

# BANCO PLAZA



*Tú cuentas*

## Referencia de Uso: A PI Banco Plaza

**Domiciliación a cuenta**

**Enero 2026**

**Versión 4.0.0**

# Tabla de Contenido

Tabla de Contenido	2
Datos de Dominio	3
Procedimiento para la autenticación de APIs con Oauth 2.0	3
Resumen de Operaciones	9
Pagos CCE Inmediata	9
<b>Recurso: /cce/dom/v1/v1/cce/dom/cobroDI/Id</b>	10
Argumentos de Entrada	10
Descripción	12
Estructura de Salida	13
Códigos de Respuesta	14
<b>Recurso: /cce/dom/v1/v1/cce/dom/consultaLiq/Id</b>	15
Argumentos de Entrada	15
Descripción	17
Estructura de Salida	17
Códigos de Respuesta	18

# Datos de Dominio

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

## URL para consumo API

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

## URL para generación de token de autenticación API

<b>Ambiente</b>	<b>Protocolo</b>	<b>Dominio</b>	<b>Puerto</b>
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`

`VgV9jtz6ozPr_mqwIckJFR5C1oa`

## 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`

`FJ2IciJ3xXW7LhFa2f7oZ_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 reemplazar 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

```
Pre-request 1 // 1. Definir los datos para la solicitud del token
2 const tokenUrl = 'https://portalapiqa.bancoplaza.com/oauth2/token';
Post-response 3 const clientId = 'N6nAqhrTrPd7rwjAazbeZ4EWpWMa'; //boradado qa
4 const clientSecret = 'rtUiIwvA4SWSMH8anD2eUdREQ6Aa'; //boradado qa
```

**Nota:** cada uno de los recursos se deberá hacer el reemplazo 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).

```
Pre-request 1 // 1. Definir los datos para la solicitud del token
2 const tokenUrl = 'https://portalapiqa.bancoplaza.com/oauth2/token';
```

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

```
Post-response 3 const clientId = 'N6nAqhrTrPd7rwjAazbeZ4EwpWma'; //boradado qa
4 const clientSecret = 'rtUIiwvA4SWSMH8anD2eUdREQ6Aa'; //boradado qa
```

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

```
7 const base64Auth = btoa(clientId + ":" + clientSecret);
```

## 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).

```
9 const requestOptions = {
10   url: tokenUrl,
11   method: 'POST',
12   header: {
13     'Authorization': 'Basic ' + base64Auth,
14     'Content-Type': 'application/x-www-form-urlencoded'
15   },
16   body: {
17     mode: 'urlencoded',
18     urlencoded: [
19       { key: 'grant_type', value: 'client_credentials' }
20     ]
21   }
22 };
23
```

## 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.

```
25 pm.sendRequest(requestOptions, (err, response) => {
26     if (err) {
27         console.log("Error obteniendo el token:", err);
28     } else {
29         const jsonResponse = response.json();
30         const accessToken = jsonResponse.access_token;
31
32         // 3. Guardar el token en una variable de colección
33         // "current_token" es el NOMBRE de la etiqueta en Postman
34         pm.collectionVariables.set("current_token", accessToken);
35
36         console.log("Token actualizado correctamente");
37
38         // CORRECCIÓN: Usamos accessToken que es la variable de JS
39         console.log("Token: " + accessToken);
40     }
41 });
```

#### 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.

```
34 pm.collectionVariables.set("current_token", accessToken);
35
36 console.log("Token actualizado correctamente");
37
38 // CORRECCIÓN: Usamos accessToken que es la variable de JS
39 console.log("Token: " + accessToken);
40 }
41 });
```

# Resumen de Operaciones

## Pagos CCE Inmediata

Requiere autenticación	Endpoint	Método HTTP	Permisos
Sí	<a href="/cce/dom/v1/v1/cce/dom/cobroDI/Id">/cce/dom/v1/v1/cce/dom/cobroDI/Id</a>	POST	WRITE
Sí	<a href="/cce/dom/v1/v1/cce/dom/consultaLiq/Id">/cce/dom/v1/v1/cce/dom/consultaLiq/Id</a>	POST	READ

