



**Manual de integración con el TPV Virtual para
comercios con conexión por Redirección**

SERVICIO TECNICO TPV VIRTUAL

Teléfono: 902 365 650 opción 2

tpvvirtual@bancsabadell.com

Atención especial para migración SHA1 a SHA2: Lunes a
Viernes de 10h a 19h

ÍNDICE DE CONTENIDO

1. Introducción	4
1.1 Objetivo	4
1.2 Definiciones, siglas y abreviaturas	¡Error! Marcador no definido.
1.3 Referencias	¡Error! Marcador no definido.
2. Descripción general del flujo	5
2.1 Envío de petición al TPV Virtual	5
2.2 Recepción del resultado (Notificación on-line)	6
2.3 Retorno del control de la navegación del titular	6
3. Formulario de envío de petición	7
3.1 Identificar la versión de algoritmo de firma a utilizar	7
3.2 Montar la cadena de datos de la petición	7
3.3 Identificar la clave a utilizar para la firma	8
3.4 Firmar los datos de la petición.....	8
3.5 Utilización de librerías de ayuda	9
3.5.1 Librería PHP	9
3.5.2 Librería JAVA.....	11
4. Recepción de la notificación on-line	13
4.1 Notificación Síncrona y Asíncrona	14
4.1.1 Librería PHP	14
4.1.2 Librería JAVA.....	15
4.2 Notificación SOAP	17
4.2.1 Librería PHP	17
4.2.2 Librería JAVA.....	19
5. Retorno del control de la navegación	21
5.1 Utilización de librerías de ayuda	21
5.1.1 Librería PHP	21
5.1.2 Librería JAVA.....	23
6. Códigos de error.....	25

7. ANEXOS.....	30
7.1 Datos de la solicitud de pago	30
7.2 Datos de la notificación on-line.....	32
7.3 Notificación SOAP	35

1. Introducción

1.1 Objetivo

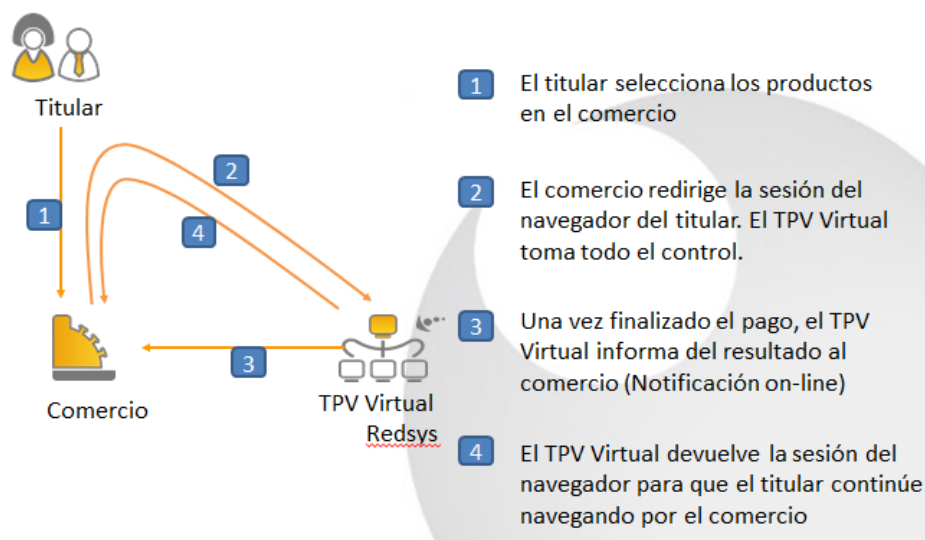
Este documento recoge los aspectos técnicos necesarios para que un comercio realice la integración con el TPV Virtual mediante conexión por Redirección del navegador del cliente comprador.

Esta forma de conexión permite trasladar la sesión del cliente final al TPV Virtual, de forma que la selección del medio de pago y la introducción de datos se llevan a cabo en el entorno seguro del servidor del TPV Virtual y fuera de la responsabilidad del comercio. Además de la sencillez de implementación para el comercio y la tranquilidad respecto a la responsabilidad de los datos de pago, este modo de conexión da cabida a la utilización de mecanismos de autenticación como el 3D Secure, donde el banco de la tarjeta solicita directamente al titular un dato secreto que permite dotar de más seguridad a las compras.

NOTA: la conexión requiere del uso de un sistema de firma basado en HMAC SHA-256, que autentica entre sí al servidor del comercio y al TPV Virtual. Para desarrollar el cálculo de este tipo de firma, el comercio puede realizar el desarrollo por sí mismo utilizando las funciones estándar de los diferentes entornos de desarrollo, si bien para facilitar los desarrollos ponemos a su disposición librerías (PHP y JAVA).

2. Descripción general del flujo

El siguiente esquema presenta el flujo general de una operación realizada con el TPV Virtual.



2.1 Envío de petición al TPV Virtual

Como se muestra en el paso 2 del esquema anterior, el comercio debe enviar al TPV Virtual los datos de la petición de pago a través del navegador del titular. Para ello deberá preparar un formulario con los siguientes campos:

- **Ds_SignatureVersion:** Constante que indica la versión de firma que se está utilizando.
- **Ds_MerchantParameters:** Cadena en formato JSON con todos los parámetros de la petición codificada en Base 64 y sin retornos de carro (En el Anexo 1 del apartado Anexos del presente documento se incluye la lista de parámetros que se pueden enviar en una solicitud de pago).
- **Ds_Signature:** Firma de los datos enviados. Es el resultado del HMAC SHA256 de la cadena JSON codificada en Base 64 enviada en el parámetro anterior.

Este formulario debe enviarse a las siguientes URLs dependiendo de si se quiere realizar una petición de pruebas u operaciones reales:

URL Conexión	Entorno
https://sis-t.redsys.es:25443/sis/realizarPago	Pruebas
https://sis.redsys.es/sis/realizarPago	Real

2.2 Recepción del resultado (Notificación on-line)

Una vez gestionada la transacción, el TPV Virtual puede informar al servidor del comercio el resultado de la misma mediante una conexión directa al servidor del comercio (paso 3 del flujo descrito). Esta notificación es opcional y debe configurarse para cada terminal en el Modulo de Administración.

La notificación on-line consiste en un POST HTTP con la información del resultado. En el POST se incluirán los siguientes campos:

- **Ds_SignatureVersion:** Constante que indica la versión de firma que se está utilizando.
- **Ds_MerchantParameters:** Cadena en formato JSON con todos los parámetros de la respuesta codificada en Base 64 y sin retornos de carro (En el Anexo 2 del apartado Anexos del presente documento se incluye la lista de parámetros que se pueden incluir en la notificación on-line).
- **Ds_Signature:** Firma de los datos enviados. Resultado del HMAC SHA256 de la cadena JSON codificada en Base 64 enviada en el parámetro anterior. **El comercio es responsable de validar el HMAC enviado por el TPV Virtual para asegurarse de la validez de la respuesta. Esta validación es necesaria para garantizar que los datos no han sido manipulados y que el origen es realmente el TPV Virtual.**

NOTA: El TPV Virtual envía la notificación on-line a la URL informada por el comercio en el parámetro **Ds_Merchant_MerchantURL**.

2.3 Retorno del control de la navegación del titular

En el paso 4 del flujo el TPV Virtual devuelve al comercio el control de la navegación del titular. De esta forma el comercio puede completar el flujo del pago manteniendo una secuencia de navegación natural para el cliente/comprador.

Opcionalmente el TPV Virtual puede incluir los mismos campos de la notificación on-line.

3. Formulario de envío de petición

El comercio deberá montar un formulario con los parámetros de la petición de pago que debe hacer llegar al TPV Virtual a través del navegador del cliente. A continuación se muestra un ejemplo del formulario de petición de pago:

```
<form action="https://sis.redsys.es/sis/realizarPago" method="POST" >

  <input type="text" name="Ds_SignatureVersion"
    value="HMAC_SHA256_V1"/>

  <input type="text" name="Ds_MerchantParameters"
    value="eyJEU19NRVJDSEFOVF9BTU9VTlQIiOiIxNDUiLCJEU19NRVJDSEFOVF9PUkrFUiI6IjE0NDI3NzI2NDUiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9DVVJSRU5DWSi6Ijk3OCIsIkRTX01FukNIQU5UX1RSQU5TQUNUSU9OVFIQRSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCi6Ijg3MSIsIkRTX01FukNIQU5UX01FukNIQU5UVVJMIjoiaHR0cHM6XC9cL2VqZW1wbG9cL2VqZW1wbG9fVWJMX05vdGlmLnBocCIIsIkRTX01FukNIQU5UX1VSTE9LIjoiaHR0cHM6XC9cL2VqZW1wbG9cL2VqZW1wbG9fVWJMX09LX0tPLnBocCIIsIkRTX01FukNIQU5UX1VSTE9LIjoiaHR0cHM6XC9cL2VqZW1wbG9cL2VqZW1wbG9fVWJMX09LX0tPLnBocCJ9"/>

  <input type="text" name="Ds_Signature"
    value="hueCwD/cbvrCi+9IDY86WteMpXulII0IDNXNIYgcZHM="/>

  <input type="submit" value="Realizar Pago"/>

</form>
```

Para facilitar la integración del comercio, a continuación se explica de forma detallada los pasos a seguir para montar el formulario de petición de pago.

