	*IP Cliente*. Se refiere a la dirección IP del dispositivo desde la cual se está invocando la transacción P2P/P2C. Formato IPv4, cuatro octetos separados por puntos. Por ejemplo: 192.168.1.100. **Este argumento debe ser enviado en la cabecera de la petición HTTP en caso de querer especificarlo.**
	longitud	String	*Longitud*. Parámetro para definir las coordenadas de geolocalización del ordenante de la transacción P2P/P2C. **Este argumento debe ser enviado en la cabecera de la petición HTTP en caso de querer especificarlo.**
	latitud	String	*Latitud*. Parámetro para definir las coordenadas de geolocalización del ordenante de la transacción P2P/P2C. **Este argumento debe ser enviado en la cabecera de la petición HTTP en caso de querer especificarlo.**

	precision	String	*Precisión*. Parámetro para definir las coordenadas de geolocalización del ordenante de la transacción P2P/P2C. **Este argumento debe ser enviado en la cabecera de la petición HTTP en caso de querer especificarlo.**
Permisos	WRITE		
Método HTTP	POST		
HTTP Status Code			201 Created 400 Bad Request 401 Unauthorized 404 Not Found 500 Internal Server Error

Descripción

Permite realizar pagos P2P/P2C a un tercero en nombre del cliente que invoca el recurso.

A continuación, un ejemplo de la petición:

POST /pagos/v1/p2p HTTP/1.1

Request Header

```
Content-Type: application/json

Authorization: Bearer {{current_token}}
```

Body

```
{
            "banco": "0102"
              , "idBeneficiario": "V16588736"
             , "telefono": "4245378879"
                , "telefonoAfiliado": "4127331029"
           , "monto": 150000.00
             , "motivo": "Pago almuerzo"
           , "canal": "10"
            , "tipoCuenta": "E"
```

}

Estructura de Salida

A continuación se listan los datos que componen la respuesta del servicio.

Campo	Tipo	Descripción
codigoRespuesta	String	*Código de Respuesta*. Se refiere a un código de cuatro (4) dígitos que identifica la respuesta generada por el servicio. **Este campo será agregado a la cabecera de la respuesta HTTP.**
descripcionCliente	String	*Descripción para el Cliente*. Es la descripción asociada al *Código de Respuesta* diseñada para ser mostrada o desplegada a la capa front. **Este campo será agregado a la cabecera de la respuesta HTTP.**
descripcionSistema	String	*Descripción del Sistema*. Es la descripción asociada al Código de Respuesta diseñada para albergar información técnica sobre la respuesta a la petición. **Este campo será agregado a la cabecera de la respuesta HTTP.**
fechaHora	Datetime	*Fecha y Hora*. Es una marca de tiempo o timestamp del momento exacto en que se entrega la respuesta a la petición. **Este campo será agregado a la cabecera de la respuesta HTTP.**
numeroReferencia	String	*Número de referencia.* Código generado por el sistema para identificar de forma unívoca el pago dentro del sistema interbancario.

Un ejemplo de cómo luce una respuesta es el siguiente:

Response Header
HTTP/1.1 201 Created

codigoRespuesta: 0000

descripcionCliente: Transaccion Exitosa descripcionSistema:
Transaccion Exitosa fechaHora: 2019-11-28 09:12:45

Body

{
"numeroReferencia": "872675937"
}

Códigos de Respuesta

La lista de los posibles Códigos de Respuesta se presenta a continuación:

Código de Respuesta	Descripción Sistema	Tipo de Resultado	HTTP Status Code
0000	TRANSACCIÓN EXITOSA	Transacción exitosa.	201
0002	PARÁMETRO [parámetro] OBLIGATORIO	Error de sistema.	400
E001	CLIENTE NO REGISTRADO	Validación de sistema	500
E002	TELÉFONO NO REGISTRADO	Validación de sistema	500
E003	REGISTRO NO EXISTE	Validación de sistema	404
E010	MONTO NO VALIDO	Validación de sistema	400
E012	TELÉFONO NO HABILITADO A PAGAR	Validación de sistema	500
E013	EXCEDE CANTIDAD DE PAGOS DIARIOS	Validación de sistema	500
E014	EXCEDE MONTO TOTAL DE PAGOS DIARIOS	Validación de sistema	500
E015	EXCEDE MONTO MÁXIMO PARA UN PAGO	Validación de sistema	500
E016	EXCEDE CANTIDAD DE PAGOS DIARIOS	Validación de sistema	500
E017	EXCEDE MONTO TOTAL DE PAGOS DIARIOS	Validación de sistema	500
E018	ERROR EN NÚMERO DE BANCO	Validación de sistema	400
E019	BANCO NO AUTORIZADO PARA PAGOS	Validación de sistema	500
E021	TELÉFONO RECEPTOR NO REGISTRADO	Validación de sistema	500
E022	TELÉFONO RECEPTOR NO ACEPTA PAGOS	Validación de sistema	500
E023	IDENTIF. DEL BENEFICIARIO NO COINCIDE	Validación de sistema	400

