



MANUAL GENERAL DE INTEGRACIÓN ESTÁNDAR EN JAVA PAGOEFFECTIVO- COMERCIO ELECTRONICO

CONTENIDO

| | |
|---|----|
| 1. OBJETIVO..... | 5 |
| 2. ALCANCE..... | 5 |
| 3. GENERALIDADES..... | 5 |
| 3.1. ¿Qué es PagoEfectivo? | 5 |
| 3.2. ¿Cómo funciona? | 5 |
| 3.3. Estados de PagoEfectivo..... | 5 |
| 3.4. Requerimientos técnicos | 6 |
| 4. AMBIENTES DE INTEGRACION | 6 |
| 4.1. Ambiente de Integraciones..... | 6 |
| 4.2. Ambiente de producción | 6 |
| 5. ESPECIFICACIONES TÉCNICAS A ALTO NIVEL | 7 |
| 5.1. Diagrama de flujo técnico | 7 |
| 5.2. Descripción del diagrama de flujo técnico..... | 7 |
| 6. ESPECIFICACION DE TÉCNICAS A DETALLE..... | 8 |
| 6.1. Configuración de parámetros: | 8 |
| 6.2. Configuración de rutas: | 9 |
| 6.3. Generación del CIP | 10 |
| 6.4. Request URLNotificacion (notificación de estados del CIP)..... | 15 |
| 6.5. Consultar CIP | 18 |
| 6.6. Eliminar CIP..... | 22 |
| 6.7. Actualizar CIP | 24 |
| 6.8. Encriptación de recuperar tramas..... | 26 |
| 6.9. Desencriptación de tramas | 26 |

1. OBJETIVO

El objetivo de este documento es explicar cómo funciona la pasarela de pago PagoEfectivo y cómo este puede ser integrado en la plataforma de un comercio afiliado.

2. ALCANCE

El documento presenta la información del proceso de integración entre PagoEfectivo y la plataforma del comercio afiliado, así mismo, se incluye las especificaciones técnicas resaltando la descripción del contrato de servicios web y el sistema de notificación.

3. GENERALIDADES

3.1. ¿Qué es PagoEfectivo?

PagoEfectivo es el servicio de Pasarela de Pago para efectuar transacciones por Internet del Grupo El Comercio y busca brindar una alternativa de pago vía internet a toda persona que desee realizar pagos online sin la necesidad de contar con tarjetas de crédito o débito.

3.2. ¿Cómo funciona?

El proceso consiste en generar un Código de identificación de pago (CIP) que luego el usuario podrá pagarlo acercándose a uno de nuestros centros de pago afiliado (más de 45000 puntos a nivel nacional); cuando se efectúe el pago, automáticamente el centro de pago notificará al sistema PagoEfectivo y a la vez PagoEfectivo notificará al comercio afiliado que el pago se ha realizado satisfactoriamente.

Ver <https://www.youtube.com/watch?v=N1NJabZgUKQ>

3.3. Estados de PagoEfectivo

Cuando el cliente decide pagar con PagoEfectivo, se crea un CIP desde la plataforma del comercio afiliado, este CIP nace con estado Pendiente de pago.

EL CIP creado cambia a estado Cancelado sólo cuando el cliente paga el CIP a través de uno de nuestros centros autorizados (Banco de crédito del Perú, BBVA Banco Continental, Banco Scotiabank, Banco Interbank, Banco Banbif, Pago de servicios Western Union, FullCarga y Kasnet).

En caso ocurra un error en la transacción y el centro de pago *extorne* la operación, este notificará a PagoEfectivo y la vez PagoEfectivo notificará al comercio afiliado; el CIP que estaba en estado Cancelado pasa a estado Pendiente de pago.

Cuando se crea un CIP, este tiene una fecha de expiración (varía dependiendo del rubro del negocio), si aún no se ha pagado el CIP y la fecha de expiración vence, el CIP pasa de estado Pendiente de pago ha estado Expirado.

En caso el comercio afiliado decida que el CIP no debe pagarse, puede eliminar el CIP lógicamente (siempre en cuando este aún se encuentre en estado pendiente de pago), evitando que el usuario pague el CIP. El CIP pasa de estado Pendiente de pago ha estado Eliminado.

3.4. Requerimientos técnicos

Los requisitos para integrarse con la pasarela PagoEfectivo son:

- Ambiente de integración y ambiente de producción.
- IP fija pública (integración y producción).
- SOAP Client
- Tener acceso a su código fuente.

4. AMBIENTES DE INTEGRACION

PagoEfectivo cuenta con dos ambientes para el proceso de integración, un ambiente de INTEGRACIONES y otro ambiente de PRODUCCIÓN.

Dichos ambientes son los siguientes:

4.1. Ambiente de Integraciones

Ambiente donde se desarrollará inicialmente el proyecto de integración.

| Parámetro | URL |
|-------------------|---|
| UriSPE | https://pre1a.pagoefectivo.pe/ |
| UriPEService | https://pre1a.pagoefectivo.pe/PagoEfectivoWSGeneralv2/service.asmx |
| UriPECryptography | https://pre1a.pagoefectivo.pe/PagoEfectivoWSCrypto/WSCrypto.asmx |

4.2. Ambiente de producción

Ambiente final donde el comercio afiliado podrá realizar los recaudos.

| Parámetro | URL |
|-------------------|---|
| UriSPE | https://cip.pagoefectivo.pe/ |
| UriPEService | https://cip.pagoefectivo.pe/PagoEfectivoWSGeneralv2/service.asmx |
| UriPECryptography | https://cip.pagoefectivo.pe/PagoEfectivoWSCrypto/WSCrypto.asmx |

5. ESPECIFICACIONES TÉCNICAS A ALTO NIVEL

5.1. Diagrama de flujo técnico



5.2. Descripción del diagrama de flujo técnico

5.2.1. Invocar al WS Solicitar Pago (Generación de CIP)

PagoEfectivo pone a disposición un servicio web que le permitirá generar CIP (Método solicitar pago), para ello debe enviar datos relevantes de la transacción, tales como monto, moneda, nro transacción, nombre y apellidos del cliente, entre otros. Toda esta información es enviada en una trama XML encriptada.