3.1 Identificar la versión de algoritmo de firma a utilizar

En la petición se debe identificar la versión concreta de algoritmo que se está utilizando para la firma. Actualmente se utiliza el valor **HMAC_SHA256_V1** para identificar la versión de todas las peticiones, por lo que este será el valor del parámetro **Ds_SignatureVersion**, tal y como se puede observar en el ejemplo de formulario mostrado al inicio del apartado 3.

3.2 Montar la cadena de datos de la petición

Se debe montar una cadena con todos los datos de la petición en formato JSON. JSON es un formato abierto de intercambio de datos basado en texto. Al igual que el XML está diseñado para ser legible e independiente de la plataforma tecnológica. La codificación de datos en JSON es muy ligera por lo que es ideal para intercambio de datos en aplicaciones Web.

El nombre de cada parámetro debe indicarse en mayúsculas o con estructura "CamelCase" (Por ejemplo: DS_MERCHANT_AMOUNT o Ds_Merchant_Amount). La lista de parámetros que se pueden incluir en la petición se describe en el Anexo 1 (Datos de la petición de pago) del apartado Anexos del presente documento. A continuación se muestra un ejemplo del objeto JSON de una petición:

```
{
  "DS_MERCHANT_AMOUNT": "145",
  "DS_MERCHANT_ORDER": "1442772645",
  "DS_MERCHANT_MERCHANTCODE": "999008881",
  "DS_MERCHANT_CURRENCY": "978",
  "DS_MERCHANT_TRANSACTIONTYPE": "0",
  "DS_MERCHANT_TERMINAL": "871",
  "DS_MERCHANT_MERCHANTURL": "https://ejemplo/ejemplo_URL_Notif.php",
  "DS_MERCHANT_URLLOK": "https://ejemplo/ejemplo_URL_OK_KO.php",
  "DS_MERCHANT_URLKO": "https://ejemplo/ejemplo_URL_OK_KO.php"
}
```

Una vez montada la cadena JSON con todos los campos, es necesario codificarla en BASE64 sin retornos de carro para asegurarnos de que se mantiene constante y no es alterada en su paso por el navegador del cliente/comprador.

A continuación se muestra el objeto JSON que se acaba de mostrar codificado en BASE64:

```
eyJEU19NRVJDSEFOVF9BTU9VTIQiOiIxNDUiLCJEU19NRVJDSEFOVF9PUkrFUiI6IjE0NDI3
NzI2NDUiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NR
VJDSEFOVF9DVVJSRU5DWSi6Ijk3OCIsIkRTX01FUKNIQU5UX1RSQU5TQUNUSU9OVFIQR
SI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCi6Ijg3MSIsIkRTX01FUKNIQU5UX01FUKNI
QU5UVVJMIjoiaHR0cHM6XC9cL2VqZW1wbG9cL2VqZW1wbG9fVVJMX05vdGlmLnBocCIiI
kRTX01FUKNIQU5UX1VSTE9LIjoiaHR0cHM6XC9cL2VqZW1wbG9cL2VqZW1wbG9fVVJMX
09LX0tPLnBocCIiIkkRTX01FUKNIQU5UX1VSTE9LIjoiaHR0cHM6XC9cL2VqZW1wbG9cL2Vq
ZW1wbG9fVVJMX09LX0tPLnBocCJ9
```

La cadena resultante de la codificación en BASE64 será el valor del parámetro **Ds_MerchantParameters**, tal y como se puede observar en el ejemplo de formulario mostrado al inicio del apartado 3.

NOTA: La utilización de las librerías de ayuda proporcionadas por Banco Sabadell para la generación de este campo, se expone en el apartado 3.5.

3.3 Identificar la clave a utilizar para la firma

Para calcular la firma es necesario utilizar una clave específica para cada terminal. La clave de comercio que debe utilizar es la que recibió a través de SMS desde Banco Sabadell.

NOTA IMPORTANTE: Esta clave debe ser almacenada en el servidor del comercio de la forma más segura posible para evitar un uso fraudulento de la misma. El comercio es responsable de la adecuada custodia y mantenimiento en secreto de dicha clave.

3.4 Firmar los datos de la petición

Una vez se tiene montada la cadena de datos a firmar y la clave específica del terminal se debe calcular la firma siguiendo los siguientes pasos:

1. Se genera una clave específica por operación. Para obtener la clave derivada a utilizar en una operación se debe realizar un cifrado 3DES entre la clave del comercio y el valor del número de pedido de la operación (Ds_Merchant_Order).

2. Se calcula el HMAC SHA256 del valor del parámetro **Ds_MerchantParameters** y la clave obtenida en el paso anterior.
3. El resultado obtenido se codifica en BASE 64, y el resultado de la codificación será el valor del parámetro **Ds_Signature**, tal y como se puede observar en el ejemplo de formulario mostrado al inicio del apartado 3.

NOTA: La utilización de las librerías de ayuda proporcionadas por Banco Sabadell para la generación de este campo, se expone en el apartado 3.5.

3.5 Utilización de librerías de ayuda

En los apartados anteriores se ha descrito la forma de acceso al SIS utilizando conexión por Redirección y el sistema de firma basado en HMAC SHA256. En este apartado se explica cómo se utilizan las librerías disponibles en PHP y JAVA para facilitar los desarrollos y la generación de los campos del formulario de pago. El uso de las librerías suministradas por Banco Sabadell es opcional, si bien simplifican los desarrollos a realizar por el comercio.

3.5.1 Librería PHP

A continuación se presentan los pasos que debe seguir un comercio para la utilización de la librería PHP proporcionada por Banco Sabadell:

1. Importar el fichero principal de la librería, tal y como se muestra a continuación:

```
include_once 'redsysHMAC256_API_PHP_4.0.2/apiRedsys.php';
```

El comercio debe decidir si la importación desea hacerla con la función "include" o "required", según los desarrollos realizados.

2. Definir un objeto de la clase principal de la librería, tal y como se muestra a continuación:

```
$miObj = new RedsysAPI;
```

3. Calcular el parámetro **Ds_MerchantParameters**. Para llevar a cabo el cálculo de este parámetro, inicialmente se deben añadir todos los parámetros de la petición de pago que se desea enviar, tal y como se muestra a continuación:

```

$miObj->setParameter("DS_MERCHANT_AMOUNT",$amount);
$miObj->setParameter("DS_MERCHANT_ORDER",$id);
$miObj->setParameter("DS_MERCHANT_MERCHANTCODE",$fuc);
$miObj->setParameter("DS_MERCHANT_CURRENCY",$moneda);
$miObj->setParameter("DS_MERCHANT_TRANSACTIONTYPE",$trans);
$miObj->setParameter("DS_MERCHANT_TERMINAL",$terminal);
$miObj->setParameter("DS_MERCHANT_MERCHANTURL",$url);
$miObj->setParameter("DS_MERCHANT_URLOK",$urlOK);
$miObj->setParameter("DS_MERCHANT_URLKO",$urlKO);

```

Por último, para calcular el parámetro **Ds_MerchantParameters**, se debe llamar a la función de la librería "createMerchantParameters()", tal y como se muestra a continuación:

```
$params = $miObj->createMerchantParameters();
```

4. Calcular el parámetro **Ds_Signature**. Para llevar a cabo el cálculo de este parámetro, se debe llamar a la función de la librería "createMerchantSignature()" con la clave de comercio facilitada, tal y como se muestra a continuación:

```

$claveModuloAdmin = 'Mk9m98IfEblmPfrpsawt7BmxObt98Jev';
$signature = $miObj->createMerchantSignature($claveModuloAdmin);

```

5. Una vez obtenidos los valores de los parámetros **Ds_MerchantParameters** y **Ds_Signature**, se debe rellenar el formulario de pago con dichos valores, tal y como se muestra a continuación:

```

<form action="https://sis.redsys.es/sis/realizarPago"
method="POST" target="_blank">

    <input type="text" name="Ds_SignatureVersion"
value="HMAC_SHA256_V1"/>
    <input type="text" name="Ds_MerchantParameters"
value="<?php echo $params; ?>"/>
    <input type="text" name="Ds_Signature"
value="<?php echo $signature; ?>"/>
    <input type="submit" value="Realizar Pago"/>

</form>

```

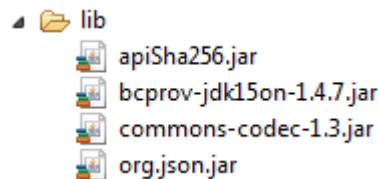
3.5.2 Librería JAVA

A continuación se presentan los pasos que debe seguir un comercio para la utilización de la librería JAVA proporcionada por Banco Sabadell:

1. Importar la librería, tal y como se muestra a continuación:

```
<%@page import="sis.redsys.api.ApiMacSha256"%>
```

El comercio debe incluir en la vía de construcción del proyecto todas las librerías (JARs) que se proporcionan:



2. Calcular el parámetro **Ds_MerchantParameters**. Para llevar a cabo el cálculo de este parámetro, inicialmente se deben añadir todos los parámetros de la petición de pago que se desea enviar, tal y como se muestra a continuación:

```
ApiMacSha256.setParameter("DS_MERCHANT_AMOUNT", amount);  
ApiMacSha256.setParameter("DS_MERCHANT_ORDER", id);  
ApiMacSha256.setParameter("DS_MERCHANT_MERCHANTCODE", fuc);  
ApiMacSha256.setParameter("DS_MERCHANT_CURRENCY", moneda);  
ApiMacSha256.setParameter("DS_MERCHANT_TRANSACTIONTYPE", trans);  
ApiMacSha256.setParameter("DS_MERCHANT_TERMINAL", terminal);  
ApiMacSha256.setParameter("DS_MERCHANT_MERCHANTURL", url);  
ApiMacSha256.setParameter("DS_MERCHANT_URLOK", urlOK);  
ApiMacSha256.setParameter("DS_MERCHANT_URLKO", urlKO);
```

