



**Guía de migración a firma HMAC SHA256  
Conexión por Redirección**

**SERVICIO TECNICO TPV VIRTUAL**

Teléfono: 902 365 650 opción 2

[tpvvirtual@bancsabadell.com](mailto:tpvvirtual@bancsabadell.com)

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

## ÍNDICE DE CONTENIDO

<b>1. Introducción</b>	<b>4</b>
1.1 Objetivo	4
<b>2. Resumen de las diferencias del nuevo modelo (HMAC SHA-256) respecto del modelo anterior (SHA-1)</b>	<b>5</b>
<b>3. Descripción general del flujo</b>	<b>6</b>
3.1 Envío de petición al TPV Virtual	6
3.2 Recepción del resultado (Notificación on-line)	7
3.3 Retorno del control de la navegación del titular	7
<b>4. Formulario de envío de petición</b>	<b>8</b>
4.1 Identificar la versión de algoritmo de firma a utilizar	10
4.2 Montar la cadena de datos de la petición	10
4.3 Identificar la clave a utilizar para la firma	11
4.4 Firmar los datos de la petición	12
4.5 Utilización de librerías de ayuda	12
4.5.1 Librería PHP	12
4.5.2 Librería JAVA	14
<b>5. Recepción de la notificación on-line</b>	<b>16</b>
5.1 Notificación Síncrona y Asíncrona	16
5.1.1 Librería PHP	16
5.1.2 Librería JAVA	18
5.2 Notificación SOAP	19
5.2.1 Librería PHP	19
5.2.2 Librería JAVA	21
<b>6. Retorno del control de la navegación</b>	<b>23</b>
6.1 Utilización de librerías de ayuda	23
6.1.1 Librería PHP	23
6.1.2 Librería JAVA	25
<b>7. Entorno de pruebas</b>	<b>27</b>
<b>8. Códigos de error asociados</b>	<b>29</b>

<b>9. ANEXOS.....</b>	<b>33</b>
9.1 Datos de la solicitud de pago .....	33
9.2 Datos de la notificación on-line.....	35
9.3 Notificación SOAP .....	38

## **1. Introducción**

---

### **1.1 Objetivo**

Este documento recoge los aspectos técnicos necesarios para que un comercio, que actualmente esté operando en el SIS, realice la migración con el TPV Virtual utilizando el nuevo sistema de firma basado en HMAC SHA256.

El algoritmo actual (SHA-1) en el que se basa la seguridad de la conexión de la tienda con el tpv virtual (y viceversa) es un algoritmo obsoleto según los estándares de seguridad, y podría ser objeto de ataques. Para garantizar la mayor seguridad en la conexión con el TPV Virtual SIS los métodos anteriores deben actualizarse al nuevo sistema de firma basado en HMAC SHA256.

Esta documentación aplica a los comercios que acceden al SIS por redirección del navegador del cliente/comprador, (entrada "Realizar Pago") y también afecta a las notificaciones desde el TPV Virtual hacia el servidor del comercio para comunicación de los resultados (modalidades "Síncrona" o "Asíncrona")

Para desarrollar el cálculo de este nuevo tipo de firma, el comercio puede realizar el desarrollo por sí mismo utilizando las funciones estándar o utilizar las librerías facilitadas (PHP y JAVA).