El servicio web retornará dos valores importantes: el nro token y el código de pago (CIP). Con el nro token se podrá presentar la generación de CIP (Iframe o redirect) ^(5.2.6)

5.2.2. Request URLNotificacion

Luego de generado el CIP este puede ser pagado, expirado o incluso extornado (reversa de pago), en estos casos PagoEfectivo notifica al portal mediante una petición request enviada con método post a una página de notificación del portal configurada previamente en el sistema PagoEfectivo. ^(5.2.6)

5.2.3. Consultar CIP

PagoEfectivo pone a disposición un servicio web que le permitirá consultar un CIP (Método consultar) en cualquier momento, con la finalidad que el comercio afiliado pueda tener una información en tiempo real de las transacciones, esto puede servir al portal como una contingencia en caso haya problemas en la página de notificación del comercio afiliado. ^(5.2.6)

5.2.4. Eliminar CIP

PagoEfectivo pone a disposición un servicio web que le permitirá eliminar un CIP (Método eliminar) siempre y cuando este se encuentre en estado pendiente de pago, esto puede servir al portal para casos especiales en que decide que ya no utilizará el CIP como medio de pago. ^(5.2.6)

5.2.5. Actualizar CIP

PagoEfectivo pone a disposición un servicio web que le permitirá actualizar la fecha de expiración de un CIP (Método actualizar) siempre y cuando este se encuentre en estado pendiente de pago, la nueva fecha de expiración debe ser mayor a la fecha actual. ^(5.2.6)

5.2.6. Llaves de encriptación

PagoEfectivo para la seguridad de los mensajes usa un proceso de encriptación basado en RSA lo que lleva al comercio afiliado usar las Llaves de encriptación, que por entidad serán 2 (pública y privada) cabe recalcar que las llaves del ambiente de integración y producción son distintas.

6. ESPECIFICACION DE TÉCNICAS A DETALLE

6.1. Configuración de parámetros:

Los atributos a modificar son los detallados a continuación:

```
// ruta del archivo: WebPages/WEB-INF/web.xml
<web-app>
  <context-param>
    <param-name>PE_SERVER</param-name>
    <param-value>https://pre1a.pagoefectivo.pe/</param-value>
  </context-param>
  <context-param>
    <param-name>PE_MERCHAND_ID</param-name>
    <param-value>SRV</param-value>
  </context-param>
  <context-param>
    <param-name>PE_COMERCIO_CONCEPTO_PAGO</param-name>
    <param-value>PAYCASH</param-value>
  </context-param>
  <context-param>
```

```
<param-name>PE_EMAIL_PORTAL</param-name>
<param-value>admin@paycash.com.pe</param-value>
</context-param>
<context-param>
  <param-name>PE_TIEMPO_EXPIRACION</param-name>
  <param-value>16</param-value>
</context-param>
<context-param>
  <param-name>PE_PUBLICKEY</param-name>
  <param-value>C:\PagoEfectivo2\Claves\SPE_PublicKey.1pz</param-value>
</context-param>
<context-param>
  <param-name>PE_PRIVATEKEY</param-name>
  <param-value>C:\PagoEfectivo2\Claves\SRV_PrivateKey.1pz</param-value>
</context-param></web-app>
```

“PE_SERVER” es la clave para indicar la url del web services que corresponde a las operaciones que ofrece PagoEfectivo (Generación de Solicitud, consulta, etc.).

“PE_MERCHAN_ID” es el código abreviado del servicio (proporcionado por PagoEfectivo).

“PE_COMERCIO_CONCEPTO_PAGO” es el concepto o nombre del portal más el número de transacción.

“PE_EMAIL_PORTAL” es el mail administrador de la tienda.

“PE_TIEMPO_EXPIRACION” es el tiempo de vida de un CIP representado en horas.

“PE_PUBLICKEY” es la ruta física donde se ubicará la clave pública.

“PE_PRIVATEKEY” es la ruta física donde se ubicará la clave privada.

6.2. Configuración de rutas:

En las operaciones se requiere usar las claves de encriptación de la entidad que se está integrando con PagoEfectivo, previamente han sido generadas y entregadas por PagoEfectivo, en los 5 procesos se requerirá usar una o dos claves dependiendo de la operación:

- Generar CIP (clave privada, clave pública).
- Consultar CIP (clave privada, clave pública).
- Eliminar CIP (clave privada, clave pública).
- Actualizar CIP (clave privada, clave pública).
- Desencriptar texto (clave privada).

Se recomienda asignar una ruta que sea segura para estas claves

6.3. Generación del CIP

El método **GenerarCIPMod1** nos permite generar CIP, el método recibe como entrada un objeto **BEWSGenCIPRequestMod1**.

Dentro del objeto que responde el Servicio a esta solicitud se encuentra un campo XML, donde contiene información de la solicitud, uno de las propiedades de la solicitud es el campo token con el cual podrá visualizar el CIP generado según sea la presentación del CIP.

La presentación de CIP puede darse de 2 formas:

- Iframe, Una página embebida en la página del comercio afiliado.

Página = URL + [Token]

Dónde:

URL= <https://cip.pagoefectivo.pe/genpagof.aspx?token=> o
<https://pre1a.pagoefectivo.pe/genpagof.aspx?token=> ,
dependiendo del ambiente

Token=Valor único por CIP.

- Redirect, Sale de la página del comercio afiliado y se presenta en la página de PagoEfectivo.

Página = URL + [Token]

Dónde:

URL= <https://cip.pagoefectivo.pe/genpagof.aspx?token=> o
<https://pre1a.pagoefectivo.pe/genpagof.aspx?token=> , dependiendo
del ambiente

Token=Valor único por CIP.

A continuación, un ejemplo en lenguaje JAVA:

```
//Configuramos nuestros parámetros de WS y rutas de las claves.  
PagoEfectivo.PublicPath = getServletContext().getInitParameter("PublicPath");  
PagoEfectivo.PrivatePath = getServletContext().getInitParameter("PrivatePath");  
PagoEfectivo.UrlPECryptography =  
getServletContext().getInitParameter("UrlPECryptography");  
PagoEfectivo.UrlPEService = getServletContext().getInitParameter("UrlPEService");  
  
//construimos el objeto request: BEWSGenCIPRequestMod1 requestPagoEfectivo = new  
BEWSGenCIPRequestMod1();  
requestPagoEfectivo.setXml(request.getParameter("xml"));  
  
BEWSGenCIPResponseMod1 responsePagoEfectivo =  
PagoEfectivo.GenerarCIP(requestPagoEfectivo); String xml =  
responsePagoEfectivo.getXml();  
//Aquí la forma de recuperar los datos  
switch (Integer.parseInt(responsePagoEfectivo.getEstado())) {
```

```

case 1: //Se generó la solicitud correctamente
    DocumentBuilder db =
    DocumentBuilderFactory.newInstance().newDocumentBuilder();
    Document doc = db.parse(new
    ByteArrayInputStream(xml.getBytes("UTF-8")));
    String numeroCip = "";
    String token = "";
    NodeList nlSolpago =
    doc.getElementsByTagName("ConfirSolPago");
    if (nlSolpago.getLength() > 0) {
        /**
        * numeroCip sera el id con el cual se relacionará la
        * transacción en pagoEfectivo, se usará este para
        * consultar, eliminar o actualizar dicho CIP*
        */
        NodeList nlCIP = ((Element)
        nlSolpago.item(0)).getElementsByTagName("CIP");
        if (nlCIP.getLength() > 0) {
            NodeList nlNumeroCIP = ((Element)
            nlCIP.item(0)).getElementsByTagName("NumeroOrdenPago");
            numeroCip = nlNumeroCIP.item(0).getTextContent();
        }

        //Del mismo modo recuperar el campo Token
        NodeList nlToken = ((Element)
        nlSolpago.item(0)).getElementsByTagName("Token");
        token = nlToken.item(0).getTextContent();
        //guardarEnBD(numeroCip,...);
    }
    //Establecemos la variable CIP para mostrarlo en la web
    request.setAttribute("cip", numeroCip);
    //Aquí para generar el iframe
    request.setAttribute("urlIframe",
    response.encodeRedirectURL(urlWebPagoEfectivo+"/GenPagoIF.aspx?Token="+token));
    break;