Por último se debe llamar a la función de la librería "createMerchantParameters()", tal y como se muestra a continuación:

```
String params = ApiMacSha256.createMerchantParameters();
```

3. Calcular el parámetro **Ds_Signature**. Para llevar a cabo el cálculo de este parámetro, se debe llamar a la función de la librería "createMerchantSignature()" con la clave de comercio facilitada, tal y como se muestra a continuación:

```
String claveModuloAdmin = "Mk9m98IfEblmPfrpsawt78mx0bt98Jev";  
String signature = ApiMacSha256.createMerchantSignature(claveModuloAdmin);
```

4. Una vez obtenidos los valores de los parámetros **Ds_MerchantParameters** y **Ds_Signature**, se debe rellenar el formulario de pago con los valores obtenidos, tal y como se muestra a continuación:

```
<form action="https://sis.redsys.es/sis/realizarPago"
method="POST" target="_blank">

  <input type="text" name="Ds_SignatureVersion"
value="HMAC_SHA256_V1" />
  <input type="text" name="Ds_MerchantParameters"
value="<%= params %>" />
  <input type="text" name="Ds_Signature"
value="<%= signature %>" />
  <input type="submit" value="Realizar Pago" />

</form>
```

4. Recepción de la notificación on-line

La notificación on-line es una función opcional que permite a la tienda web recibir el resultado de una transacción de forma on-line y en tiempo real, una vez que el cliente ha completado el proceso en el TPV Virtual.

El comercio debe capturar **y validar todos los parámetros junto a la firma** de la notificación on-line de forma previa a cualquier ejecución en su servidor.

El TPV Virtual cuenta con diferentes tipos de notificación y son los siguientes:

- 1. Síncrona.** Implica que el resultado de la compra primero se envía al comercio y a continuación al cliente y con el valor. Aunque la notificación sea errónea la operación no se cambia.
- 2. Asíncrona.** Implica que el resultado de la autorización se comunica a la vez al comercio y al cliente.
- 3. SíncronaSOAP.** La notificación que se envía al comercio es una petición SOAP a un servicio que deberá tener publicado el comercio. Con este tipo de notificación, el SIS no da respuesta al titular hasta que recibe la confirmación del comercio de haber recibido la notificación. En el caso en el que la respuesta SOAP que envíe el comercio tenga un valor KO o que se produzca un error en el proceso de notificación, se dará una respuesta negativa al titular y la operación no se autorizará. Este tipo de notificación solo aplicará a las siguientes operaciones: Autorización, Preautorización, Transacción Recurrente y Autenticación. Para las demás operaciones la notificación se enviará de forma síncrona. En subapartado 4.2 se explica detalladamente este tipo de sincronización.
- 4. SíncronaSOAPcon WSDL.** Igual a la SíncronaSOAP, pero en este caso el servidor SOAP que desarrolla el cliente se ajusta a las especificaciones de una WSDL que se describe en el Anexo 3 (Notificación SOAP) del apartado Anexos del presente documento. Se recomienda este último tipo de notificación, que garantiza un entendimiento perfecto entre servidor y cliente.

La utilización de las librerías de ayuda proporcionadas por Banco Sabadell se expone en los siguientes subapartados y dependerá del tipo de notificación configurada:

4.1 Notificación Síncrona y Asíncrona

En los apartados anteriores se ha descrito la forma de acceso al SIS utilizando conexión por Redirección y el sistema de firma basado en HMAC SHA256. En este apartado se explica cómo se utilizan las librerías disponibles PHP y JAVA para facilitar los desarrollos **para la recepción de los parámetros de la notificación on-line y la validación de la firma**. El uso de las librerías suministradas por Banco Sabadell es opcional, si bien simplifican los desarrollos a realizar por el comercio.

4.1.1 Librería PHP

A continuación se presentan los pasos que debe seguir un comercio para la utilización de la librería PHP proporcionada por Banco Sabadell:

1. Importar el fichero principal de la librería, tal y como se muestra a continuación:

```
include_once 'redsysHMAC256_API_PHP_4.0.2/apiRedsys.php';
```

El comercio debe decidir si la importación desea hacerla con la función "include" o "required", según los desarrollos realizados.

2. Definir un objeto de la clase principal de la librería, tal y como se muestra a continuación:

```
$miObj = new RedsysAPI;
```

3. Capturar los parámetros de la notificación on-line:

```
$version = $_POST["Ds_SignatureVersion"];  
$params = $_POST["Ds_MerchantParameters"];  
$signatureRecibida = $_POST["Ds_Signature"];
```

4. Decodificar el parámetro **Ds_MerchantParameters**. Para llevar a cabo la decodificación de este parámetro, se debe llamar a la función de la librería "decodeMerchantParameters()", tal y como se muestra a continuación:

```
$decoder = $miObj->decodeMerchantParameters($params);
```

Una vez se ha realizado la llamada a la función "decodeMerchantParameters()", se puede obtener el valor de cualquier parámetro que sea susceptible de incluirse en la notificación on-line (Anexo 2 del apartado Anexos del presente documento). Para llevar a cabo la obtención del valor de un parámetro se debe llamar a la función "getParameter()" de la librería con el nombre de parámetro, tal y como se muestra a continuación para obtener el código de respuesta:

```
$codigoRespuesta = $miObj->getParameter("Ds_Response");
```

NOTA IMPORTANTE: Para garantizar la seguridad y el origen de las notificaciones el comercio debe llevar a cabo la validación de la firma recibida y de todos los parámetros que se envían en la notificación.

5. Validar el parámetro **Ds_Signature**. Para llevar a cabo la validación de este parámetro se debe calcular la firma y compararla con el parámetro **Ds_Signature** capturado. Para ello se debe llamar a la función de la librería "createMerchantSignatureNotif()" con la clave de comercio facilitada y el parámetro **Ds_MerchantParameters** capturado, tal y como se muestra a continuación:

```
$claveModuloAdmin = 'Mk9m98IfEblmPfrpsawt7BmxObt98Jev';  
$signatureCalculada = $miObj->createMerchantSignatureNotif($claveModuloAdmin,  
                                                           $params);
```

Una vez hecho esto, ya se puede validar si el valor de la firma enviada coincide con el valor de la firma calculada, tal y como se muestra a continuación:

```
if ($signatureCalculada === $signatureRecibida){  
    echo "FIRMA OK. Realizar tareas en el servidor";  
} else {  
    echo "FIRMA KO. Error, firma inválida";  
}
```

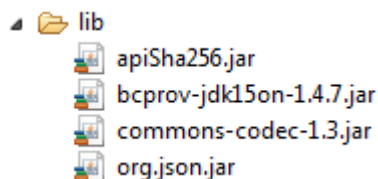
4.1.2 Librería JAVA

A continuación se presentan los pasos que debe seguir un comercio para la utilización de la librería JAVA proporcionada por Banco Sabadell:

1. Importar la librería, tal y como se muestra a continuación:

```
<%@page import="sis.redsys.api.ApiMacSha256"%>
```

El comercio debe incluir en la vía de construcción del proyecto todas las librerías(JARs) que se proporcionan:



2. Capturar los parámetros de la notificación on-line:

```
String version = request.getParameter("Ds_SignatureVersion");
String params = request.getParameter("Ds_MerchantParameters");
String signatureRecibida = request.getParameter("Ds_Signature");
```

3. Decodificar el parámetro **Ds_MerchantParameters**. Para llevar a cabo la decodificación de este parámetro, se debe llamar a la función de la librería "decodeMerchantParameters()", tal y como se muestra a continuación:

```
String decodec = ApiMacSha256.decodeMerchantParameters(params);
```

Una vez se ha realizado la llamada a la función "decodeMerchantParameters()", se puede obtener el valor de cualquier parámetro que sea susceptible de incluirse en la notificación on-line (Anexo 2 del apartado Anexos del presente documento). Para llevar a cabo la obtención del valor de un parámetro se debe llamar a la función "getParameter()" de la librería con el nombre de parámetro, tal y como se muestra a continuación para obtener el código de respuesta:

```
String codigoRespuesta = ApiMacSha256.getParameter("Ds_Response");
```

NOTA IMPORTANTE: Para garantizar la seguridad y el origen de las notificaciones el comercio debe llevar a cabo la validación de la firma recibida y de todos los parámetros que se envían en la notificación.