E024	CLIENTE BLOQUEADO	Validación de sistema	500
E030	BANCO NO ACTIVO EN PAGO MÓVIL	Validación de sistema	500
V002	NACIONALIDAD CEDULA/RIF NO VÁLIDA	Validación de sistema	400
0005	TIEMPO DE RESPUESTA EXCEDIDO	Validación de sistema	500
0012	TRANSACCIÓN NO VALIDA	Validación de sistema	500
0013	MONTO INVALIDO	Validación de sistema	400
0014	NRO DEL RECEPTOR ERRADO/NO AFILIADO	Validación de sistema	400
0022	CEDULA/RIF RECEPTOR ERRADA	Validación de sistema	400
0030	ERROR DE FORMATO	Validación de sistema	400
0041	SERVICIO NO ACTIVO	Error de sistema	500
0043	SERVICIO NO ACTIVO	Error de sistema	500
0051	SALDO INSUFICIENTE	Validación de sistema	400
0056	CELULAR NO COINCIDE	Validación de sistema	400
0057	NEGADA POR EL RECEPTOR	Validación de sistema	500
0062	CUENTA RESTRINGIDA	Validación de sistema	500
0080	AUTORIZADOR/RED NO DISPONIBLE	Error de sistema	500
0084	TIME-OUT	Error de sistema	500
0091	INSTITUCIÓN NO DISPONIBLE	Validación de sistema	500
0092	BANCO RECEPTOR NO AFILIADO	Validación de sistema	500
0096	ERROR EN SISTEMA	Error de sistema	500
1016	SALDO INSUFICIENTE	Validación de sistema	400
A001	FIRMA DIGITAL INVÁLIDA	Error de sistema	400
A002	FIRMA DIGITAL VENCIDA	Error de sistema	400
A003	RECURSO NO AUTORIZADO	Validación de sistema	401
A004	API-KEY INVÁLIDA O REVOCADA	Validación de sistema	500
A010	CANAL INVÁLIDO	Validación de sistema	400

Recurso: /pagos/v1/v1/pagos/p2p/{id}

Argumentos de Entrada

	Argumento	Tipo	Descripción
Argumentos Requeridos	id	String	*Identificación*. Se refiere al documento de identidad del ordenante de la transacción. Debe incluir el tipo de persona. Por ejemplo, para persona natural el valor sería V13759368. En contraste, para un comercio sería J00378944781. Longitud de doce (12) caracteres.
Argumentos Opcionales	-	-	-
Permisos	WRITE		
Método HTTP	POST		
HTTP Status Code			201 Created 400 Bad Request 401 Unauthorized 404 Not Found 500 Internal Server Error

Descripción

Permite realizar pagos P2P/P2C a un tercero en nombre del cliente que se encuentra indicado en el argumento de entrada {id}, y que es el único elemento adicional respecto a la lista de argumentos de entrada del recurso /v1/pagos/p2p.

A continuación, un ejemplo de la petición:

POST /pagos/v1/p2p/J00126647368 HTTP/1.1

Request Header

```
Content-Type: application/json

Authorization: Bearer {{current_token}}
```

Body


```
{
                "banco": "0138"
               , "id": "V16588736"
                , "telefono": "4245378879"
                   , "telefonoafiliado": "4127331029"
               , "monto": 150000.00
                , "motivo": "Pago almuerzo"
               , "canal": "10"
                , "tipoCuenta": "E"
}
```


Estructura de Salida

La estructura de salida de este recurso es la misma que se presenta en la especificación del recurso /v1/p2p/pagos.

Códigos de Respuesta

La lista de los posibles Códigos de Respuesta es la misma que la que se presenta en la especificación del recurso /v1/pagos/p2p.