```

Descripción de objetos:

➤ BEWSGenCIPRequestMod1

| Nombre | Tipo de dato | Mandatorio | Descripción y Estructura (Tipo de Dato o Estructura) |
|----------------|--------------|------------|--|
| CodServ | Cadena (3) | SI | Código del comercio afiliado (asignado por PagoEfectivo), consta de 3 caracteres. Ejemplo: SRV |
| Firma | Cadena | SI | Es la trama firmada con la clave privada (asignado por PagoEfectivo) a través del WebService WSCrypto. |
| (*)XML | Cadena | SI | Es la trama en formato XML que contiene todos los datos relevantes de la operación y debe ser encriptado a través del WebService WSCrypto. |

(*)XML

| Nombre | Tipo de dato | Mandatorio | Descripción y Estructura (Tipo de Dato o Estructura) |
|-------------------------|---------------|------------|--|
| IdMoneda | Cadena (1) | SI | Código de la moneda: 1: Soles y 2: Dólares |
| Total | Decimal(18,2) | SI | Monto total del pago. Debe ser mayor a 0. Cadena 18 enteros, 2 decimales y separados por un punto. |
| MetodosPago | Cadena (1) | SI | Por defecto el valor 1 |
| CodServicio | Cadena (3) | SI | Código del comercio (asignado por PagoEfectivo). Ejemplo: SRV |
| Codtransaccion | Cadena (12) | SI | Código de Referencia de la transacción (nro pedido del comercio), debe ser único. |
| EEmailComercio | Cadena (30) | NO | Correo electrónico del comercio afiliado. A esta cuenta le estará llegando una copia del correo de generación, recordatorio y pago de CIP. |
| FechaAExpirar | Cadena (19) | SI | La fecha y hora en el que el CIP expirará. Formato: DD/MM/AAAA HH:MM:SS (24 horas) |
| UsuarioID | Cadena (12) | NO | Código del usuario del comercio. |
| DataAdicional | Cadena (30) | NO | Información de referencia. |
| UsuarioNombre | Cadena (12) | SI | Nombre del usuario de referencia de la generación del CIP. No se aceptan valores extraños (tilde, ñ, entre otros) |
| UsuarioApellidos | Cadena (12) | SI | Apellidos del usuario de referencia de la generación del CIP. No se aceptan valores extraños (tilde, ñ, entre otros) |
| UsuarioLocalidad | Cadena (12) | NO | Nombre de la Localidad del usuario. |
| UsuarioProvincia | Cadena (12) | NO | Nombre de la Provincia del usuario. |
| UsuarioPais | Cadena (12) | NO | Nombre del País del usuario. |
| UsuarioAlias | Cadena (8) | NO | Alias del usuario. |
| UsuarioTipoDoc | Cadena (3) | NO | Tipo de Documento: DNI, LE, RUC, etc. |
| UsuarioNumeroDoc | Cadena (11) | NO | Número del Documento. |
| UsuarioEmail | Cadena (30) | SI | Email del usuario. |
| ConceptoPago | Cadena (16) | SI | Nombre del comercio. No se aceptan valores extraños (tilde, ñ, entre otros) |
| Celular | Cadena (15) | SI | Número telefónico del usuario. |
| CodPais | Cadena (3) | SI | Código Del País. |
| (**)Detalles | XML | SI | Contiene los detalles del pago asociado, especificado más abajo. |

(**)Detalles

| Nombre | Tipo de dato | Mandatorio | Descripción y Estructura (Tipo de Dato o Estructura) |
|---------------------|---------------|------------|---|
| ConceptoPago | Cadena (16) | SI | Nombre del comercio. |
| Importe | Decimal(18,2) | SI | Monto total del pago. Cadena 18 enteros, 2 decimales y separados por un punto. |

Ver ejemplos:

| Nombre del archivo | Archivo |
|------------------------------------|--|
| Generación_Request_Encriptado |  Generación_Reques t_Encriptado.txt |
| Generación_Request_XML_descryptado |  Generación_Reques t_XML_descrypta |

➤ **BEWSGenCIPResponseMod1**

| Nombre | Tipo de dato | Descripción y Estructura (Tipo de Dato o Estructura) |
|----------------|--------------|---|
| Estado | Cadena (1) | Representa el resultado de la operación: “1”: Se llegó a realizar la solicitud “0”: No se llegó a realizar la solicitud “-1”: Se dio una excepción no controlada |
| Mensaje | Cadena | Contiene el mensaje del resultado de la operación: Estado “1”: Se ha Generado el CIP: 00000006239418 . ⁽¹⁾ Estado “0”: -El Código de Identificación de Transacción ya se ha generado anteriormente para esta operación. Código Transacción: 001000 -El monto total del CIP no concuerda con la suma del importe de los detalles. Estado “-1”: -Ha ocurrido un Error no Manejado. -Error al obtener xml según contrato. <small>(1) Solo en el estado “1” el XML contiene dato.</small> |
| (*)Xml | Cadena | Texto encriptado con la estructura de un Xml que retorna información adicional asociado a la operación. |

(*)XML

| Nombre | Tipo de dato | Descripción y Estructura (Tipo de Dato o Estructura) |
|---------------------|--------------|---|
| idResSolPago | Cadena (14) | Id de la solicitud de pago. |
| CodTrans | Cadena (12) | Código de Referencia de la transacción (Nro pedido del comercio), equivalente a Codtransaccion. |
| Token | Cadena | Código Único por CIP, generado para la operación. |
| MetodoPago | Cadena (1) | Por defecto el valor es 1 |
| Codigo | Cadena (3) | Código del comercio, equivalente a CodServicio. Ejemplo: SRV |
| Fecha | DateTime | Campo la fecha en que realizo la operación, el formato es AAAA-MM-DD HH:MM:SS (24 horas) |

| | | |
|------------------|---------|---|
| Estado | Integer | Estado en que se encuentra la Operación: 592: Pendiente de pago 593: Pagado 595: Expirado (sobrepaso el tiempo de limite). |
| ParamsURL | Cadena | Por defecto el valor es null. |
| (**)CIP | XML | Si la solicitud genero un CIP se retornará los datos de dicho CIP en un formato XML. |