4. Validar el parámetro **Ds_Signature**. Para llevar a cabo la validación de este parámetro se debe calcular la firma y compararla con el parámetro **Ds_Signature** capturado. Para ello se debe llamar a la función de la librería "createMerchantSignatureNotif()" con la clave de comercio facilitada y el parámetro **Ds_MerchantParameters** capturado, tal y como se muestra a continuación:

```
String claveModuloAdmin = "Mk9m98IfEblmPfrpsawt7Bmx0bt98Jev";
String signatureCalculada = ApiMacSha256.createMerchantSignatureNotif(claveModuloAdmin,
                                                                    params);
```

Una vez hecho esto, ya se puede validar si el valor de la firma enviada coincide con el valor de la firma calculada, tal y como se muestra a continuación:

```
if (signatureCalculada.equals(signatureRecibida)) {
    System.out.println("FIRMA OK. Realizar tareas en el servidor");
} else {
    System.out.println("FIRMA KO. Error, firma inválida");
}
```


4.2 Notificación SOAP

Este método de sincronización permite al comercio recibir una notificación de la transacción en un servicio SOAP. Si el comercio no tiene privilegios para activar este permiso con su usuario, deberá solicitar la activación a través de su entidad. Esta sincronización es una notificación en sí, por lo que no tiene sentido rellenar el campo de notificación online, ya que no se tomará en cuenta.

Si la opción Sincronización SOAP está habilitada para un comercio significará que el SIS enviará las notificaciones para operaciones de Autorización, Preautorización, Autorización en diferido, Transacción Recurrente y Autenticación como peticiones SOAP a un servicio que tendrá publicado el comercio. Para el resto de operaciones las notificaciones se realizarán de forma síncrona y según la opción elegida en la configuración del comercio para las notificaciones on-line.

La principal particularidad de esta notificación es que el SIS espera una respuesta a la notificación antes de presentar el resultado de la operación al titular que está realizando la compra. En el caso en el que el comercio devuelva una respuesta con valor KO o se produzca un error durante el proceso de notificación, el SIS anulará la operación y presentará al titular un recibo con el resultado KO, es decir, el SIS supedita el resultado de la operación a la respuesta que obtenga del comercio en la notificación.

La URL del rpcrouter al que se conectará el SIS y donde estará publicado el servicio SOAP, deberá enviarla el comercio en el parámetro 'Ds_Merchant_MerchantURL' del formulario de entrada al SIS. Las características del servicio SOAP que deben publicar los comercios se describe en el Anexo 3(Notificación SOAP) del apartado Anexos del presente documento.

En este apartado se explica cómo se utilizan las librerías disponibles PHP y JAVA para facilitar los desarrollos para la recepción de los parámetros de la notificación on-line(SOAP) y la validación de la firma. El uso de las librerías suministradas por Banco Sabadell es opcional, si bien simplifican los desarrollos a realizar por el comercio.

4.2.1 Librería PHP

A continuación se presentan los pasos que debe seguir un comercio para la utilización de la librería PHP proporcionada por Banco Sabadell:

1. Importar el fichero principal de la librería, tal y como se muestra a continuación:

```
include_once 'redsysHMAC256_API_PHP_4.0.2/apiRedsys.php';
```

El comercio debe decidir si la importación desea hacerla con la función "include" o "required", según los desarrollos realizados.

2. Definir un objeto de la clase principal de la librería, tal y como se muestra a continuación:

```
$miObj = new RedsysAPI;
```

3. Validar la firma que se envía en la notificación. Para llevar a cabo la validación de este parámetro se debe calcular la firma y compararla con la firma que se envía en la notificación. Para realizar el cálculo de la firma se debe llamar a la función de la librería "createMerchantSignatureNotifSOAPRequest()" con la clave de comercio facilitada y el valor del mensaje recibido en la notificación.

```
function procesaNotificacionSIS($XML) {  
  
$claveModuloAdmin = 'Mk9m98IfEblmPfrpsawt7BmxObt98Jev';  
$signatureCalculada = $miObj->createMerchantSignatureNotifSOAPRequest($claveModuloAdmin,$XML);
```

Una vez hecho esto, el comercio debe capturar el valor de la firma recibida (parámetro **<Signature>**) y validar si el valor de esta coincide con el valor de la firma calculada, tal y como se muestra a continuación:

```
if ($signatureCalculada === $signatureRecibida){  
    echo "FIRMA OK. Realizar tareas en el servidor";  
} else {  
    echo "FIRMA KO. Error, firma inválida";  
}
```

NOTA IMPORTANTE: Para garantizar la seguridad y el origen de las notificaciones el comercio debe llevar a cabo la validación de la firma recibida y de todos los parámetros que se envían en la notificación.

4. Una vez validada la firma, el comercio debe enviar la respuesta de la notificación. Esta respuesta está firmada y para llevar a cabo el cálculo de la firma primero se debe capturar el número de pedido del mensaje recibido en la notificación. Para obtener el número de pedido se debe llamar a la función de la librería "getOrderNotifSOAP()" con el valor del mensaje recibido en la notificación.

Una vez obtenido el número de pedido, tan sólo falta calcular la firma que se enviará en la respuesta. Para realizar el cálculo de la firma se debe llamar a la función de la librería "createMerchantSignatureNotifSOAPResponse()" con la clave de comercio facilitada, el valor del mensaje de respuesta y el número de pedido capturado, tal y como se muestra a continuación:

```

$numPedido = $miObj->getOrderNotifSOAP($XML);

$response='<Response Ds_Version="0.0">
    <Ds_Response_Merchant>OK</Ds_Response_Merchant>
</Response>';

$claveModuloAdmin = 'Mk9m98IfEblmPfrpsawt7BmxObt98Jev';

$firmaRespuesta = $miObj->createMerchantSignatureNotifSOAPResponse($claveModuloAdmin,
                                                                    $response,
                                                                    $numPedido);

```

Por último se debe formar el mensaje final mediante el mensaje de respuesta y la firma obtenida, tal y como se describe en el Anexo 3 (Notificación SOAP) del apartado Anexos del presente documento.

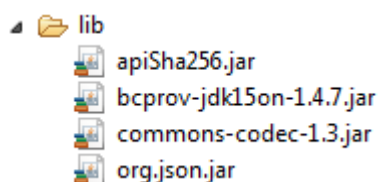
4.2.2 Librería JAVA

A continuación se presentan los pasos que debe seguir un comercio para la utilización de la librería JAVA proporcionada por Banco Sabadell:

1. Importar la librería, tal y como se muestra a continuación:

```
<%@page import="sis.redsys.api.ApiMacSha256"%>
```

El comercio debe incluir en la vía de construcción del proyecto todas las librerías (JARs) que se proporcionan:



2. Validar la firma que se envía en la notificación. Para llevar a cabo la validación de este parámetro se debe calcular la firma y compararla con la firma que se envía en la notificación. Para realizar el cálculo de la firma se debe llamar a la función de la librería "createMerchantSignatureNotifSOAPRequest()" con la clave de comercio facilitada y el valor del mensaje recibido en la notificación.

```

String claveModuloAdmin = "Mk9m98IfEblmPfrpsawt7BmxObt98Jev";
String signatureCalculada = ApiMacSha256.createMerchantSignatureNotifSOAPRequest(claveModuloAdmin,XML);

```

Una vez hecho esto, el comercio debe capturar el valor de la firma recibida (parámetro **<Signature>**) y validar si el valor de esta coincide con el valor de la firma calculada, tal y como se muestra a continuación:

```

if (signatureCalculada.equals(signatureRecibida)) {
    System.out.println("FIRMA OK. Realizar tareas en el servidor");
} else {
    System.out.println("FIRMA KO. Error, firma inválida");
}

```

NOTA IMPORTANTE: Para garantizar la seguridad y el origen de las notificaciones el comercio debe llevar a cabo la validación de la firma recibida y de todos los parámetros que se envían en la notificación.

3. Una vez validada la firma, el comercio debe enviar la respuesta de la notificación. Esta respuesta está firmada y para llevar a cabo el cálculo de la firma primero se debe capturar el número de pedido del mensaje recibido en la notificación. Para obtener el número de pedido se debe llamar a la función de la librería "getOrderNotifSOAP()" con el valor del mensaje recibido en la notificación.

Una vez obtenido el número de pedido, tan sólo falta calcular la firma que se enviará en la respuesta. Para realizar el cálculo de la firma se debe llamar a la función de la librería "createMerchantSignatureNotifSOAPResponse()" con la clave de comercio facilitada, el valor del mensaje de respuesta y el número de pedido capturado, tal y como se muestra a continuación:

```

String numPedido = ApiMacSha256.getOrderNotifSOAP(XML);

String respons = "<Response Ds_Version='0.0'><Ds_Response_Merchant>OK</Ds_Response_Merchant></Response>";

String claveModuloAdmin = "Mk9m98IfEblmPfrpsawt7Bmx0bt98Jev";

String signatureCalculada = ApiMacSha256.createMerchantSignatureNotifSOAPResponse(claveModuloAdmin,
                                                                                      respons,
                                                                                      numPedido);

```

Por último se debe formar el mensaje final mediante el mensaje de respuesta y la firma obtenida, tal y como se describe en el Anexo 3(Notificación SOAP) del apartado Anexos del presente documento.

5. Retorno del control de la navegación

Una vez que el cliente ha realizado el proceso en el TPV Virtual, se redirige la navegación hacia a la tienda web. Este retorno a la web de la tienda se realiza hacia la URL comunicada como parámetro en la llamada inicial al TPV Virtual o en su defecto, se obtiene de la configuración del terminal en el módulo de administración del TPV Virtual. Se pueden disponer de URLs de retorno distintas según el resultado de la transacción (URL OK y URL KO).

El comercio debe capturar y validar, en caso de que la configuración de su comercio así lo requiera (Parámetro en las URLs = SI), los parámetros del retorno de control de navegación previo a cualquier ejecución en su servidor.

La utilización de las librerías de ayuda proporcionadas por Banco Sabadell para la captura y validación de los parámetros del retorno de control de navegación, se expone a continuación.