Recurso: /pagos/v1/v1/pagos/p2p/bancos

Argumentos de Entrada

A continuación, se listan los argumentos de entrada que se utilizarán para definir criterios de búsqueda especializados. Todos los argumentos de entradas deben ser indicados en el QueryString.

	Argumento	Tipo	Descripción
Argumentos Requeridos	-	-	
Argumentos Opcionales	codigo	String	*Código de Banco*. Parámetro que se utiliza par filtrar la lista de bancos por el código de la entidad bancaria seleccionada. El formato de entrada es como sigue: 0138.
Permisos	READ		
Método HTTP	GET		
HTTP Status Code			200 OK 500 Internal Server Error

Descripción

Permite obtener la lista de entidades bancarias adscritas al método de pago interbancario P2P/2C. Es un recurso enteramente público.

Ejemplos de uso son los siguientes:

QueryString	Resultado Esperado
Omitida	Retorna la lista de entidades bancarias adscritas al método de pago interbancario P2P/P2C.
?codigo=[valor]	Retorna la lista de entidades bancarias adscritas al método de pago interbancario P2P/P2C filtrada por el código de la entidad bancaria cuyo código sea igual a [valor].

Estructura de Salida

Se refiere a los datos que componen la estructura de salida para el recurso. Básicamente, es la respuesta a la petición. Se describe en la siguiente tabla:

Campo	Tipo	Descripción
codigoRespuesta	String	*Código de Respuesta*. Se refiere a un código de cuatro (4) dígitos que identifica la respuesta generada por el servicio. **Este campo será agregado a la cabecera de la respuesta HTTP.**
descripcionCliente	String	*Descripción para el Cliente*. Es la descripción asociada al *Código de Respuesta* diseñada para ser mostrada o desplegada a la capa front. **Este campo será agregado a la cabecera de la respuesta HTTP.**
descripcionSistema	String	*Descripción de Sistema*. Es la descripción asociada al Código de Respuesta diseñada para albergar información técnica sobre la respuesta a la petición. **Este campo será agregado a la cabecera de la respuesta HTTP.**
fechaHora	Datetime	*Fecha y Hora*. Es una marca de tiempo o timestamp del momento exacto en que se entrega la respuesta a la petición. **Este campo será agregado a la cabecera de la respuesta HTTP.**
bancos	List<Banco>	*Lista de Bancos*. Se refiere a una lista de objetos JSON que contiene la información de cada uno de los bancos adscritos al método de pago interbancario P2P/P2C. **Este campo será agregado a la cabecera de la respuesta HTTP.**

cantidadBancos	Integer	*Cantidad de Bancos*. Se refiere a la cantidad de objetos Banco dentro de la lista bancos. **Este campo será agregado a la cabecera de la respuesta HTTP.**

Dado que bancos es una lista de objetos Banco, es necesario detallar la estructura de dicho objeto. En la siguiente tabla se listan los atributos que lo componen:

Campo	Tipo	Descripción
codigo	String	*Código de Banco*. Código de cuatro (4) dígitos que identifica a la entidad bancaria. El formato de salida es como sigue: 0138.
nombre	String	*Nombre de Banco.* Nombre de la entidad bancaria. Se refiere al nombre comercial.

A continuación, se muestra un ejemplo de una respuesta para este recurso:


```
{
        "codigoRespuesta": "0000"
          , "descripcionCliente": "Consulta exitosa."
          , "descripcionSistema": "Consulta exitosa."
        , "fechaHora": "2019-11-28 09:12:45"
      , "bancos": [
            {
                  "codigo": "0102"
                  , "banco": "Banco de Venezuela"
            },
            {
                  "codigo": "0138"
                , "banco": "Banco Plaza"
            },
            {
                  "codigo": "0115"
                  , "banco": "Banco Exterior"
            },
            {
                  "codigo": "0134"
                , "banco": "Banesco"
            }
      ]
        , "cantidadBancos": "4"
```

}

Códigos de Respuesta

La lista de los posibles Códigos de Respuesta se presenta a continuación:

Código de Respuesta	Descripción Sistema	Tipo de Resultado	Aplica comision castigo?
0000	TRANSACCIÓN EXITOSA	Transacción exitosa.	NO
0002	PARÁMETRO [parametro] OBLIGATORIO	Error de sistema.	NO
0096	ERROR EN SISTEMA	Error de sistema	NO