()CIP**

| Nombre | Tipo de dato | Descripción y Estructura (Tipo de Dato o Estructura) |
|-------------------------|---------------------|--|
| IdOrdenPago | Cadena (14) | Identificador único del CIP |
| IdCliente | Integer | Identificador único del Cliente en Pago Efectivo |
| IdEstado | Integer | Indica el estado de la operación: 21: Expirado 22: Pendiente de pago 23: Pagado 25: Eliminado |
| IdServicio | Integer | Indicador único del servicio en PagoEfectivo. |
| IdMoneda | Integer | Código de la moneda: 1: Soles 2: Dólares |
| NumeroOrdenPago | Cadena (14) | Número de CIP Generado. |
| OrderIdComercio | Cadena (12) | Código de Referencia de la transacción (nro pedido del comercio), equivalente a Codtransaccion y CodTrans. |
| Total | Decimal (18,2) | Monto total del pago. |
| MerchantID | Cadena (3) | Código del comercio, equivalente a CodServicio y codigo. Ejemplo: SRV |
| FechaEmision | Cadena | Fecha en que realizo la generación del CIP, el formato es AAAA-MM-DD HH:MM:SS (24 horas) |
| FechaCancelado | Cadena | Fecha en la que se pagó el CIP. Por defecto valor vacío. |
| FechaConciliado | Cadena | Fecha en la que se concilió el CIP. Por defecto valor vacío. |
| FechaAnulado | Cadena | Fecha en la que se anuló el CIP. Por defecto valor vacío. |
| FechaExpirado | Cadena | Fecha en la que se expiró el CIP. Por defecto valor vacío. |
| FechaEliminado | Cadena | Fecha en la que se eliminó el CIP. Por defecto valor vacío. |
| TiempoExpiracion | Cadena (30) | Tiempo de vigencia en horas del código CIP generado. |
| EstaConciliado | Cadena | Campo que indica si está conciliado: Por defecto valor vacío en la generación de CIP. |
| MailComercio | Cadena | Correo electrónico del comercio afiliado. A esta cuenta le estará llegando una copia del correo de generación, recordatorio y pago de CIP. |
| UsuarioID | Cadena | Código del usuario del comercio afiliado. |
| DataAdicional | Cadena | Información de referencia |
| UsuarioNombre | Cadena | Nombre de usuario de referencia de la generación del CIP |
| UsuarioApellidos | Cadena | Apellidos del usuario de referencia de la generación del CIP |
| UsuarioAlias | Cadena | Alias del usuario de referencia de la generación del CIP |
| UsuarioEmail | Cadena | Email del usuario de referencia de la generación del CIP |
| UsuarioDomicilio | Cadena | Descripción del domicilio del usuario. Por defecto valor vacío. |

| | | |
|-------------------------|--------|---|
| UsuarioLocalidad | Cadena | Descripción de la localidad del usuario |
| UsuarioProvincia | Cadena | Descripción de la provincia del usuario |
| UsuarioPaís | Cadena | Descripción del país del usuario |
| (**)Detalles | XML | Contiene los detalles del pago asociado |

()Detalles:**

| Nombre | Tipo de dato | Descripción y Estructura (Tipo de Dato o Estructura) |
|---------------------------|---------------------|---|
| IdDetalleOrdenPago | Cadena | Id del detalle de pago. |
| ConceptoPago | Cadena | Descripción del Concepto de Pago |
| Importe | Decimal (18,2) | Monto del pago del detalle. |
| Tipo_Origen | Cadena | Por defecto el valor es null. |
| Cod_Origen | Cadena | Por defecto el valor es null. |
| Campo1 | Cadena | Por defecto el valor es null. |
| Campo2 | Cadena | Por defecto el valor es null. |
| Campo3 | Cadena | Por defecto el valor es null. |

Ver ejemplos:

| Nombre del archivo | Archivo |
|-------------------------------------|---|
| Generación_Response_Encryptado |  Generación_Response_Encryptado.txt |
| Generación_Response_XML_descryptado |  Generación_Response_XML_descrip |

6.4.Request URLNotificacion (notificación de estados del CIP)

La notificación de PagoEfectivo hacia el comercio afiliado es por medio de una petición web (WebRequest), hacia una ruta (UrlNotificación) que haya establecido el comercio afiliado.

Esta notificación se da en los siguientes casos:

- Pago de CIP
- Reversa de Pago (Extorno) de CIP
- Expiración de CIP

La trama de notificación está conformada por:

[data]+[versión]

Donde:

| Parámetro | Tipo | Descripción |
|-----------|------------|--|
| (*)Data | Cadena | Data, Contiene toda la información de la transacción. Parámetro a descryptar usando la función WSCrypto.DecryptText(), estructura XML donde encontrara el Estado final de la transacción. |
| Versión | Cadena (1) | Valor: 2, Modalidad utilizada. |

(*)Data

| Nombre | Tipo de dato | Descripción y Estructura (Tipo de Dato o Estructura) |
|---------------------|--------------|---|
| idResSolPago | Cadena (14) | Id de la solicitud de pago. |
| CodTrans | Cadena (12) | Código de Referencia de la transacción (Nro pedido del comercio), equivalente a Codtransaccion. |
| Token | Cadena | Código Único por CIP, generado para la operación. |
| MetodoPago | Cadena (1) | Por defecto el valor es 1 |
| Codigo | Cadena (3) | Código del comercio, equivalente a CodServicio. Ejemplo: SRV |
| Fecha | DateTime | Campo la fecha en que realizo la operación, el formato es AAAA-MM-DD HH:MM:SS (24 horas) |
| Estado | Integer | Estado en que se encuentra la Operación: 592: Pendiente de pago 593: Pagado 595: Expirado (sobrepaso el tiempo de limite). |
| ParamsURL | Cadena | Por defecto el valor es null. |
| (**)CIP | XML | Trama XML con datos de la transacción. |

