



Referencia de Uso: API Banco Plaza

Pago P2P/P2C

Noviembre 2019

Versión 3.3.3

Tabla de Contenido

Tabla de Contenido	2
Datos de Dominio	3
Procedimiento para el firmado de las peticiones	4
Definición: API-KEY	4
Definición: API-KEY-SECRET	4
Definición: NONCE	4
Definición: API-SIGNATURE	5
Resumen de Operaciones	6
Pagos P2P/P2C	6
Recurso: /v1/pagos/p2p	7
Argumentos de Entrada	7
Descripción	10
Estructura de Salida	11
Códigos de Respuesta	12
Recurso: /v1/pagos/p2p/{id}	14
Argumentos de Entrada	14
Descripción	14
Estructura de Salida	15
Códigos de Respuesta	15
Recurso: /v1/pagos/p2p/bancos	16
Argumentos de Entrada	16
Descripción	16
Estructura de Salida	16
Códigos de Respuesta	18

Datos de Dominio

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

Ambiente	Protocolo	Dominio	Puerto
Prueba	HTTPS	apiqa.bancoplaza.com	8585
Productivo	HTTPS	api.bancoplaza.com	8282

Procedimiento para el firmado de las peticiones

Todas aquellas peticiones a recursos que requieran autenticación (ver la sección *Resumen de Operaciones*) deben ser firmados usando un algoritmo HMAC basado en SHA-384. Éste es el mecanismo utilizado para validar que las peticiones provienen de una fuente de confianza.

Para ello, el cliente debe contar dos elementos importantes: Un `api-key` y un `api-key-secret`.

Definición: API-KEY

Llave única de 32 caracteres que identifica al cliente que realiza la petición sobre el recurso (URL). **Debe ser enviado en la cabecera de cada petición que requiera autenticación.**
Ejemplo:

```
api-key = f09a0c81d70542deb6e6bfb66d24c845
```

Este parámetro es generado por el cliente a través del portal **Tu Plaza Línea Empresa**. Es de uso exclusivo del cliente.

Definición: API-KEY-SECRET

Llave única de 32 caracteres que hace las veces de un código secreto con el que el cliente firma todas las peticiones que envía al banco. Dado que es elemento de seguridad que se encuentra en custodia tanto del cliente como del banco, sirve entonces para comprobar que toda petición que haya sido firmada con el mismo, provenga del dueño legítimo, es decir, del cliente.

Éste parámetro **no se envía** en ninguna sección de la petición HTTP. Como se indicó, solo se usa para firmar digitalmente la petición.

En términos de estructura, es idéntico al `api-key`. Por ejemplo:
`bd18e09732874b04b4046f9c311426e3`.

Definición: NONCE

Cada petición requiere también de un nonce. Un nonce es un número entero que se debe incrementar con cada petición. Su valor, en cada petición, debe ser mayor que el de la petición previa. Se usa para garantizar que la firma de cada petición sea diferente, de tal modo que evita vectores de ataque que eventualmente puedan copiar la estructura de una petición entera y enviarla en un momento posterior. El nonce debe enviarse como un dato en la cabecera de la petición. Por ejemplo:

```
nonce = 616766488366836
```

La forma recomendada para la generación un nonce es a través de una función que retorne un unix timestamp en milisegundos. Por ejemplo:

- En Javascript: `Date.now()`
- En Java: `System.currentTimeMillis()`
- En PHP: `round(microtime(true) * 1000)`

Definición: API-SIGNATURE

Se refiere a la firma de la petición propiamente dicha. Para firmar cada petición se debe:

1. Concatenar el path del recurso al que se le alude en la petición, con el nonce y con el cuerpo (JSON) de la petición.
2. Firmar el resultado de la concatenación anterior con el `api-key-secret` haciendo uso de una función `HMAC.SHA-384`.
3. El resultado de la función `HMAC.SHA-384` (48 bytes) debe ser convertido a una cadena hexadecimal (96 caracteres). Dicha cadena **es lo que se coloca en la cabecera** de la petición bajo el parámetro `api-signature`.