## 2. **Resumen de las diferencias del nuevo modelo (HMAC SHA-256) respecto del modelo anterior (SHA-1)**

---

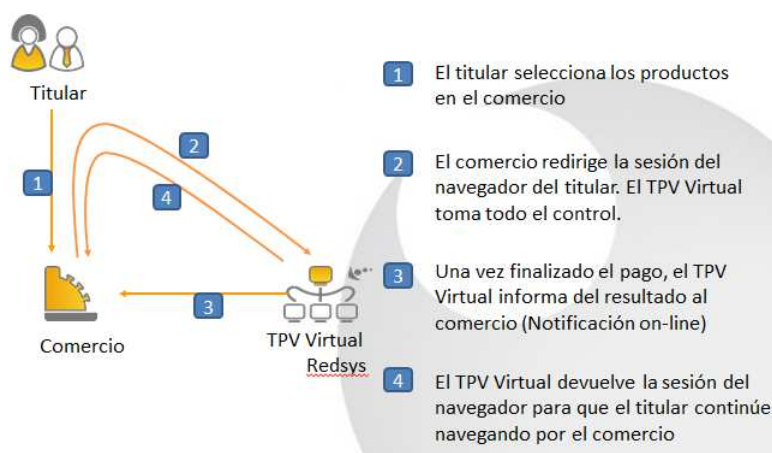
Las diferencias del nuevo modelo de conexión basado en HMAC SHA-256 respecto al modelo anterior en uso (basado en SHA-1) se pueden resumir en los siguientes 3 puntos:

- A. **Conexión desde la tienda web hacia el TPV Virtual** para iniciar de una operación:
1. Formato de los parámetros enviados.
    - **ANTES:** En el modelo anterior los parámetros eran enviados como campos independientes de un formulario.
    - **AHORA:** Los parámetros se agrupan en formato JSON y se envían como un único campo en el formulario.
  2. Cálculo de la firma enviada por la tienda web.
    - **ANTES:** La firma se calculaba como un SHA-1 sobre la concatenación de los campos enviados. No se enviaba ningún parámetro indicador de la versión de firma utilizada.
    - **AHORA:** La firma se calcula con una nueva clave diversifica por operación, y un nuevo algoritmo (HMAC SHA256). Se firma la lista de parámetros enviados en formato JSON. Además se incluye un campo nuevo que indica la versión de firma utilizada.
- B. **Retorno de la navegación del cliente desde el TPV virtual a la web de la tienda.** Los cambios son los mismos descritos en el punto A).
- C. **Envío de notificaciones on-line desde el TPV virtual** al servidor de la tienda web.
1. Comercios con notificaciones HTTP (tanto asíncrona como síncrona): Los cambios son los mismos descritos en el punto A).
  2. Comercios con notificaciones SOAP:
    - **ANTES:** La cadena de campos del mensaje enviado se firma con algoritmo basado en SHA-1.
    - **AHORA:** La cadena de campos del mensaje enviado se firma con la nueva clave y utilizando un nuevo algoritmo (HMAC SHA256).
- D. **Los comercios que utilizan la entrada XML-Entidad deben utilizar la entrada "realizarPago".**

**NOTA: No es necesario modificar ningún parámetro dentro del módulo de administración del comercio. El TPV Virtual aceptará de forma automática las conexiones basadas en HMAC SHA-256, una vez que la tienda online empiece a enviarlas al TPV Virtual. De igual forma, el TPV Virtual utilizará el formato HMAC SHA-256 para confirmar dichas operaciones al servidor de la tienda (notificaciones y retorno de la navegación del cliente).**

### 3. Descripción general del flujo

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



#### 3.1 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). Entre los parámetros que se pueden enviar ya no tiene cabida el parámetro "Ds\_Merchant\_MerchantSignature", como se puede comprobar en el apartado Anexos.
- **Ds\_Signature:** Firma de los datos enviados. Resultado del HMAC SHA256 de la cadena JSON codificada en Base 64 enviada en el parámetro anterior. Este nuevo parámetro es la firma del comercio en sustitución del parámetro "Ds\_Merchant\_MerchantSignature" que se enviaba anteriormente.

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

URL Conexión	Entorno
<a href="https://sis-t.redsys.es:25443/sis/realizarPago">https://sis-t.redsys.es:25443/sis/realizarPago</a>	Pruebas
<a href="https://sis.redsys.es/sis/realizarPago">https://sis.redsys.es/sis/realizarPago</a>	Real

### 3.2 Recepción del resultado (Notificación on-line)

Una vez gestionado el pago, 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 para su recepción debe estar configurada 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). Entre los parámetros que forman parte de este campo ya no tiene cabida el parámetro "Ds\_Signature" que se utilizaba anteriormente como firma del comercio, como se puede observar en Anexos.
- **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.**

### 3.3 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 cliente. 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.