5.1 Utilización de librerías de ayuda

En los apartados anteriores se ha descrito la forma de acceso al SIS utilizando conexión por Redirección. En este apartado se explica cómo se utilizan las librerías disponibles PHP y JAVA para facilitar los desarrollos para la recepción de los parámetros para la recepción de los parámetros del retorno de control de navegación. El uso de las librerías suministradas por Banco Sabadell es opcional, si bien simplifican los desarrollos a realizar por el comercio.

5.1.1 Librería PHP

A continuación se presentan los pasos que debe seguir un comercio para la utilización de la librería PHP proporcionada por Banco Sabadell:

1. Importar el fichero principal de la librería, tal y como se muestra a continuación:

```
include_once 'redsysHMAC256_API_PHP_4.0.2/apiRedsys.php';
```

El comercio debe decidir si la importación desea hacerla con la función "include" o "required", según los desarrollos realizados.

2. Definir un objeto de la clase principal de la librería, tal y como se muestra a continuación:

```
$miObj = new RedsysAPI;
```

3. Capturar los parámetros de la notificación on-line:

```
$version = $_GET["Ds_SignatureVersion"];  
$params = $_GET["Ds_MerchantParameters"];  
$signatureRecibida = $_GET["Ds_Signature"];
```

4. Decodificar el parámetro **Ds_MerchantParameters**. Para llevar a cabo la decodificación de este parámetro, se debe llamar a la función de la librería "decodeMerchantParameters()", tal y como se muestra a continuación:

```
$decodec = $miObj->decodeMerchantParameters($params);
```

Una vez se ha realizado la llamada a la función "decodeMerchantParameters()", se puede obtener el valor de cualquier parámetro que sea susceptible de incluirse en la notificación on-line (Anexo 2 del apartado Anexos del presente documento). Para llevar a cabo la obtención del valor de un parámetro se debe llamar a la función "getParameter()" de la librería con el nombre de parámetro, tal y como se muestra a continuación para obtener el código de respuesta:

```
$codigoRespuesta = $miObj->getParameter("Ds_Response");
```

NOTA IMPORTANTE: Es importante llevar a cabo la validación de todos los parámetros que se envían en la comunicación. Para actualizar el estado del pedido de forma on-line NO debe usarse esta comunicación, sino la notificación on-line descrita en los otros apartados, ya que el retorno de la navegación depende de las acciones del cliente en su navegador.

5. Validar el parámetro **Ds_Signature**. Para llevar a cabo la validación de este parámetro se debe calcular la firma y compararla con el parámetro **Ds_Signature** capturado. Para ello se debe llamar a la función de la librería "createMerchantSignatureNotif()" con la clave de comercio facilitada y el parámetro **Ds_MerchantParameters** capturado, tal y como se muestra a continuación:

```
$claveModuloAdmin = 'Mk9m98IfEblmPfrpsawt7BmxObt98Jev';  
$signatureCalculada = $miObj->createMerchantSignatureNotif($claveModuloAdmin,  
.....  
..... $params);
```

Una vez hecho esto, ya se puede validar si el valor de la firma enviada coincide con el valor de la firma calculada, tal y como se muestra a continuación:

```
if ($signatureCalculada === $signatureRecibida){  
..... echo "FIRMA OK. Realizar tareas en el servidor";  
} else {  
..... echo "FIRMA KO. Error, firma inválida";  
}
```

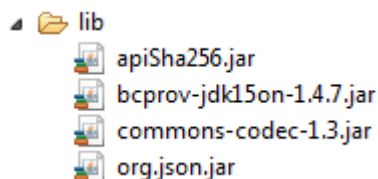
5.1.2 Librería JAVA

A continuación se presentan los pasos que debe seguir un comercio para la utilización de la librería JAVA proporcionada por Banco Sabadell:

1. Importar la librería, tal y como se muestra a continuación:

```
<%@page import="sis.redsys.api.ApiMacSha256"%>
```

El comercio debe incluir en la vía de construcción del proyecto todas las librerías(JARs) que se proporcionan:



2. Capturar los parámetros del retorno de control de navegación:

```
String version = request.getParameter("Ds_SignatureVersion");  
String params = request.getParameter("Ds_MerchantParameters");  
String signatureRecibida = request.getParameter("Ds_Signature");
```

3. Decodificar el parámetro **Ds_MerchantParameters**. Para llevar a cabo la decodificación de este parámetro, se debe llamar a la función de la librería "decodeMerchantParameters()", tal y como se muestra a continuación:

```
String decodec = ApiMacSha256.decodeMerchantParameters(params);
```

Una vez se ha realizado la llamada a la función "decodeMerchantParameters()", se puede obtener el valor de cualquier parámetro que sea susceptible de incluirse en la retorno de control de navegación (Anexo 2 del apartado Anexos del presente documento). Para llevar a cabo la obtención del valor de un parámetro se debe llamar a la función "getParameter()" de la librería con el nombre de parámetro, tal y como se muestra a continuación para obtener el código de respuesta:

```
String codigoRespuesta = ApiMacSha256.getParameter("Ds_Response");
```

NOTA IMPORTANTE: Es importante llevar a cabo la validación de todos los parámetros que se envían en la comunicación. Para actualizar el estado del pedido de forma on-line NO debe usarse esta comunicación, sino la notificación on-line descrita en los otros apartados, ya que el retorno de la navegación depende de las acciones del cliente en su navegador.

4. Validar el parámetro **Ds_Signature**. Para llevar a cabo la validación de este parámetro se debe calcular la firma y compararla con el parámetro **Ds_Signature** capturado. Para ello

se debe llamar a la función de la librería "createMerchantSignatureNotif()" con la clave de comercio facilitada y el parámetro **Ds_MerchantParameters** capturado, tal y como se muestra a continuación:

```
String claveModuloAdmin = "Mk9m98IfEblmPfrpsawt7Bmx0bt98Jev";  
String signatureCalculada = ApiMacSha256.createMerchantSignatureNotif(claveModuloAdmin,  
                                                                    params);
```

Una vez hecho esto, ya se puede validar si el valor de la firma enviada coincide con el valor de la firma calculada, tal y como se muestra a continuación:


```
if (signatureCalculada.equals(signatureRecibida)) {  
    System.out.println("FIRMA OK. Realizar tareas en el servidor");  
} else {  
    System.out.println("FIRMA KO. Error, firma inválida");  
}
```


6. Códigos de error

En este apartado se presenta un glosario de los errores que se pueden producir en el proceso de integración.

El error que se ha producido se puede obtener consultando el código fuente de la página de resultado de la operación, tal y como se muestra a continuación:

Página de resultado de la operación

 Seleccione su idioma Castellano

1 Seleccione método de pago


2 Comprobación autenticación


3 Solicitando Autorización

4 Resultado Transacción

Datos de la operación


Importe:	1,45 Euros
Código Comercio:	Comercio Pruebas
Terminal:	999008881-1
Número pedido:	2063xJ990Y4
Fecha:	06/10/2015 16:45

 No se puede realizar la operación
Número de pedido repetido

 **CANCELAR**

Página de resultado de la operación (código fuente)

```
192 <div class="result-code error">
193 <p>
194 <text lngid="noSePuedeRealizarOperacion">No se puede
195 <br>Número de pedido repetido<br>
196 <!--SIS0051:-->
197 </p>
198 </div>
199 </div>
200 <div class="result-info">
201 <table class="tablereults">
202 <tbody></tbody>
```



Glosario de errores del SIS

ERROR	DESCRIPCIÓN	MENSAJE (ANEXO VI)
SIS0007	Error al desmontar el XML de entrada	MSG0008
SIS0008	Error faltaDs_Merchant_MerchantCode	MSG0008
SIS0009	Error de formato en Ds_Merchant_MerchantCode	MSG0008
SIS0010	Error falta Ds_Merchant_Terminal	MSG0008
SIS0011	Error de formato en Ds_Merchant_Terminal	MSG0008
SIS0014	Error de formato en Ds_Merchant_Order	MSG0008
SIS0015	Error falta Ds_Merchant_Currency	MSG0008
SIS0016	Error de formato en Ds_Merchant_Currency	MSG0008
SIS0017	Error no se admiten operaciones en pesetas	MSG0008
SIS0018	Error falta Ds_Merchant_Amount	MSG0008
SIS0019	Error de formato en Ds_Merchant_Amount	MSG0008
SIS0020	ErrorfaltaDs_Merchant_MerchantSignature	MSG0008
SIS0021	Error la Ds_Merchant_MerchantSignaturevienevacía	MSG0008
SIS0022	Error de formato en Ds_Merchant_TransactionType	MSG0008
SIS0023	Error Ds_Merchant_TransactionType desconocido	MSG0008
SIS0024	Error Ds_Merchant_ConsumerLanguage tiene mas de 3 posiciones	MSG0008
SIS0025	Error de formato en Ds_Merchant_ConsumerLanguage	MSG0008
SIS0026	Error No existe el comercio / terminal enviado	MSG0008
SIS0027	Error Moneda enviada por el comercio es diferente a la que tiene asignada para ese terminal	MSG0008
SIS0028	Error Comercio / terminal está dado de baja	MSG0008
SIS0030	Error en un pago con tarjeta ha llegado un tipo de operación que no es ni pago ni preautorización	MSG0000
SIS0031	Método de pago no definido	MSG0000
SIS0033	Error en un pago con móvil ha llegado un tipo de operación que no es ni pago ni preautorización	MSG0000
SIS0034	Error de acceso a la Base de Datos	MSG0000
SIS0037	El número de teléfono no es válido	MSG0000
SIS0038	Error en java	MSG0000
SIS0040	Error el comercio / terminal no tiene ningún método de pago asignado	MSG0008
SIS0041	Error en el cálculo de la HASH de datos del comercio.	MSG0008
SIS0042	La firma enviada no es correcta	MSG0008
SIS0043	Error al realizar la notificación on-line	MSG0008
SIS0046	El bin de la tarjeta no está dado de alta	MSG0002
SIS0051	Error número de pedido repetido	MSG0001
SIS0054	Error no existe operación sobre la que realizar la devolución	MSG0008
SIS0055	Error existe más de un pago con el mismo número de pedido	MSG0008