A continuación, se muestra un código de ejemplo en Javascript:

```
1  const CryptoJS = require('crypto-js') // Standard JavaScript cryptography library
2  const request = require('request') // "Request" HTTP req library
3
4  const apiKey = 'f09a0c81d70542deb6e6bfb66d24c845' // api-key entregado por Banco Plaza
5  const apiSecret = 'bd18e09732874b04b4046f9c311426e3' // api-key-secret entregando por Banco Plaza
6
7  const apiPath = 'pagos/v0/p2p' // Path al recurso
8
9  const nonce = (Date.now() * 1000).toString() // Generación del nonce generator. Timestamp * 1000
10 const body = {} // Cuerpo de la petición JSON. Puede variar dependiendo del recurso que se esté invocando
11
12 let signature = `/${apiPath}${nonce}${JSON.stringify(body)}`
13 // Compuesto por el Path de la URL, el nonce, y el cuerpo de la petición
14
15 const sig = CryptoJS.HmacSHA384(signature, apiSecret).toString()
16 // Se firma la petición on la función HMAC usando la llave secreta o api-key-secret
17
18 const options = {
19   url: 'https://apiqa.bancoplaza.com:8585/${apiPath}',
20   headers: {
21     'api-key': apiKey,
22     'api-signature': sig,
23     'nonce': nonce
24   },
25   body: body,
26   json: true
27 }
28
29 request.post(options, (error, response, body) => {
30   console.log(body); // Se imprime la respuesta
31 })
32
```

Resumen de Operaciones

Pagos P2P/P2C

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

Recurso: /v1/pagos/p2p

Argumentos de Entrada

	Argumento	Tipo	Descripción
Argumentos Requeridos	banco	String	<i>Banco</i> . 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.
	idBeneficiario	String	<i>Identificación Beneficiario</i> . 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	<i>Teléfono</i> . 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: <ul style="list-style-type: none">• 4245378879• 4125387761• 4169873678 Se debe omitir el cero (0) en el Código de la Operadora.
	monto	Double	<i>Monto</i> . 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	<i>Motivo</i> . 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	<i>Canal</i> . Código que busca clasificar el origen de la transacción. Los posibles valores son los

siguientes:

- “07” para POS
- “10” para MERCHANT
- “11” para VPOS
- “12” para BOTÓN DE PAGO
- “13” para BILLETERAS DIGITALES

Argumentos Opcionales	id-externo	String	<i>ID Externo.</i> 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.
	cuenta	String	<i>Cuenta.</i> 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 API-KEY, 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 API-KEY posee privilegios para subrogar la petición, en cuyo caso, la petición será procesada de forma ordinaria. De lo contrario, será rechazada.
	telefonoAfiliado	String	<i>Teléfono Afiliado.</i> 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: <ul style="list-style-type: none">• 4245378879• 4125387761• 4169873678

		Se debe omitir el cero (0) en el Código de la Operadora.
sucursal	String	<i>Sucursal.</i> Se refiere al código con el que el cliente identifica la sucursal de donde proviene la operación C2P.
cajero	String	<i>Cajero.</i> Se refiere a un código que identifica de forma unívoca al cajero dentro del comercio. Por ejemplo: “0032”.
caja	String	<i>Caja.</i> Se refiere al número o código de la caja dentro del comercio y desde la cual se efectuó la operación.
ipCliente	String	<i>IP Cliente.</i> 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	<i>Longitud.</i> 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	<i>Latitud.</i> 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	<i>Precisión.</i> 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

```
Host: api.bancoplaza.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:76.0)
Accept: text/html,application/xhtml+xml,application/xml
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Api-key: bd18e09732874b04b4046f9c311426e3
Nonce: 758380027754
Api-signature:
b3e3742e9528bac62cbd1331a4976e29925e92d12361b49be6b09a8e33593f055152b2f1ef4100
05ff5fad49034b816e
ipCliente: 201.54.78.123
Longitud: 40.689247
Latitud: -74.044502
Precision: 13
```