## Recurso: /cce/dom/v1/v1/cce/dom/cobroDI/Id

### Argumentos de Entrada

	Argumento	Tipo	Descripción
<b>Argumentos Requeridos</b>	moneda	String	Código de la Moneda. Se refiere al código de la moneda asociado a la cuenta. Utiliza el estándar ISO 4217. VES.
	canal	String	<i>Canal</i> . Código que busca clasificar el origen de la transacción. Los posibles valores son los siguientes:  "23" para BOTON DE PAGO/Open Bank
	Id	String	Identificación del beneficiario del cobro. Se refiere al documento de identidad de la empresa o comercio que realiza la cobranza de la domiciliación. Debe incluir el tipo de persona.  Por ejemplo, para un comercio sería J00378944781.  Longitud de doce (12) caracteres.
	cuenta_cobrador	String	Cuenta del beneficiario del Cobro. Se refiere al número de cuenta de donde se realizará el crédito. Debe pertenecer a la identificación del beneficiario del cobro. El formato que debe tener los 20 caracteres numéricos de la cuenta.  Ejemplos: 01380011130117777777.  No debe llevar ningún carácter especial.
	contrato	String	Número que identifica el contrato de domiciliación del cliente  Máximo 15 caracteres alfanumérico
	fecha_contrato	Date	Fecha del contrato del cliente.

			El formato para el argumento es el ISO 8601. Ej. YYYY-MM-DD
	identificacion_d	String	Identificación del deudor. Se refiere al documento de identidad del deudor (pagador) en 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.
	cuenta_deudor	String	Cuenta del deudor. Se refiere al número de cuenta a donde se realizará el débito por el cobro. Pertenece a la identificación del deudor. El formato que debe tener los 20 caracteres numéricos de la cuenta.  Ejemplos: 01380011130118888888 01020011130118888888
	nombre_d	String	Nombre del deudor. Nombre del deudor (pagador). Longitud de (80) caracteres alfanuméricos. Ejemplo: Doris Wells
	monto	Double	Monto del cobro. Se refiere a la cantidad de dinero que se desea cobrar al deudor de la transacción. Es un valor numérico decimal. Por ejemplo: 100000.00
	concepto	String	Concepto: Se refiere a un texto asociado al cobro, definido por el cliente ordenante del cobro.  Máximo 60 caracteres alfanumérico
	direccion_ip	String	IP Cliente. Se refiere a la dirección IP del dispositivo desde la cual se está invocando la transacción. 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.
<b>Argumentos</b>	factura	String	Número que identifica la factura del

<b>Opcionales</b>		cliente
		Máximo 12 caracteres alfanumérico
	fecha_factura	Date Fecha de la factura
		El formato para el argumento es el ISO 8601. Ej. YYYY-MM-DD
	fecha_vence_factura	Date Fecha de vencimiento de la factura
		El formato para el argumento es el ISO 8601. Ej. YYYY-MM-DD
<b>Permisos</b>	WRITE	
<b>Método HTTP</b>	POST	
<b>HTTP Status Code</b>		200 Accepted 204 No Content 400 Bad Request 401 Unauthorized 500 Internal Server Error

## Descripción

Permite iniciar una transacción de cobro de domiciliación interbancaria en nombre del cliente que invoca el recurso.

Algunos ejemplos de uso podrían ser los siguientes:

**POST /cce/dom/v1/v1/cce/dom/cobroDI/Id HTTP/1.1**

### Request Header

Content-Type: application/json

Authorization: Bearer {{current\_token}}

### Body

```
{
  "moneda": "VES"
  ,"canal": "23"
  ,"Id": "J00001888999"
  ,"cuenta_cobrador": "01380011130110249976"
  ,"contrato": "202411151239"
  ,"fecha_contrato": " 2024-11-28"
  ,"identificacion_d": "J00345685467"
  ,"cuenta_deudor": "01340011130110248643"
```

```

    ,"nombre_d": "Doris Wells"
    ,"monto": 53098.50
    ,"concepto": "PAGO ID5677"
    ,"direccion_ip": "127.0.0.1"
    ,"factura": ""
    ,"fecha_factura": ""
    ,"fecha_vence_factura": ""
}

```

## Estructura de Salida

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

<b>Campo</b>	<b>Tipo</b>	<b>Descripción</b>
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><b>Este campo será agregado a la cabecera de la respuesta HTTP.</b></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><b>Este campo será agregado a la cabecera de la respuesta HTTP.</b></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><b>Este campo será agregado a la cabecera de la respuesta HTTP.</b></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><b>Este campo será agregado a la cabecera de la respuesta HTTP.</b></p>
referencia_c	String	<p>Número de referencia. Código enviado por cliente o generado por el sistema para identificar de forma unívoca el pago dentro del sistema interbancario</p> <p>Máximo 9 caracteres alfanumérico</p>

transaccionId	String	Identificador único en el sistema para seguimiento de la operación.  Máximo 32 caracteres alfanumérico
---------------	--------	--