(**)CIP

| Nombre | Tipo de dato | Descripción y Estructura (Tipo de Dato o Estructura) |
|------------------------|----------------|--|
| IdOrdenPago | Cadena (14) | Identificador único del CIP |
| IdCliente | Integer | Identificador único del Cliente en PagoEfectivo |
| IdEstado | Integer | Indica el estado de la operación: 21: Expirado 22: Pendiente de pago 23: Pagado 25: Eliminado |
| IdServicio | Integer | Indicador único del servicio en PagoEfectivo. |
| IdMoneda | Integer | Código de la moneda: 1: Soles 2: Dólares |
| NumeroOrdenPago | Cadena (14) | Número de CIP Generado. |
| OrderIdComercio | Cadena (12) | Código de Referencia de la transacción (nro pedido del comercio), equivalente a Codtransaccion y CodTrans. |
| Total | Decimal (18,2) | Monto total del pago. |

| | | |
|-------------------------|-------------|--|
| MerchantID | Cadena (3) | Código del comercio, equivalente a CodServicio y código. Ejemplo: SRV |
| FechaEmision | Cadena | Fecha en la que se emitió el CIP. Formato AAAA-MM-DD HH:MM:SS (24 horas) |
| FechaCancelado | Cadena | Fecha en la que se pagó el CIP. Siempre y cuando IdEstado=23 y Estado=593, de lo contrario el valor será vacío. Formato AAAA-MM-DD HH:MM:SS (24 horas) |
| FechaConciliado | Cadena | Fecha en la que se concilió el CIP. Por defecto valor vacío. |
| FechaAnulado | Cadena | Fecha en la que se anuló el CIP. Siempre y cuando IdEstado=22 o IdEstado=25 y Estado=592, de lo contrario el valor será vacío. Formato AAAA-MM-DD HH:MM:SS (24 horas) |
| FechaExpirado | Cadena | Fecha en la que se expiró el CIP. Siempre y cuando IdEstado=21 y Estado=595, de lo contrario el valor será vacío. Formato AAAA-MM-DD HH:MM:SS (24 horas) |
| FechaEliminado | Cadena | Fecha en la que se eliminó el CIP. Por defecto valor vacío. |
| TiempoExpiracion | Cadena (30) | Tiempo de vigencia en horas del código CIP generado. |
| EstaConciliado | Cadena | Campo que indica si está conciliado: Por defecto valor vacío en la generación de CIP. |
| MailComercio | Cadena | Correo electrónico del comercio afiliado. A esta cuenta le estará llegando una copia del correo de generación, recordatorio y pago de CIP. |
| UsuarioID | Cadena | Código del usuario del comercio afiliado. |
| DataAdicional | Cadena | Información de referencia |
| UsuarioNombre | Cadena | Nombre de usuario de referencia de la generación del CIP |
| UsuarioApellidos | Cadena | Apellidos del usuario de referencia de la generación del CIP |
| UsuarioAlias | Cadena | Alias del usuario de referencia de la generación del CIP |
| UsuarioEmail | Cadena | Email del usuario de referencia de la generación del CIP |
| UsuarioDomicilio | Cadena | Descripción del domicilio del usuario. Por defecto valor vacío. |
| UsuarioLocalidad | Cadena | Descripción de la localidad del usuario |
| UsuarioProvincia | Cadena | Descripción de la provincia del usuario |
| UsuarioPais | Cadena | Descripción del país del usuario |
| (***)Detalles | XML | Contiene los detalles del pago asociado |

(*)Detalles:**

| Nombre | Tipo de dato | Descripción y Estructura (Tipo de Dato o Estructura) |
|---------------------------|---------------------|---|
| IdDetalleOrdenPago | Cadena | Id del detalle de pago. |
| ConceptoPago | Cadena | Descripción del Concepto de Pago |
| Importe | Decimal (18,2) | Monto del pago del detalle. |
| Tipo_Origen | Cadena | Por defecto el valor es vacío. |
| Cod_Origen | Cadena | Por defecto el valor es vacío. |
| Campo1 | Cadena | Por defecto el valor es vacío. |
| Campo2 | Cadena | Por defecto el valor es vacío. |
| Campo3 | Cadena | Por defecto el valor es vacío. |

Ver ejemplos:

| Nombre del archivo | Archivo |
|---|---|
| Notificación_Pago_Request_Encriptado |  Notificación_Pago_Request_Encriptado |
| Notificación_Pago_Request_Descriptado |  Notificación_Pago_Request_Descencrip |
| Notificación_Extorno_Request_Encriptado |  Notificación_Extorno_Request_Encrptar |
| Notificación_Extorno_Request_Descriptado |  Notificación_Extorno_Request_Descenci |
| Notificación_Expirado_Request_Encriptado |  Notificación_Expirado_Request_Encrpt |
| Notificación_Expirado_Request_Descriptado |  Notificación_Expirado_Request_Descen |

La plataforma de PagoEfectivo espera que el portal responda estado 200 en la recepción de la notificación. De acontecer un error en la recepción, la plataforma intentará realizar un segundo intento.

6.5.Consultar CIP

Nos permite consultar uno o más CIPs (Máximo 10), el método recibe como entrada un objeto **BEWSConsultarCIPRequestMod1**.

El resultado de la operación devuelve una entidad del tipo **BEWSConsultarCIPResponseMod1**.

A continuación, un ejemplo en lenguaje JAVA:

```
//Para este caso solo necesitamos asignar la url del WS de PagoEfectivo
PagoEfectivo.PublicPath = getServletContext().getInitParameter("PublicPath");
PagoEfectivo.PrivatePath = getServletContext().getInitParameter("PrivatePath");
PagoEfectivo.UrlPECryptography =
getServletContext().getInitParameter("UrlPECryptography");
PagoEfectivo.UrlPEService = getServletContext().getInitParameter("UrlPEService");
```

```

BEWSConsultarCIPRequestMod1 requestPagoEfectivo = new
BEWSConsultarCIPRequestMod1();

//El capi puede ser asignado directamente del web.xml
requestPagoEfectivo.setCodServ(getServletContext().getInitParameter("CAPI"));

//Los cips asignados manualmente (más de 1 enviar separados por coma)
requestPagoEfectivo.setCIPS(request.getParameter("cips"));

//Ejecutamos la Consulta de la clase PagoEfectivo(API)
BEWSConsultarCIPResponseMod1 responsePagoEfectivo =
PagoEfectivo.ConsultarCIP(requestPagoEfectivo); String xml =
responsePagoEfectivo.getXML();

```

Descripción de objetos

➤ BEWSConsultarCIPRequestMod1

| Nombre | Tipo de dato | Mandatorio | Descripción y Estructura (Tipo de Dato o Estructura) |
|--------------------|--------------|------------|--|
| CodServ | Cadena (3) | SI | Código del comercio afiliado (asignado por PagoEfectivo), consta de 3 caracteres. Ejemplo: SRV |
| Firma | Cadena | SI | Es la trama firmada con la clave privada (asignado por PagoEfectivo) a través de un Webservice WSCrypto. |
| CIPS | Cadena | SI | Es el código de identificador de pago generado, el cual se desea consultar. |
| InfoRequest | Cadena | NO | Enviar valor vacío. |