Body

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

Estructura de Salida

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

Campo	Tipo	Descripción
codigoRespuesta	String	<i>Código de Respuesta.</i> 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	<i>Descripción para el Cliente.</i> Es la descripción asociada al <i>Código de Respuesta</i> diseñada para ser mostrada o desplegada a la capa front. Este campo será agregado a la cabecera de la respuesta HTTP.
descripcionSistema	String	<i>Descripción del Sistema.</i> Es la descripción asociada al <i>Código de Respuesta</i> diseñada para albergar información técnica sobre la respuesta a la petición. Este campo será agregado a la cabecera de la respuesta HTTP.
fechaHora	Datetime	<i>Fecha y Hora.</i> 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	<i>Número de referencia.</i> 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: /v1/pagos/p2p/{id}

Argumentos de Entrada

	Argumento	Tipo	Descripción
Argumentos Requeridos	id	String	<i>Identificación.</i> 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 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

```
Host: api.bancoplaza.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:76.0)
Accept: text/html,application/xhtml+xml,application/xml
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

Api-key: bd18e09732874b04b4046f9c311426e3
Nonce: 758380027754
Api-signature:
b3e3742e9528bac62cbd1331a4976e29925e92d12361b49be6b09a8e33593f055152b2f1ef4100
05ff5fad49034b816e
ipCliente: 201.54.78.123
Longitud: 40.689247
Latitud: -74.044502
Precision: 13

Body

```
{  
    "banco": "0102"  
    , "id": "V16588736"  
    , "telefono": "4245378879"  
    , "telefonoafiliado": "4127331029"  
    , "monto": 150000.00  
    , "motivo": "Pago almuerzo"  
}
```

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: /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	<i>Código de Banco.</i> Parámetro que se utiliza para 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	<p><i>Código de Respuesta.</i> Se refiere a un código de cuatro (4) dígitos que identifica la respuesta generada por el servicio.</p> <p>Este campo será agregado a la cabecera de la respuesta HTTP.</p>
descripcionCliente	String	<p><i>Descripción para el Cliente.</i> Es la descripción asociada al <i>Código de Respuesta</i> diseñada para ser mostrada o desplegada a la capa front.</p> <p>Este campo será agregado a la cabecera de la respuesta HTTP.</p>
descripcionSistema	String	<p><i>Descripción del Sistema.</i> Es la descripción asociada al <i>Código de Respuesta</i> diseñada para albergar información técnica sobre la respuesta a la petición.</p> <p>Este campo será agregado a la cabecera de la respuesta HTTP.</p>
fechaHora	Datetime	<p><i>Fecha y Hora.</i> Es una marca de tiempo o timestamp del momento exacto en que se entrega la respuesta a la petición.</p> <p>Este campo será agregado a la cabecera de la respuesta HTTP.</p>
bancos	List<Banco>	<p><i>Lista de Bancos.</i> 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.</p>
cantidadBancos	Integer	<p><i>Cantidad de Bancos.</i> Se refiere a la cantidad de objetos Banco dentro de la lista bancos.</p> <p>Este campo será agregado a la cabecera de la respuesta HTTP.</p>

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	<i>Código de Banco.</i> Código de cuatro (4) dígitos que identifica a la entidad bancaria. El formato de salida es como sigue: 0138.
nombre	String	<i>Nombre de Banco.</i> Nombre de la entidad bancaria. Se refiere al nombre comercial.

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

Response Header

HTTP/1.1 200 OK

```
codigoRespuesta: 0000
descripcionCliente: Transaccion Exitosa
descripcionSistema: Transaccion Exitosa
fechaHora: 2019-11-28 09:12:45
cantidadBancos: 4
```

Body

```
{
  "bancos": [
    {"codigo": "0102", "nombre": "Banco de Venezuela"},
    {"codigo": "0138", "nombre": "Banco Plaza"},
    {"codigo": "0115", "nombre": "Banco Exterior"},
    {"codigo": "0134", "nombre": "Banesco"}
  ]
}
```

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.	200
0096	ERROR EN SISTEMA	Error de sistema	500