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

#### Response Header

HTTP/1.1 200 OK

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

#### Body

```
{
  "referencia_c": "45678976"
  "transaccionId": "6d60902aacf64fc0b28aa1f09cf07728"
}
```

## 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/ENVIADA A BCV	Transacción exitosa.	200
0002	PARÁMETRO [parámetro] OBLIGATORIO	Validación de sistema	400
0003	PARÁMETRO [parámetro] NO CUMPLE FORMATO	Validación de sistema	400
0004	SISTEMA FUERA DE LA FRANJA ESTABLECIDA	Validación de sistema	200
9999	ERROR INTERNO	Error de sistema	500
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	401
A12	Saldo Insuficiente	Validación de sistema	204

AB07	AGENTE RECEPTOR FUERA DE LÍNEA	Validación de sistema	400
900	Sistema Fuera de Línea	Validación de sistema	204
901	En este momento no se puede procesar la operación	Validación de sistema	204
0030	Cedula Beneficiaria no coincide con la Cuenta Destino	Validación de sistema	204
0032	Excede cantidad de operaciones diarias	Validación de sistema	204
0033	Excede limite diario	Validación de sistema	204
0035	Beneficiario no registrado por el cliente Ordenante	Validación de sistema	204
0036	Cliente no posee beneficiarios	Validación de sistema	204
0042	Saldo insuficiente	Validación de sistema	204
0045	Cuenta no permite débitos	Validación de sistema	204
0048	Cuenta no permite créditos	Validación de sistema	204
0049	Excede el límite configurado. Por favor valide el límite establecido e intente nuevamente	Validación de sistema	204
103	Moneda no coincide con cuenta crédito	Validación de sistema	204
105	Monto Invalido no se encuentra en el Rango Permitido	Validación de sistema	204
106	En este momento no podemos procesar esta operación, intente nuevamente	Validación de sistema	204
109	La identificación y la cuenta del deudor no están afiliados al servicio de cobro inmediato	Validación de sistema	204

**Recurso: /cce/dom/v1/v1/cce/dom/consultaLiq/Id**

Argumentos de Entrada

Argumento	Tipo	Descripción
-----------	------	-------------

<b>Argumentos Requeridos</b>	transaccionId	String	Identificador único en el sistema para seguimiento de la operación.  Máximo 32 caracteres alfanumérico
	canal	String	<i>Canal</i> . Código que busca clasificar el origen de la transacción. Los posibles valores son los siguientes:  "23" para BOTON DE PAGO/Open Bank
	Id	String	Identificación del beneficiario del cobro. Se refiere al documento de identidad de la empresa o comercio que realiza la cobranza de la domiciliación. Debe incluir el tipo de persona.  Por ejemplo, para un comercio sería J00378944781.  Longitud de doce (12) caracteres.
	cuenta_cobrador	String	Cuenta del beneficiario del Cobro. Se refiere al número de cuenta de donde se realizará el crédito. Debe pertenecer a la identificación del beneficiario del cobro. El formato que debe tener los 20 caracteres numéricos de la cuenta.  Ejemplos: 01380011130117777777.  No debe llevar ningún carácter especial.
	direccion_ip	String	IP Cliente. Se refiere a la dirección IP del dispositivo desde la cual se está invocando la transacción. 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.
<b>Argumentos Opcionales</b>	-	-	-
<b>Permisos</b>	READ		
<b>Método HTTP</b>	POST		

---

<b>HTTP Status Code</b>	200 Accepted
	204 No Content
	400 Bad Request
	401 Unauthorized
	500 Internal Server Error

---

## Descripción

Permite consultar el estatus o resultado de la cobranza que fue enviada previamente por el `/v1/cce/dom/cobroDI/Id`

Algunos ejemplos de uso podrían ser los siguientes:

**POST /cce/dom/v1/v1/cce/dom/consultaLiq/Id HTTP/1.1**

### Request Header

Content-Type: application/json

Authorization: Bearer {{current\_token}}

### Body

```
{
  "transaccionId": "6d60902aacf64fc0b28aa1f09cf07728"
  , "canal": "23"
  , "Id": "J00001888999"
  , "cuenta_cobrador": "01380011130110249976"
  , "direccion_ip": "127.0.0.1"
}
```

## 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.  <b>Este campo será agregado a la cabecera de la respuesta HTTP.</b>
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.