ERROR	DESCRIPCIÓN	MENSAJE (ANEXO VI)
SIS0056	La operación sobre la que se desea devolver no está autorizada	MSG0008
SIS0057	El importe a devolver supera el permitido	MSG0008
SIS0058	Inconsistencia de datos, en la validación de una confirmación	MSG0008
SIS0059	Error no existe operación sobre la que realizar la confirmación	MSG0008
SIS0060	Ya existe una confirmación asociada a la preautorización	MSG0008
SIS0061	La preautorización sobre la que se desea confirmar no está autorizada	MSG0008
SIS0062	El importe a confirmar supera el permitido	MSG0008
SIS0063	Error. Número de tarjeta no disponible	MSG0008
SIS0064	Error. El número de tarjeta no puede tener más de 19 posiciones	MSG0008
SIS0065	Error. El número de tarjeta no es numérico	MSG0008
SIS0066	Error. Mes de caducidad no disponible	MSG0008
SIS0067	Error. El mes de la caducidad no es numérico	MSG0008
SIS0068	Error. El mes de la caducidad no es válido	MSG0008
SIS0069	Error. Año de caducidad no disponible	MSG0008
SIS0070	Error. El Año de la caducidad no es numérico	MSG0008
SIS0071	Tarjeta caducada	MSG0000
SIS0072	Operación no anulable	MSG0000
SIS0074	Error falta Ds_Merchant_Order	MSG0008
SIS0075	Error el Ds_Merchant_Order tiene menos de 4 posiciones o más de 12	MSG0008
SIS0076	Error el Ds_Merchant_Order no tiene las cuatro primeras posiciones numéricas	MSG0008
SIS0077	Error el Ds_Merchant_Order no tiene las cuatro primeras posiciones numéricas. No se utiliza	MSG0000
SIS0078	Método de pago no disponible	MSG0005
SIS0079	Error al realizar el pago con tarjeta	MSG0000
SIS0081	La sesión es nueva, se han perdido los datos almacenados	MSG0007
SIS0084	El valor de Ds_Merchant_Conciliation es nulo	MSG0008
SIS0085	El valor de Ds_Merchant_Conciliation no es numérico	MSG0008
SIS0086	El valor de Ds_Merchant_Conciliation no ocupa 6 posiciones	MSG0008
SIS0089	El valor de Ds_Merchant_ExpiryDate no ocupa 4 posiciones	MSG0008
SIS0092	El valor de Ds_Merchant_ExpiryDate es nulo	MSG0008
SIS0093	Tarjeta no encontrada en la tabla de rangos	MSG0006
SIS0094	La tarjeta no fue autenticada como 3D Secure	MSG0004
SIS0097	Valor del campo Ds_Merchant_CComercio no válido	MSG0008
SIS0098	Valor del campo Ds_Merchant_CVentana no válido	MSG0008
SIS0112	Error El tipo de transacción especificado en Ds_Merchant_Transaction_Type no esta permitido	MSG0008
SIS0113	Excepción producida en el servlet de operaciones	MSG0008
SIS0114	Error, se ha llamado con un GET en lugar de un POST	MSG0000
SIS0115	Error no existe operación sobre la que realizar el pago de la cuota	MSG0008

ERROR	DESCRIPCIÓN	MENSAJE (ANEXO VI)
SIS0116	La operación sobre la que se desea pagar una cuota no es una operación válida	MSG0008
SIS0117	La operación sobre la que se desea pagar una cuota no está autorizada	MSG0008
SIS0118	Se ha excedido el importe total de las cuotas	MSG0008
SIS0119	Valor del campo Ds_Merchant_DateFrequency no válido	MSG0008
SIS0120	Valor del campo Ds_Merchant_ChargeExpiryDate no válido	MSG0008
SIS0121	Valor del campo Ds_Merchant_SumTotal no válido	MSG0008
SIS0122	Valor del campo Ds_Merchant_DateFrequency o no Ds_Merchant_SumTotal tiene formato incorrecto	MSG0008
SIS0123	Se ha excedido la fecha tope para realizar transacciones	MSG0008
SIS0124	No ha transcurrido la frecuencia mínima en un pago recurrente sucesivo	MSG0008
SIS0132	La fecha de Confirmación de Autorización no puede superar en más de 7 días a la de Preautorización.	MSG0008
SIS0133	La fecha de Confirmación de Autenticación no puede superar en más de 45 días a la de Autenticación Previa.	MSG0008
SIS0139	Error el pago recurrente inicial está duplicado	MSG0008
SIS0142	Tiempo excedido para el pago	MSG0000
SIS0197	Error al obtener los datos de cesta de la compra en operación tipo pasarela	MSG0000
SIS0198	Error el importe supera el límite permitido para el comercio	MSG0000
SIS0199	Error el número de operaciones supera el límite permitido para el comercio	MSG0008
SIS0200	Error el importe acumulado supera el límite permitido para el comercio	MSG0008
SIS0214	El comercio no admite devoluciones	MSG0008
SIS0216	Error Ds_Merchant_CVV2 tiene más de 3 posiciones	MSG0008
SIS0217	Error de formato en Ds_Merchant_CVV2	MSG0008
SIS0218	El comercio no permite operaciones seguras por la entrada /operaciones	MSG0008
SIS0219	Error el número de operaciones de la tarjeta supera el límite permitido para el comercio	MSG0008
SIS0220	Error el importe acumulado de la tarjeta supera el límite permitido para el comercio	MSG0008
SIS0221	Error el CVV2 es obligatorio	MSG0008
SIS0222	Ya existe una anulación asociada a la preautorización	MSG0008
SIS0223	La preautorización que se desea anular no está autorizada	MSG0008
SIS0224	El comercio no permite anulaciones por no tener firma ampliada	MSG0008
SIS0225	Error no existe operación sobre la que realizar la anulación	MSG0008
SIS0226	Inconsistencia de datos, en la validación de una anulación	MSG0008
SIS0227	Valor del campo Ds_Merchant_TransactionDate no válido	MSG0008
SIS0229	No existe el código de pago aplazado solicitado	MSG0008
SIS0252	El comercio no permite el envío de tarjeta	MSG0008
SIS0253	La tarjeta no cumple el check-digit	MSG0006
SIS0254	El número de operaciones de la IP supera el límite permitido por el comercio	MSG0008
SIS0255	El importe acumulado por la IP supera el límite permitido por el comercio	MSG0008
SIS0256	El comercio no puede realizar preautorizaciones	MSG0008

ERROR	DESCRIPCIÓN	MENSAJE (ANEXO VI)
SIS0257	Esta tarjeta no permite operativa de preautorizaciones	MSG0008
SIS0258	Inconsistencia de datos, en la validación de una confirmación	MSG0008
SIS0261	Operación detenida por superar el control de restricciones en la entrada al SIS	MSG0008
SIS0270	El comercio no puede realizar autorizaciones en diferido	MSG0008
SIS0274	Tipo de operación desconocida o no permitida por esta entrada al SIS	MSG0008
SIS0429	Error en la versión enviada por el comercio en el parámetro Ds_SignatureVersion	MSG0008
SIS0430	Error al decodificar el parámetro Ds_MerchantParameters	MSG0008
SIS0431	Error del objeto JSON que se envía codificado en el parámetro Ds_MerchantParameters	MSG0008
SIS0432	Error FUC del comercio erróneo	MSG0008
SIS0433	Error Terminal del comercio erróneo	MSG0008
SIS0434	Error ausencia de número de pedido en la operación enviada por el comercio	MSG0008
SIS0435	Error en el cálculo de la firma	MSG0008

7. ANEXOS

7.1 Datos de la solicitud de pago

En la petición de pago hacia el TPV Virtual SIS se tendrán que enviar una serie de datos obligatorios y otros opcionales.

Los datos obligatorios para la gestión de la transacción están marcados como tales en la columna *Comentarios* de la tabla siguiente.