## 4. 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.

En el antiguo acceso (basado en SHA-1) los parámetros se enviaban de forma individual, tal y como se muestran en los siguientes ejemplos:

Ejemplo de petición de pago **sin envío** de datos de tarjeta:

```
<form name="compra" action="https://sis-t.redsys.es:25443/sis/realizarPago"
method="POST">
    <input type="hidden" name="Ds_Merchant_Amount" value="100" />
    <input type="hidden" name="Ds_Merchant_Currency" value="978" />
    <input type="hidden" name="Ds_Merchant_Order" value="361134" />
    <input type="hidden" name="Ds_Merchant_MerchantCode" value="327234688" />
    <input type="hidden" name="Ds_Merchant_MerchantSignature"
value="A179F1416E44D5FDE3FEA0D71961C7BFBBB05263" />
    <input type="hidden" name="Ds_Merchant_Terminal" value="1" />
    <input type="hidden" name="Ds_Merchant_TransactionType" value="0" />
    <input type="hidden" name="Ds_Merchant_MerchantURL"
value="http://www.bancsabadell.com/urlNotificacion.php" />
    <input type="hidden" name="Ds_Merchant_UrlOK"
value="http://www.bancsabadell.com/urlOK.php" />
    <input type="hidden" name="Ds_Merchant_UrlKO"
value="http://www.bancsabadell.com/urlKO.php" />
</form>
```

Ejemplo de petición de pago **con envío** de datos de tarjeta (XMLEntidad):

```
<form name="compra" action="https://sis-t.redsys.es:25443/sis/entradaXMLEntidad"
method="post">
<input type="HIDDEN" name="entrada" value='
    <DATOSENTRADA><DS_Version>0.1</DS_Version>
    <DS_MERCHANT_AMOUNT>100</DS_MERCHANT_AMOUNT>
    <DS_MERCHANT_CURRENCY>978</DS_MERCHANT_CURRENCY>
    <DS_MERCHANT_ORDER>660982</DS_MERCHANT_ORDER>
    <DS_MERCHANT_MERCHANTCODE>327234688</DS_MERCHANT_MERCHANTCODE>
    <DS_MERCHANT_MERCHANTURL>http://www.bancsabadell.com/urlNotificacion.php</
DS_MERCHANT_MERCHANTURL>
    <DS_MERCHANT_TERMINAL>1</DS_MERCHANT_TERMINAL>
    <DS_MERCHANT_TRANSACTIONTYPE>A</DS_MERCHANT_TRANSACTIONTYPE>
    <DS_MERCHANT_PAN>4548812049400004</DS_MERCHANT_PAN>
    <DS_MERCHANT_EXPIRYDATE>2012</DS_MERCHANT_EXPIRYDATE>
    <DS_MERCHANT_CVV2>123</DS_MERCHANT_CVV2>
    <DS_MERCHANT_MERCHANTSIGNATURE>71787263B6D37C841668DDEAA1152C2DFF
FFBE5C</DS_MERCHANT_MERCHANTSIGNATURE>
    <DS_MERCHANT_URLLOK>http://www.bancsabadell.com/urlOK.php</DS_MERCHANT_
URLLOK>
    <DS_MERCHANT_URLKO>http://www.bancsabadell.com/urlKO.php</DS_MERCHANT_
URLKO>
```



```
</DATOSENTRADA>' />
```

```
</form>
```

A continuación se muestra un ejemplo del formulario de la nueva petición de pago (se puede observar que los campos se envían agrupados en formato JSON y empaquetados en Base64sin retornos de carro):