Ver ejemplos:

| Nombre del archivo | Archivo |
|----------------------------------|---|
| Consultar_Request_Encriptado |  Consultar_Request_Encriptado.txt |
| Consultar_Request_Descencriptado |  Consultar_Request_Descencriptado.txt |

➤ BEWSConsultarCIPResponseMod1

| Nombre | Tipo de dato | Descripción y Estructura (Tipo de Dato o Estructura) |
|----------------|--------------|---|
| Estado | Cadena | Representa el resultado de la operación: “-1” : Se dio un error no controlado. “0” : No existe la solicitud de pago. “1” : Se encontró la Solicitud de pago. |
| Mensaje | Cadena | Contiene el mensaje resultado de haber realizado la operación: En el caso del Estado “-1”: - “Ha ocurrido un Error no Manejado” En el caso del Estado “0”: - “No se encontraron CIPs asociados” |

| | | |
|---------------------|--------|--|
| | | En el caso del Estado "1": - "La consulta de los CIPs se realizó con éxito" |
| InfoResponse | Cadena | Valor vacío. |
| (*)XML | Cadena | Es la trama en formato XML que contiene todos los datos relevantes de la operación. Esta información viaja encriptada. |

(*)XML

| Nombre | Tipo de dato | Descripción y Estructura (Tipo de Dato o Estructura) |
|---------------------|--------------|---|
| idResSolPago | Cadena (14) | Id de la solicitud de pago. |
| CodTrans | Cadena (12) | Código de Referencia de la transacción (Nro pedido del comercio), equivalente a Codtransaccion. |
| Token | Cadena | Código Único por CIP, generado para la operación. |
| MetodoPago | Cadena (1) | Por defecto el valor es 1 |
| Codigo | Cadena (3) | Código del comercio, equivalente a CodServicio. Ejemplo: SRV |
| Fecha | DateTime | Campo la fecha en que realizo la operación, el formato es AAAA-MM-DD HH:MM:SS (24 horas) |
| Estado | Integer | Estado en que se encuentra la Operación: 592: Pendiente de pago 593: Pagado 595: Expirado (sobrepaso el tiempo de limite). |
| ParamsURL | Cadena | Por defecto el valor es vacío. |
| (**)CIP | XML | Trama XML con datos de la transacción. |

(**)CIP

| Nombre | Tipo de dato | Descripción y Estructura (Tipo de Dato o Estructura) |
|------------------------|----------------|---|
| IdOrdenPago | Cadena (14) | Identificador único del CIP |
| IdCliente | Integer | Identificador único del Cliente en PagoEfectivo |
| IdEstado | Integer | Indica el estado de la operación: 21: Expirado 22: Pendiente de pago 23: Pagado 25: Eliminado |
| IdServicio | Integer | Indicador único del servicio en PagoEfectivo. |
| IdMoneda | Integer | Código de la moneda: 1: Soles 2: Dólares |
| NumeroOrdenPago | Cadena (14) | Número de CIP Generado. |
| OrderIdComercio | Cadena (12) | Código de Referencia de la transacción (nro pedido del comercio), equivalente a Codtransaccion y CodTrans. |
| Total | Decimal (18,2) | Monto total del pago. |
| MerchantID | Cadena (3) | Código del comercio, equivalente a CodServicio y codigo. Ejemplo: SRV |
| FechaEmision | Cadena | Fecha en la que se emitió el CIP. Formato AAAA-MM-DD HH:MM:SS (24 horas) |
| FechaCancelado | Cadena | Fecha en la que se pagó el CIP. Siempre y cuando IdEstado=23 y Estado=593, de lo contrario el valor será vacío. |

| | | |
|-------------------------|-------------|--|
| | | Formato AAAA-MM-DD HH:MM:SS (24 horas) |
| FechaConciliado | Cadena | Fecha en la que se concilió el CIP. Por defecto valor vacío. |
| FechaAnulado | Cadena | Fecha en la que se anuló el CIP. Siempre y cuando IdEstado=22 o IdEstado=25 y Estado=592, de lo contrario el valor será vacío. Formato AAAA-MM-DD HH:MM:SS (24 horas) |
| FechaExpirado | Cadena | Fecha en la que se expiró el CIP. Siempre y cuando IdEstado=21 y Estado=595, de lo contrario el valor será vacío. Formato AAAA-MM-DD HH:MM:SS (24 horas) |
| FechaEliminado | Cadena | Fecha en la que se eliminó el CIP. Por defecto valor vacío. |
| TiempoExpiracion | Cadena (30) | Tiempo de vigencia en horas del código CIP generado. |
| EstaConciliado | Cadena | Campo que indica si está conciliado. Por defecto valor vacío en la generación de CIP. |
| MailComercio | Cadena | Correo electrónico del comercio afiliado. A esta cuenta le estará llegando una copia del correo de generación, recordatorio y pago de CIP. |
| UsuarioID | Cadena | Código del usuario del comercio afiliado. |
| DataAdicional | Cadena | Información de referencia |
| UsuarioNombre | Cadena | Nombre de usuario de referencia de la generación del CIP |
| UsuarioApellidos | Cadena | Apellidos del usuario de referencia de la generación del CIP |
| UsuarioAlias | Cadena | Alias del usuario de referencia de la generación del CIP |
| UsuarioEmail | Cadena | Email del usuario de referencia de la generación del CIP |
| UsuarioDomicilio | Cadena | Descripción del domicilio del usuario. Por defecto valor vacío. |
| UsuarioLocalidad | Cadena | Descripción de la localidad del usuario |
| UsuarioProvincia | Cadena | Descripción de la provincia del usuario |
| UsuarioPais | Cadena | Descripción del país del usuario |
| (***)Detalles | XML | Contiene los detalles del pago asociado |

(*)Detalles:**

| Nombre | Tipo de dato | Descripción y Estructura (Tipo de Dato o Estructura) |
|---------------------------|----------------|---|
| IdDetalleOrdenPago | Cadena | Id del detalle de pago. |
| ConceptoPago | Cadena | Descripción del Concepto de Pago |
| Importe | Decimal (18,2) | Monto del pago del detalle. |
| Tipo_Origen | Cadena | Por defecto el valor es vacío. |
| Cod_Origen | Cadena | Por defecto el valor es vacío. |
| Campo1 | Cadena | Por defecto el valor es vacío. |
| Campo2 | Cadena | Por defecto el valor es vacío. |
| Campo3 | Cadena | Por defecto el valor es vacío. |