DATO	NOMBRE DEL DATO	Long. / Tipo	COMENTARIOS
Identificación de comercio: código FUC	<i>Ds_Merchant_MerchantCode</i>	9/N.	Obligatorio. Código FUC asignado al comercio.
Número de terminal	<i>Ds_Merchant_Terminal</i>	3/N.	Obligatorio. Número de terminal que le asignará su banco. Tres se considera su longitud máxima
Tipo de transacción	<i>Ds_Merchant_TransactionType</i>	1 / Num	Obligatorio. para el comercio para indicar qué tipo de transacción es. Los posibles valores son: 0 – Autorización 1 – Preautorización 2 – Confirmación de preautorización 3 – Devolución Automática 5 – Transacción Recurrente 6 – Transacción Sucesiva 7 – Pre-autenticación 8 – Confirmación de pre-autenticación 9 – Anulación de Preautorización O – Autorización en diferido P – Confirmación de autorización en diferido Q – Anulación de autorización en diferido R – Cuota inicial diferido S – Cuota sucesiva diferido
Importe	<i>Ds_Merchant_Amount</i>	12 / Núm.	Obligatorio. Para Euros las dos últimas posiciones se consideran decimales.
Moneda	<i>Ds_Merchant_Currency</i>	4 / Núm.	Obligatorio. Se debe enviar el código numérico de la moneda según el ISO-4217, por ejemplo: 978 euros 840 dólares 826 libras 392 yenes 4 se considera su longitud máxima
Número de Pedido	<i>Ds_Merchant_Order</i>	12 / A-N.	Obligatorio. Los 4 primeros dígitos deben ser numéricos, para los dígitos restantes solo utilizar los siguientes caracteres ASCII Del 30 = 0 al 39 = 9 Del 65 = A al 90 = Z Del 97 = a al 122 = z
URL del comercio para la notificación "on-line"	<i>Ds_Merchant_MerchantURL</i>	250/A-N	Obligatorio si el comercio tiene notificación "on-line". URL del comercio que recibirá un post con los datos de la transacción.
Descripción del producto	<i>Ds_Merchant_ProductDescription</i>	125 / A-N	Opcional. 125 se considera su longitud máxima. Este campo se mostrará al titular en la pantalla de confirmación de la compra.
Nombre y apellidos del titular	<i>Ds_Merchant_Titular</i>	60/A-N	Opcional. Su longitud máxima es de 60 caracteres. Este campo se mostrará al titular en la pantalla de confirmación de la compra.
URLOK	<i>Ds_Merchant_UrlOK</i>	250/A-N	Opcional: si se envía será utilizado como URLOK ignorando el configurado en el módulo de administración en caso de tenerlo.
URL KO	<i>Ds_Merchant_UrlKO</i>	250/A-	Opcional: si se envía será utilizado como URLKO

DATO	NOMBRE DEL DATO	Long. / Tipo	COMENTARIOS
		N	ignorando el configurado en el módulo de administración en caso de tenerlo
Identificación de comercio: denominación comercial	<i>Ds_Merchant_MerchantName</i>	25/A-N	Opcional: será el nombre del comercio que aparecerá en el ticket del cliente (opcional).
Idioma del titular	<i>Ds_Merchant_ConsumerLanguage</i>	3/N.	Opcional: el Valor 0, indicará que no se ha determinado el idioma del cliente (opcional). Otros valores posibles son: Castellano-001, Inglés-002, Catalán-003, Francés-004, Alemán-005, Holandés-006, Italiano-007, Sueco-008, Portugués-009, Valenciano-010, Polaco-011, Gallego-012 y Euskera-013.
Importe total (cuota recurrente)	<i>Ds_Merchant_SumTotal</i>	12/N.	Obligatorio. Representa la suma total de los importes de las cuotas. Las dos últimas posiciones se consideran decimales.
Datos del comercio	<i>Ds_Merchant_MerchantData</i>	1024 /A-N	Opcional para el comercio para ser incluidos en los datos enviados por la respuesta "on-line" al comercio si se ha elegido esta opción.
Frecuencia	<i>Ds_Merchant_DateFrequency</i>	5/ N	Frecuencia en días para las transacciones recurrentes y recurrentes diferidas (obligatorio para recurrentes)
Fecha límite	<i>Ds_Merchant_ChargeExpiryDate</i>	10/ A-N	Formato yyyy-MM-dd fecha límite para las transacciones Recurrentes (Obligatorio para recurrentes y recurrentes diferidas)
Código de Autorización	<i>Ds_Merchant_AuthorisationCode</i>	6 / Num	Opcional. Representa el código de autorización necesario para identificar una transacción recurrente sucesiva en las devoluciones de operaciones recurrentes sucesivas. Obligatorio en devoluciones de operaciones recurrentes.
Fecha de la operación recurrente sucesiva	<i>Ds_Merchant_TransactionDate</i>	10 / A-N	Opcional. Formato yyyy-mm-dd. Representa la fecha de la cuota sucesiva, necesaria para identificar la transacción en las devoluciones. Obligatorio en las devoluciones de cuotas sucesivas y de cuotas sucesivas diferidas.

7.2 Datos de la notificación on-line

Recomendamos el uso de este método, ya que permite que la tienda web reciba el resultado de las transacciones, de forma on-line en tiempo real. La Notificación ON-LINE es configurable en el módulo de administración, y admite varias posibilidades en función de la necesidad del comercio. Tanto la notificación HTTP como la notificación por mail tienen exactamente el mismo formato.

La notificación por HTTP es una comunicación en paralelo y de forma independiente al proceso de navegación del cliente por el TPV Virtual, mediante la cual se envía al comercio un POST con los datos del resultado de la operación. Evidentemente, en el lado del servidor del comercio, deberá haber un proceso que recoja esta respuesta y realice las tareas necesarias para la gestión de los pedidos. Para ello tendrá que facilitar, como parámetro, una URL donde recibir estas respuestas en el formulario web que envía al realizar la solicitud de autorización (ver el campo `Ds_Merchant_MerchantURL` en "Datos del formulario de pago"). Esta URL será un CGI, Servlet, etc. desarrollado en el lenguaje que el comercio considere adecuado para integrar en su Servidor (C, Java, Perl, PHP, ASP, etc.), capaz de interpretar la respuesta que le envíe el TPV Virtual. Se puede especificar un URL diferente las operaciones con resultado OK y otra para las KO.

NOTA: Estos mismos datos se incorporarán en la URL OK (`Ds_Merchant_UrlOK`) o URL KO (`Ds_Merchant_UrlKO`) si el comercio tiene activado el envío de parámetros en la redirección de respuesta.

DATO	NOMBRE DEL DATO	LONG/TIPO	COMENTARIOS
Fecha	<code>Ds_Date</code>	<code>dd/mm/yyyy</code>	Fecha de la transacción
Hora	<code>Ds_Hour</code>	<code>HH:mm</code>	Hora de la transacción
Importe	<code>Ds_Amount</code>	<code>12 / Núm.</code>	Mismo valor que en la petición.
Moneda	<code>Ds_Currency</code>	<code>4 / Núm.</code>	Mismo valor que en la petición. 4 se considera su longitud máxima.
Número de pedido	<code>Ds_Order</code>	<code>12 / A-N.</code>	Mismo valor que en la petición.
Identificación de comercio: código FUC	<code>Ds_MerchantCode</code>	<code>9 / N.</code>	Mismo valor que en la petición.
Terminal	<code>Ds_Terminal</code>	<code>3 / Núm.</code>	Número de terminal que le asignará su banco. 3 se considera su longitud máxima.
Código de respuesta	<code>Ds_Response</code>	<code>4 / Núm.</code>	Ver tabla siguiente (Posibles valores del <code>Ds_Response</code>).
Datos del comercio	<code>Ds_MerchantData</code>	<code>1024 / A-N</code>	Información opcional enviada por el comercio en el formulario de pago.
Pago Seguro	<code>Ds_SecurePayment</code>	<code>1 / Núm.</code>	0 – Si el pago NO es seguro 1 – Si el pago es seguro
Tipo de operación	<code>Ds_TransactionType</code>	<code>1 / A-N</code>	Tipo de operación que se envió en el formulario de pago
País del titular	<code>Ds_Card_Country</code>	<code>3/Núm</code>	Opcional: País de emisión de la tarjeta con la que se ha intentado realizar el pago. En el siguiente enlace es posible consultar los códigos de país y su correspondencia:

DATO	NOMBRE DEL DATO	LONG/TIPO	COMENTARIOS
			http://unstats.un.org/unsd/methods/m49/m49alpha.htm
Código de autorización	Ds_AuthorisationCode	6/ A-N	Opcional: Código alfanumérico de autorización asignado a la aprobación de la transacción por la institución autorizadora.
Idioma del titular	Ds_ConsumerLanguage	3 / Núm	Opcional: El valor 0, indicará que no se ha determinado el idioma del cliente. (opcional). 3 se considera su longitud máxima.
Tipo de Tarjeta	Ds_Card_Type	1 / A-N	Opcional: Valores posibles: C – Crédito D - Débito


Estos son los posibles valores del Ds_Response o "Código de respuesta":

CÓDIGO	SIGNIFICADO
101	Tarjeta caducada
102	Tarjeta en excepción transitoria o bajo sospecha de fraude
106	Intentos de PIN excedidos
125	Tarjeta no efectiva
129	Código de seguridad (CVV2/CVC2) incorrecto
180	Tarjeta ajena al servicio
184	Error en la autenticación del titular
190	Denegación del emisor sin especificar motivo
191	Fecha de caducidad errónea
202	Tarjeta en excepción transitoria o bajo sospecha de fraude con retirada de tarjeta
904	Comercio no registrado en FUC
909	Error de sistema
913	Pedido repetido
944	Sesión Incorrecta
950	Operación de devolución no permitida
9912/912	Emisor no disponible
9064	Número de posiciones de la tarjeta incorrecto
9078	Tipo de operación no permitida para esa tarjeta
9093	Tarjeta no existente
9094	Rechazo servidores internacionales
9104	Comercio con "titular seguro" y titular sin clave de compra segura
9218	El comercio no permite op. seguras por entrada /operaciones
9253	Tarjeta no cumple el check-digit
9256	El comercio no puede realizar preautorizaciones
9257	Esta tarjeta no permite operativa de preautorizaciones
9261	Operación detenida por superar el control de restricciones en la entrada al SIS
9913	Error en la confirmación que el comercio envía al TPV Virtual (solo aplicable en la opción de sincronización SOAP)
9914	Confirmación "KO" del comercio (solo aplicable en la opción de sincronización SOAP)
9915	A petición del usuario se ha cancelado el pago
9928	Anulación de autorización en diferido realizada por el SIS (proceso batch)
9929	Anulación de autorización en diferido realizada por el comercio