Ejemplo de formulario de pago **sin envío** de datos de tarjeta:

```
<form name="from" action="https://sis-t.redsys.es:25443/sis/realizarPago" method="POST">
  <input type="hidden" name="Ds_SignatureVersion" value="HMAC_SHA256_V1"/>
  <input type="hidden" name="Ds_MerchantParameters"
    value="eyJEU19NRVJDSEFOVF9BTU9VTlQioiIxNDUiLCJEU19NRVJDSEFOVF9PUkRFUuiI6Ij
    E0NDYxMTc1NTUiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUioiIzMjcyMzQ2ODgiLCJE
    U19NRVJDSEFOVF9DdVJJSRU5DWSiI6Ijk3OCIsIkRTX01FUKNIQU5UX1RSQU5TQUNUSU9
    OVFIQRSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCiI6IjEiLCJEU19NRVJDSEFOVF9NRVJ
    DSEFOVFVSTCI6Imh0dHA6XC9cL3d3dy5iYW5jc2FiYWRLbGwuY29tXC91cmxOb3RpZmljY
    WNpb24ucGhwIiwifRNFtUVSQ0hBTIRfVJMT0siOiJodHRwOlwvXC93d3cuYmFuY3NhYmF
    kZWxsLmNvbVwvdXJsT0sucGhwIiwifRNFtUVSQ0hBTIRfVJMS08iOiJodHRwOlwvXC93d3
    cuYmFuY3NhYmFkZWxsLmNvbVwvdXJsS08ucGhwIn0=" />
  <input type="hidden" name="Ds_Signature"
    value="QfLVUv4nF2Nw7jBAkw0w8H0eRlwh2E1w/ZIKHdA2Sq0=" />
</form>
```

Ejemplo de formulario de pago **con envío** de datos de tarjeta:

```
<form name="from" action="https://sis-t.redsys.es:25443/sis/realizarPago" method="POST">
  <input type="hidden" name="Ds_SignatureVersion" value="HMAC_SHA256_V1"/>
  <input type="hidden" name="Ds_MerchantParameters"
    value="eyJEU19NRVJDSEFOVF9BTU9VTlQioiIxNDUiLCJEU19NRVJDSEFOVF9PUkRFUuiI6I
    jE0NDYwNjg1ODEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUioiIzMjcyMzQ2ODgiLCJ
    EU19NRVJDSEFOVF9DdVJJSRU5DWSiI6Ijk3OCIsIkRTX01FUKNIQU5UX1RSQU5TQUNUSU
    9OVFIQRSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCiI6IjEiLCJEU19NRVJDSEFOVF9NR
    VJDSEFOVFVSTCI6Imh0dHA6XC9cL3d3dy5iYW5jc2FiYWRLbGwuY29tXC91cmxOb3RpZm
    ljYWVpb24ucGhwIiwifRNFtUVSQ0hBTIRfVJMT0siOiJodHRwOlwvXC93d3cuYmFuY3NhY
    mFkZWxsLmNvbVwvdXJsT0sucGhwIiwifRNFtUVSQ0hBTIRfVJMS08iOiJodHRwOlwvXC9
    3d3cuYmFuY3NhYmFkZWxsLmNvbVwvdXJsS08ucGhwIiwifRNFtUVSQ0hBTIRfUEFOIjo
    iNFTU0ODgxMjA0OTQwMDAwNCIsIkRTX01FUKNIQU5UX0VYUeISWURBVEUioiIjNTEyIiwif
    RNFtUVSQ0hBTIRfQ1ZWMiI6IjEjEjYmYj9" />
  <input type="hidden" name="Ds_Signature"
    value="1KyJDBDUoD/62iK45MGFWcaB2WOPXcmqJrYOfzSx8hg=" />
</form>
```

Para facilitar la adaptación del comercio al nuevo tipo de firma, a continuación se explica de forma detallada los pasos a seguir para montar el formulario de petición de pago.