---

		<b>Este campo será agregado a la cabecera de la respuesta HTTP.</b>
descripcionSistema	String	<i>Descripción del Sistema.</i> Es la descripción asociada al Código de Respuesta diseñada para albergar información técnica sobre la respuesta a la petición.
		<b>Este campo será agregado a la cabecera de la respuesta HTTP.</b>
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.
		<b>Este campo será agregado a la cabecera de la respuesta HTTP.</b>

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

#### Response Header

HTTP/1.1 200 OK

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

## Códigos de Respuesta

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

<b>Código de Respuesta</b>	<b>Descripción Sistema</b>	<b>Tipo de Resultado</b>	<b>HTTP Status Code</b>
0000	TRANSACCIÓN PROCESADA CON EXITO	Transacción exitosa.	200
PE	PENDIENTE POR LIQUIDACION (DEL BCV)	Transacción exitosa.	200
NE	ID TRANSACCIÓN NO EXISTE	Transacción exitosa.	200
0004	SISTEMA FUERA DE LA FRANJA ESTABLECIDA	Validación de sistema	200
0002	PARÁMETRO [PARÁMETRO] OBLIGATORIO	Validación de sistema	400
0003	PARÁMETRO [PARÁMETRO] NO CUMPLE FORMATO	Validación de sistema	400
9999	ERROR INTERNO	Error de sistema	500

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	401
AC01	EL NÚMERO DE CUENTA NO ES VÁLIDA O FALTA.	Validación de sistema	400
AC04	EL NÚMERO DE CUENTA ENCUENTRA CANCELADO POR PARTE DEL BANCO RECEPTOR.	Validación de sistema	400
AC06	LA CUENTA ESPECIFICADA ESTÁ BLOQUEADA POR PARTE DEL BANCO RECEPTOR	Validación de sistema	400
AG01	TRANSACCIÓN RESTRINGIDA EN ESTE TIPO DE CUENTA	Validación de sistema	400
AM04	MONTO INSUFICIENTE, NO PUEDE CUBRIR EL MONTO ESPECIFICADO	Validación de sistema	400
BE01	LA IDENTIFICACIÓN DEL CLIENTE FINAL NO ES COHERENTE CON EL NÚMERO DE CUENTA ASOCIADO.	Validación de sistema	400
CUST	SOLICITUD DE CANCELACIÓN REALIZADA POR EL DEUDOR	Validación de sistema	400
MD01	EL CLIENTE ACREEDOR NO SE HA AFILIADO POR EL CLIENTE DEUDOR	Validación de sistema	400
MD09	EL CLIENTE ACREEDOR SE ENCUENTRA EN ESTADO INACTIVO EN LA LISTA DEL CLIENTE DEUDOR	Validación de sistema	400
MD15	CANTIDAD A COBRAR POR EL ACREEDOR NO CORRESPONDE A PARÁMETROS ASOCIADOS A LÍMITES DE MONTO ESTABLECIDOS POR EL CLIENTE DEUDOR	Validación de sistema	400
MD21	TRANSACCIÓN A COBRAR POR EL ACREEDOR NO CUMPLE CON UNO O VARIOS A PARÁMETROS ESTABLECIDOS POR EL CLIENTE DEUDOR	Validación de sistema	400
MD22	EL CLIENTE ACREEDOR SE ENCUENTRA EN ESTADO SUSPENDIDO EN LA LISTA DEL CLIENTE DEUDOR	Validación de sistema	400
TKCM	CÓDIGO ÚNICO DE OPERACIÓN DE ACEPTACIÓN DE DÉBITO INCORRECTO	Validación de sistema	400
AB07	AGENTE RECEPTOR FUERA DE LÍNEA	Validación de sistema	400
VE01	RECHAZOS TÉCNICO DEL BANCO RECEPTOR	Validación de sistema	400
RC08	CÓDIGO DEL BANCO NO EXISTE EN EL SISTEMA DE COMPENSACIÓN /LIQUIDACIÓN.	Validación de sistema	400
AM03	EL MONTO ESPECIFICADO SE ENCUENTRA EN UNA MONEDA NO DEFINIDA EN LOS ACUERDOS ESTABLECIDOS	Validación de sistema	400