9997	Se está procesando otra transacción en SIS con la misma tarjeta
9998	Operación en proceso de solicitud de datos de tarjeta
9999	Operación que ha sido redirigida al emisor a autenticar

Estos códigos de respuesta se muestran en el campo "Código de respuesta" de la consulta de operaciones, siempre y cuando la operación no está autorizada, tal y como se muestra en la siguiente imagen:

Página 1 de 3

Sesión / Fecha Totales	Fecha Hora	Tipo Operación Num. Pedido	Resultado NºAutorización o Cod.Respuesta	Importe	Neto Lote/Cajón
01-10-15	01-10-2015 16:50:16	Autorización Tradicional 151001165015	Sin Finalizar 9997		
01-10-15	01-10-2015 16:50:23	Autorización Tradicional 151001165022	Autorizada 581956	1,00 EUR	2 /

7.3 Notificación SOAP

El servicio SOAP que deben publicar los comercios debe tener las siguientes características:

1. El servicio deberá llamarse 'InotificacionSIS' y ofrecer un método llamado 'procesaNotificacionSIS'. Este método estará definido con un parámetro de entrada tipo cadena XML y otro parámetro de salida del mismo tipo. Para más información, se adjunta un fichero WSDL a partir del cual se puede construir el esqueleto del servidor y que servirá para definir los tipos de datos que se intercambiarán entre cliente y servidor, de cara a facilitar la comunicación.
2. El formato de los mensajes que se intercambiarán en este servicio deberán ajustarse a la siguiente dtd:
3. Mensaje de notificación enviado desde el SIS con los datos de la operación correspondiente:

```
<!ELEMENT Message (Request, Signature)>

<!ELEMENT Request (Fecha, Hora, Ds_SecurePayment, Ds_Amount, Ds_Currency, Ds_Order,
Ds_MerchantCode, Ds_Terminal, Ds_Response, Ds_MerchantData?, Ds_Card_Type?,
DS_Card_Type?, Ds_TransactionType, Ds_ConsumerLanguage, Ds_ErrorCode?,
Ds_CardCountry?, Ds_AuthorisationCode?)>

<!ATTLIST Request Ds_Version CDATA #REQUIRED>

<!ELEMENT Fecha (#PCDATA)>
<!ELEMENT Hora (#PCDATA)>
<!ELEMENTDs_SecurePayment(#PCDATA)>
<!ELEMENTDs_Amount (#PCDATA)>
<!ELEMENTDs_Currency (#PCDATA)>
<!ELEMENTDs_Order (#PCDATA)>
<!ELEMENTDs_MerchantCode (#PCDATA)>
<!ELEMENTDs_Terminal (#PCDATA)>
<!ELEMENTDs_Response (#PCDATA)>
<!ELEMENTDs_MerchantData (#PCDATA)>
<!ELEMENTDs_Card_Type (#PCDATA)>
<!ELEMENTDs_TransactionType (#PCDATA)>
<!ELEMENTDs_ConsumerLanguage (#PCDATA)>
<!ELEMENTDs_ErrorCode (#PCDATA)>
<!ELEMENTDs_CardCountry (#PCDATA)>
<!ELEMENTDs_AuthorisationCode (#PCDATA)>
<!ELEMENT Signature (#PCDATA)>
<!ELEMENTDS_Card_Type (#PCDATA)>
```

Para generar el valor del campo Signature en el mensaje de respuesta del comercio aplicaremos un HMAC SHA-256 de la cadena <Request ...>...</Request>.

Ejemplo:

Sea el siguiente mensaje:

```

    <Message>
    <RequestDs_Version="0.0">
    <Fecha>01/04/2003</Fecha>
    <Hora>16:57</Hora>
    <Ds_SecurePayment>1</Ds_SecurePayment>
    <Ds_Amount>345</Ds_Amount>
    <Ds_Currency>978</Ds_Currency>
    <Ds_Order>165446</Ds_Order>
    <Ds_MerchantCode>999008881</Ds_MerchantCode>
        <Ds_Terminal>001</Ds_Terminal>
            <Ds_Card_Country>724</Ds_Card_Country>
    <Ds_Response>0000</Ds_Response>
    <Ds_MerchantData>Alfombrilla para raton</Ds_MerchantData>
        <Ds_Card_Type>C</Ds_Card_Type>
    <Ds_TransactionType>1</Ds_TransactionType>
    <Ds_ConsumerLanguage>1</Ds_ConsumerLanguage>
    </Request>
    </Message>

```

Mensaje de respuesta del comercio a la notificación:

Ejemplo:

```

<!ELEMENT Message (Response, Signature)>
<!ELEMENT Response (Ds_Response_Merchant)>
<!ATTLIST Response Ds_Version CDATA #REQUIRED>
<!ELEMENT Ds_Response_Merchant (#PCDATA)>
<!ELEMENTSignature (#PCDATA)>

```

Los posibles valores que podrá tomar la etiqueta Ds_Response_Merchant serán:

- 'OK' cuando la notificación se ha recibido correctamente.
- 'KO' cuando se ha producido algún error.

Para generar el valor del campo Signature en el mensaje de respuesta del comercio aplicaremos un HMAC SHA-256 de la cadena <Response>...</Response>.

• **Ejemplos de mensajes intercambiados en una notificación con Sincronización SOAP:**

Mensaje de notificación enviado desde el SIS:

```

    <Message>
    <RequestDs_Version="0.0">
    <Fecha>01/04/2003</Fecha>
    <Hora>16:57</Hora>
    <Ds_SecurePayment>1</Ds_SecurePayment>
    <Ds_Amount>345</Ds_Amount>
    <Ds_Currency>978</Ds_Currency>
    <Ds_Order>165446</Ds_Order>
    <Ds_Card_Type>C</Ds_Card_Type>
    <Ds_MerchantCode>999008881</Ds_MerchantCode>

```

```

    <Ds_Terminal>001</Ds_Terminal>
    <Ds_Card_Country>724</Ds_Card_Country>
<Ds_Response>0000</Ds_Response>
<Ds_MerchantData>Alfombrilla para raton</Ds_MerchantData>
<Ds_TransactionType>1</Ds_TransactionType>
<Ds_ConsumerLanguage>1</Ds_ConsumerLanguage>
  </Request>
  <Signature>I3gacbQMEvUYN59YiHkImI-crEMwFAeogI1jLBDFiw=</Signature>
</Message>

```

Mensaje de respuesta desde el comercio al SIS:

```

  <Message>
    <ResponseDs_Version="0.0">
      <Ds_Response_Merchant>OK</Ds_Response_Merchant>
    </Response>
  <Signature>d/VtqOzNlds9MTL/QO12TvGDNT+yTfawFlg55ZcjX9Q=</Signature>
</Message>

```

WSDL para el servicio InotificacionSIS

Los comercios que deseen desarrollar un servicio SOAP deben ajustarse a esta WSDL. A partir de ella y, mediante herramientas de generación automática de código, se puede desarrollar el esqueleto del servidor SOAP de forma cómoda y rápida.

La WSDL que debe cumplir el servicio SOAP desarrollado por el cliente es la siguiente:

```

<?xml version="1.0" encoding="UTF-8"?>

  <definitions name="InotificacionSIS"
targetNamespace=https://sis.SERMEPA.es/sis/InotificacionSIS.wsdl xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:tns="https://sis.SERMEPA.es/sis/InotificacionSIS.wsdl"
  xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns="http://schemas.xmlsoap.org/wsdl/">

    <message name="procesaNotificacionSISRequest">
      <part name="XML" type="xs:string"/>
    </message>

    <message name="procesaNotificacionSISResponse">
      <part name="return" type="xs:string"/>
    </message>

    <portType name="InotificacionSISPortType">
      <operation name="procesaNotificacionSIS">
        <input message="tns:procesaNotificacionSISRequest"/>
        <output message="tns:procesaNotificacionSISResponse"/>
      </operation>
    </portType>

    <binding name="InotificacionSISBinding" type="tns:InotificacionSISPortType">
      <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
      <operation name="procesaNotificacionSIS">
        <soap:operation
soapAction="urn:InotificacionSIS#procesaNotificacionSIS" style="rpc"/>
        <input>

```

```

    <soap:body use="encoded"
encodingStyle=http://schemas.xmlsoap.org/soap/encoding/ namespace="InotificacionSIS"/>
    </input>
    <output>
    <soap:body use="encoded"
encodingStyle=http://schemas.xmlsoap.org/soap/encoding/ namespace="InotificacionSIS"/>
    </output>
    </operation>
  </binding>

  <service name="InotificacionSISService">
    <port name="InotificacionSIS" binding="tns:InotificacionSISBinding">
      <soap:address location="http://localhost/WebServiceSIS/InotificacionSIS.asmx"/>
    </port>
  </service>

</definitions>

```