## 4.1 Identificar la versión de algoritmo de firma a utilizar

En el modelo de conexión antiguo (SHA-1) no se incluía ningún parámetro identificando la versión de firma. En el nuevo formato, 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.

## 4.2 Montar la cadena de datos de la petición

Se debe montar una cadena con todos los datos de la antigua petición de pago 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:

Ejemplo **sin envío** de datos de tarjeta:

```
{ "DS_MERCHANT_AMOUNT": "145", "DS_MERCHANT_ORDER": "1446117555", "DS_MERCHANT_MERCHANTCODE": "327234688", "DS_MERCHANT_CURRENCY": "978", "DS_MERCHANT_TRANSACTIONTYPE": "0", "DS_MERCHANT_TERMINAL": "1", "DS_MERCHANT_MERCHANTURL": "http://www.bancsabadell.com/UrlNotificacion.php", "DS_MERCHANT_URLOK": "http://www.bancsabadell.com/UrlOK.php", "DS_MERCHANT_URLKO": "http://www.bancsabadell.com/UrlKO.php" }
```

Ejemplo **con envío** de datos de tarjeta:

```
{ "DS_MERCHANT_AMOUNT": "145", "DS_MERCHANT_ORDER": "1446068581", "DS_MERCHANT_MERCHANTCODE": "327234688", "DS_MERCHANT_CURRENCY": "978", "DS_MERCHANT_TRANSACTIONTYPE": "0", "DS_MERCHANT_TERMINAL": "1", "DS_MERCHANT_MERCHANTURL": "http://www.bancsabadell.com/UrlNotificacion.php", "DS_MERCHANT_URLOK": "http://www.bancsabadell.com/UrlOK.php", "DS_MERCHANT_URLKO": "http://www.bancsabadell.com/UrlKO.php", "DS_MERCHANT_PAN": "4548812049400004", "DS_MERCHANT_EXPIRYDATE": "1512", "DS_MERCHANT_CVV2": "123" }
```

Como se puede observar, el conjunto de parámetros que forma el objeto JSON debe ser exactamente igual que el conjunto de parámetros que se enviaba de manera individual por el antiguo acceso, a excepción del parámetro Ds Merchant MerchantSignature que no forma parte del objeto JSON, tal y como se ha comentado con anterioridad.

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 titular.

A continuación se muestran los objetos JSON que se acaba de mostrar codificado en BASE64:

Ejemplo JSON codificado **sin envío** de datos de tarjeta:

```
eyJEU19NRVJDSEFOVF9BTU9VTIQiOiIxNDUiLCJEU19NRVJDSEFOVF9PUKRfUii6IjE0NDYxMTc1NTUiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiIzMjcyMzQ2ODgiLCJEU19NRVJDSEFOVF9DVVJSRU5DWSI6Ijk3OCIsIkRTX01FUKNIQU5UX1RSQU5TQUNUSU9OVFIQRSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCi6IjEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVFVSTCI6Imh0dHA6XC9cL3d3dy5iYW5jc2FiYWRIbGwuY29tXC91cmxOb3RpZmJjYWNpb24ucGhwIiwifnFTUUVSQ0hBTIRfVVJMT0siOiJodHRwOlwvXC93d3cuYmFuY3NhYmFkZWxsLmNvbVwvdXJsT0sucGhwIiwifnFTUUVSQ0hBTIRfVVJMS08iOiJodHRwOlwvXC93d3cuYmFuY3NhYmFkZWxsLmNvbVwvdXJsS08ucGhwIn0=
```