Ver ejemplos:

| Nombre del archivo | Archivo |
|--------------------|---------|
|--------------------|---------|

| | |
|---|---|
| Consulta_Response_Pago_Encriptado |  Consulta_Response_Pago_Encriptado.tx |
| Consulta_Response_Pago_Descryptado |  Consulta_Response_Pago_Desccripta |
| Consulta_Response_PendientePago_Encriptado |  Consulta_Response_PendientePago_Eni |
| Consulta_Response_PendientePago_Descryptado |  Consulta_Response_PendientePago_De |
| Consulta_Response_Expirado_Encriptado |  Consulta_Response_Expirado_Encriptad |
| Consulta_Response_Expirado_Descryptado |  Consulta_Response_Expirado_Desccripti |
| Consulta_Response_Eliminado_Encriptado |  Consulta_Response_Eliminado_Encrypta |
| Consulta_Response_Eliminado_Descryptado |  Consulta_Response_Eliminado_Descenc |

6.6. Eliminar CIP

Nos permite anular un CIP, el método recibe como entrada un objeto **BEWSElimCIPRequestMod1**. El resultado de la operación devuelve una entidad del tipo **BEWSElimCIPResponseMod1**.

A continuación, un ejemplo en lenguaje JAVA:

```
//Para este caso necesitamos asignar las url de los WS de PagoEfectivo
PagoEfectivo.PublicPath = getServletContext().getInitParameter("PublicPath");
PagoEfectivo.PrivatePath = getServletContext().getInitParameter("PrivatePath");
PagoEfectivo.UrlPEcriptography =
getServletContext().getInitParameter("UrlPEcriptography");
PagoEfectivo.UrlPEService = getServletContext().getInitParameter("UrlPEService");
BEWSElimCIPRequestMod1 requestPagoEfectivo = new BEWSElimCIPRequestMod1();

//El capi puede ser asignado directamente del web.xml
requestPagoEfectivo.setCodServ(getServletContext().getInitParameter("CAPI"));
```

```
//El numero de cip asignado manualmente
requestPagoEfectivo.setCIP(request.getParameter("cip"));

//Ejecutamos la Consulta de la clase PagoEfectivo(API) BEWSElimCIPResponseMod1
responsePagoEfectivo = PagoEfectivo.EliminarCIP(requestPagoEfectivo);
```

Descripción de objetos

➤ BEWSElimCIPRequestMod1

| Nombre | Tipo de dato | Mandatorio | Descripción y Estructura (Tipo de Dato o Estructura) |
|---------------------|--------------|------------|--|
| CodServ | Cadena | SI | Código del comercio (asignado por PagoEfectivo), consta de 3 caracteres. Ejemplo: SRV |
| Firma | Cadena | SI | Es la trama firmada con la clave privada (asignado por PagoEfectivo) a través de un WebService WSCrypto. |
| CIP | Cadena | SI | Es el código identificador de pago a eliminar. |
| InfoResponse | Cadena | NO | Valor vacío. |

Ver ejemplos:

| Nombre del archivo | Archivo |
|---------------------------------|--|
| Eliminar_Request_Encriptado |  Eliminar_Request_Encriptado.txt |
| Eliminar_Request_Descencriptado |  Eliminar_Request_Descencriptado.txt |

➤ BEWSElimCIPResponseMod1

| Nombre | Tipo de dato | Descripción y Estructura (Tipo de Dato o Estructura) |
|---------------------|--------------|--|
| Estado | Cadena | Representa el resultado de la operación: "1": Se llegó a realizar la eliminación. "0": No se llegó a realizar la eliminación del CIP. "-1": Ocurrió un error no controlado. |
| Mensaje | Cadena | Contiene el mensaje resultado de haber realizado la operación: En el caso del Estado "1": - "El CIP 6239418 se ha eliminado correctamente." En el caso del Estado "0": - "La clave del servicio es incorrecto." En el caso del Estado "-1": - "Ha ocurrido un Error no Manejado: El estado del CIP no se ha identificado." |
| InfoResponse | Cadena | Valor vacío. |

Ver ejemplos:

| Nombre del archivo | Archivo |
|--------------------|--|
| Eliminar_Response |  Eliminar_Response.txt |

6.7. Actualizar CIP

Nos permite actualizar la fecha de expiración de un CIP, el método recibe como entrada un objeto **BEWSActualizaCIPRequestMod1**. El resultado de la operación devuelve una entidad del tipo **BEWSActualizaCIPResponseMod1**.

A continuación, un ejemplo en lenguaje JAVA:

```
//Para este caso solo necesitamos asignar la url del WS de PagoEfectivo
PagoEfectivo.PublicPath =
getServletContext().getInitParameter("PublicPath");
PagoEfectivo.PrivatePath =
getServletContext().getInitParameter("PrivatePath");
PagoEfectivo.UrlPEcriptography =
getServletContext().getInitParameter("UrlPEcriptography");
PagoEfectivo.UrlPEService =
getServletContext().getInitParameter("UrlPEService");

BEWSActualizaCIPRequestMod1 requestPagoEfectivo = new
BEWSActualizaCIPRequestMod1();
//El capi y cclave pueden ser asignados directamente del web.xml
requestPagoEfectivo.setCodServ(getServletContext().getInitParameter("CAPI"
));

//El cip que se actualizar asignado manualmente
requestPagoEfectivo.setCIP(request.getParameter("cip"));

//La nueva fecha de expiración, formatiamos a GregorianCalendar. DateFormat
formatter = new SimpleDateFormat("dd/MM/yyyy hh:mm:ss"); Date Fecha =
(Date) formatter.parse(request.getParameter("fecha")); GregorianCalendar
gregorianCalendar = new GregorianCalendar();
gregorianCalendar.setTime(Fecha); XMLGregorianCalendar
gregorianCalendarXml = DatatypeFactory.newInstance().newXMLGregorianCale
ndar(gregorianCalendar);
requestPagoEfectivo.setFechaExpira(gregorianCalendarXml);

//Ejecutamos la actualización de cip de la clase PagoEfectivo(API)
BEWSActualizaCIPResponseMod1 responsePagoEfectivo =
PagoEfectivo.ActualizarCIP(requestPagoEf ectivo);
```

Descripción de objetos

➤ **BEWSActualizaCIPRequestMod1:**

| Nombre | Tipo de dato | Mandatorio | Descripción y Estructura (Tipo de Dato o Estructura) |
|----------------|--------------|------------|---|
| CodServ | Cadena | SI | Código del comercio (asignado por PagoEfectivo), consta de 3 caracteres. Ejemplo: PRU |

| | | | |
|---------------------|--------|----|---|
| Firma | Cadena | SI | Es la trama firmada con la clave privada (asignado por PagoEfectivo) a través de un Webservice WSCrypto. |
| CIP | Cadena | SI | Es el código identificador de pago a actualizar. |
| InfoResponse | Cadena | SI | Es el código identificador de pago a actualizar. |
| FechaExpira | Date | SI | Es la nueva fecha de expiración del CIP, debe ser mayor a la fecha actual y no debe excederse a más de 10 años. Formato: DD/MM/AAAA HH:MM:SS p.m. o a.m. |

Ver ejemplos:

| Nombre del archivo | Archivo |
|-----------------------------------|--|
| Actualizar_Request_Encriptado |  Actualizar_Request_Encriptado.txt |
| Actualizar_Request_Descencriptado |  Actualizar_Request_Descencriptado.txt |

➤ **BEWSActualizaCIPResponseMod1:**

| Nombre | Tipo de dato | Mandatorio | Descripción y Estructura (Tipo de Dato o Estructura) |
|---------------------|--------------|------------|---|
| Estado | Cadena | SI | Representa el resultado de la operación: “1”: Se realizó la actualización. “0”: No se llegó a realizar la actualización del CIP. “-1”: Se dio un error no controlado. |
| Mensaje | Cadena | SI | Contiene el resultado de haber realizado la operación: En el caso del Estado “1”: - “La fecha de expiración del CIP 6239418 se ha modificada correctamente.” En el caso del Estado “0”: - “CIP no existe.” En el caso del Estado “-1”: - “Ha ocurrido un Error no Manejado: El estado del CIP no se ha identificado.” |
| InfoResponse | Cadena | NO | Valor vacío. |

Ver ejemplos:

| Nombre del archivo | Archivo |
|---------------------|--|
| Actualizar_Response |  Actualizar_Response.txt |

6.8. Encriptación de recuperar tramas

El servicio para solicitar la generación de un CIP también necesita primero encriptar la trama, que se enviará con la llave pública que PagoEfectivo le proporciona:

A continuación, un ejemplo en lenguaje JAVA:

```
// ruta del archivo: src/java/pagoEfecto/api/PagoEfectivo.java
//Convertimos a bytes los archivos que contienen las claves (proporcionado
por PagoEfectivo)
BepaymentResponse paymentResponse = new BepaymentResponse();
try {
    byte[] dataPublic = Files.readAllBytes(Paths.get(PublicPath));
    byte[] dataPrivate =
Files.readAllBytes(Paths.get(PrivatePath));
    //Firmamos la trama xml antes de encriptarla y el resultado lo
asiganamos en el campo firma
    solicitud.setFirma(wscryptoSoap.signer(solicitud.getXml(),
dataPrivate));
    //Encriptamos la trama xml y la asignamos al mismo campo xml
solicitud.setXml(wscryptoSoap.encryptText(solicitud.getXml(),
dataPublic));
```

| Nombre | Tipo Dato | Mandatorio | Descripción y Estructura (Tipo de Dato o Estructura) |
|------------------|-----------|------------|---|
| plainText | Cadena | SI | Trama a firmar. |
| publicKey | Binario | SI | Llave pública que brinda PagoEfectivo tras su afiliación. |

➤ EncryptTextResponse

| Nombre | Tipo Dato | Descripción y Estructura (Tipo de Dato o Estructura) |
|--------------------------|-----------|---|
| EncryptTextResult | Cadena | Trama XML Encriptada |

Nota: El texto encriptado solo podrá ser descryptado por PagoEfectivo.

6.9.Desencriptación de tramas

PagoEfectivo cuando responde a la solicitud de pago o cuando notifica al servicio sobre algún cambio de estado de CIP envía o retorna una trama encriptada. Para esto el comercio afiliado debe invocar la trama con el siguiente contrato:

A continuación, un ejemplo en lenguaje JAVA:

```
//// ruta del archivo: src/java/pagoEfecto/api/PagoEfectivo.java
String tramaEncriptada = "",tramaDesencriptada = "";
request.getParameter("version").toString().compareTo("2")==0) if(

//nos aseguramos que venga la version que usamos actualmente /**el
parametro data trae la información encriptada de la transacción * en
un XML como String **/ tramaEncriptada = request.getParameter("data");
PagoEfectivo.PublicPath =
getServletContext().getInitParameter("pathKeyPublic");
PagoEfectivo.PrivatePath =
```

```

getServletContext().getInitParameter("pathKeyPrivate"); //
Descriptamos la trama tramaDescriptada =
PagoEfectivo.DescriptarTexto(tramaEncriptada); }

//Nos preparamos para leer el XML ( DocumentBuilder db =
DocumentBuilderFactory.newInstance().newDocumentBuilder(); Document doc =
db.parse(new ByteArrayInputStream(tramaDescriptada.getBytes("UTF-8")));

String cip=0; int estado=0;

NodeList nlSolpago = doc.getElementsByTagName("ConfirSolPago");
if(nlSolpago.getLength()>0) { /** Obtenemos el idResSolpago de la trama
XML enviada, este id es el * identificador de la transacción a
actualizar**/
NodeList nlidResSolPago =
((Element)nlSolpago.item(0)).getElementsByTagName("NumeroOrdenPa go ");
cip = nlidResSolPago.item(0).getTextContent(); /** Obtenemos el
estado de la transacción**/
NodeList nlEstado =
((Element)nlSolpago.item(0)).getElementsByTagName("Estado"); String
valueEstado = nlEstado.item(0).getTextContent(); estado =
Integer.parseInt(valueEstado); //Dependiendo del estado se
procedera a actualizar, los estados pueden ser: switch(estado){
case 592: //Actualizar la transacción a Generada //(esto pasa
cuando extorna, de pagado 593 pasa a generado 592)
//ExtornarTransaccion(cip); break; case 593: //Se
realizó el pago de la transacción //PagarTransaccion(cip)
break; case 595: //Expiró la transacción despues de generada como
CIP //ExpirarTransaccion(cip) break; }
}

```

➤ DecryptText

| Nombre | Tipo Dato | Mandatorio | Descripción y Estructura (Tipo de Dato o Estructura) |
|--------------------|-----------|------------|---|
| EncryptText | Cadena | SI | Trama a descriptar. |
| PrivateKey | Binario | SI | Llave privada que brinda PagoEfectivo tras su afiliación. |

La respuesta para esta solicitud también es una cadena con la trama descriptada.

➤ DecryptTextResponse

| Nombre | Tipo Dato | Descripción y Estructura (Tipo de Dato o Estructura) |
|--------------------------|-----------|---|
| DecryptTextResult | Cadena | Trama descriptada |

Nota: La trama encriptada enviada por PagoEfectivo solo puede ser descriptada con la llave privada del portal registrado correspondiente.