Ejemplo JSON codificado **con envío** de datos de tarjeta:

```
eyJEU19NRVJDSEFOVF9BTU9VTIQiOiIxNDUiLCJEU19NRVJDSEFOVF9PUKRfUii6IjE0NDYwNjg1ODEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiIzMjcyMzQ2ODgiLCJEU19NRVJDSEFOVF9DVVJSRU5DWSI6Ijk3OCIsIkRTX01FUKNIQU5UX1RSQU5TQUNUSU9OVFIQRSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCi6IjEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVFVSTCI6Imh0dHA6XC9cL3d3dy5iYW5jc2FiYWRIbGwuY29tXC91cmxOb3RpZmJjYWNpb24ucGhwIiwifnFTUUVSQ0hBTIRfVVJMT0siOiJodHRwOlwvXC93d3cuYmFuY3NhYmFkZWxsLmNvbVwvdXJsT0sucGhwIiwifnFTUUVSQ0hBTIRfVVJMS08iOiJodHRwOlwvXC93d3cuYmFuY3NhYmFkZWxsLmNvbVwvdXJsS08ucGhwIiwifnFTUUVSQ0hBTIRfUeFOIjoINDU0ODgxMjA0OTQwMDAwNCIsIkRTX01FUKNIQU5UX0VYUEISWURBVEUiOiIxNTEyIiwifnFTUUVSQ0hBTIRfQ1ZWMiI6IjEyMyJ9
```

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.

### 4.3 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.

## 4.4 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, la cual debe ser previamente decodificada en BASE 64, 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.

## 4.5 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.

### 4.5.1 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("../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:

Ejemplo de parámetros **sin envío** de datos de tarjeta:

```
$miObj->setParameter("DS_MERCHANT_AMOUNT",$importe);
$miObj->setParameter("DS_MERCHANT_ORDER",strval($numPedido));
$miObj->setParameter("DS_MERCHANT_MERCHANTCODE",$merchantCode);
$miObj->setParameter("DS_MERCHANT_CURRENCY",$moneda);
$miObj->setParameter("DS_MERCHANT_TRANSACTIONTYPE",$transactionType);
$miObj->setParameter("DS_MERCHANT_TERMINAL",$terminal);
$miObj->setParameter("DS_MERCHANT_MERCHANTURL",$merchantURL);
$miObj->setParameter("DS_MERCHANT_URLLOK",$urLOK);
$miObj->setParameter("DS_MERCHANT_URLKO",$urlKO);
```

Ejemplo de parámetros **con envío** de datos de tarjeta:

```
$miObj->setParameter("DS_MERCHANT_AMOUNT",$importe);
$miObj->setParameter("DS_MERCHANT_ORDER",strval($numPedido));
$miObj->setParameter("DS_MERCHANT_MERCHANTCODE",$merchantCode);
$miObj->setParameter("DS_MERCHANT_CURRENCY",$moneda);
$miObj->setParameter("DS_MERCHANT_TRANSACTIONTYPE",$transactionType);
$miObj->setParameter("DS_MERCHANT_TERMINAL",$terminal);
$miObj->setParameter("DS_MERCHANT_MERCHANTURL",$merchantURL);
$miObj->setParameter("DS_MERCHANT_URLLOK",$urLOK);
$miObj->setParameter("DS_MERCHANT_URLKO",$urlKO);
$miObj->setParameter("DS_MERCHANT_PAN",$numTarjeta);
$miObj->setParameter("DS_MERCHANT_EXPIRYDATE",$expiryDate);
$miObj->setParameter("DS_MERCHANT_CVV2",$cvv2);
```

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:

```
$clave = 'sq7HjrUOBfKmC576ILgskD5srU870gJ7';
$signature = $miObj->createMerchantSignature($clave);
```

- 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 name="from" action="https://sis-t.redsys.es:25443/sis/realizarPago"
method="POST" target="_blank">

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

</form>
```

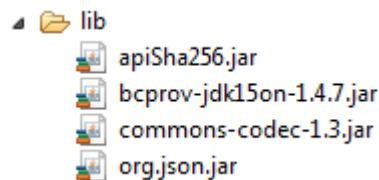
#### 4.5.2 Librería JAVA

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

- 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:



- 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:

Ejemplo de parámetros **sin envío** de datos de tarjeta:

```
ApiMacSha256.setParameter("DS_MERCHANT_AMOUNT", importe);
ApiMacSha256.setParameter("DS_MERCHANT_ORDER", numPedido);
ApiMacSha256.setParameter("DS_MERCHANT_MERCHANTCODE", merchantCode);
ApiMacSha256.setParameter("DS_MERCHANT_CURRENCY", moneda);
ApiMacSha256.setParameter("DS_MERCHANT_TRANSACTIONTYPE", transactionType);
ApiMacSha256.setParameter("DS_MERCHANT_TERMINAL", terminal);
ApiMacSha256.setParameter("DS_MERCHANT_MERCHANTURL", merchantURL);
ApiMacSha256.setParameter("DS_MERCHANT_URLOK", urlOK);
ApiMacSha256.setParameter("DS_MERCHANT_URLKO", urlKO);
```

Ejemplo de parámetros **con envío** de datos de tarjeta:

```
ApiMacSha256.setParameter("DS_MERCHANT_AMOUNT", importe);
ApiMacSha256.setParameter("DS_MERCHANT_ORDER", numPedido);
ApiMacSha256.setParameter("DS_MERCHANT_MERCHANTCODE", merchantCode);
ApiMacSha256.setParameter("DS_MERCHANT_CURRENCY", moneda);
ApiMacSha256.setParameter("DS_MERCHANT_TRANSACTIONTYPE", transactionType);
ApiMacSha256.setParameter("DS_MERCHANT_TERMINAL", terminal);
ApiMacSha256.setParameter("DS_MERCHANT_MERCHANTURL", merchantURL);
ApiMacSha256.setParameter("DS_MERCHANT_URLOK", urlOK);
ApiMacSha256.setParameter("DS_MERCHANT_URLKO", urlKO);
ApiMacSha256.setParameter("DS_MERCHANT_PAN", numTarjeta);
ApiMacSha256.setParameter("DS_MERCHANT_EXPIRYDATE", expiryDate);
ApiMacSha256.setParameter("DS_MERCHANT_CVV2", cvv2);
```

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 clave = "sq7HjrUObfKmC576ILgskD5srU870gJ7";
String signature = ApiMacSha256.createMerchantSignature(clave);
```

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>
```

## 5. 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.

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

### 5.1 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.

#### 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("../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:

```
$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: 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:

```
$clave = 'sq7HjrUOBfKmC576ILgskD5srU870gJ7';  
$signatureCalculada = $miObj->createMerchantSignatureNotif($clave,$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)  
{  
    //FIRMA OK. Realizar tareas de servidor.  
}  
else  
{  
    //FIRMA KO. Error, firma inválida.  
}
```

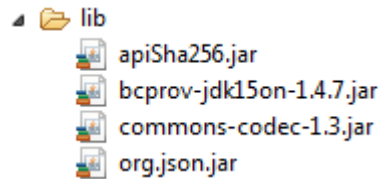
## 5.1.2 Librería JAVA

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 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 clave = "sq7HjrU0BfKmC576ILgskD5srU870gJ7";  
String signatureCalculada = ApiMacSha256.createMerchantSignatureNotif(clave, 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");  
}
```

## 5.2 Notificación SOAP

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.

### 5.2.1 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("../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)
{
    $clave = "sq7HjrUOBfKmc576ILgskD5srU870gJ7";
    $signatureCalculada = $miObj->createMerchantSignatureNotifSOAPRequest($clave, $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)
{
    //FIRMA OK. Realizar tareas de servidor.
}
else
{
    //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.

- 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:

```
$numPed = $miObj->getOrderNotifSOAP($XML);
$response='<Response Ds_Version="0.0">
    <Ds_Reponse_Merchant>OK</Ds_Reponse_Merchant>
</Response';
$clave = 'sq7HjrUOBfKmc576ILgskD5srU870gJ7';
$firmaRespuesta = $miObj->createMerchantSignatureNotifSOAPResponse($clave, $response, $numPed);
```

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.

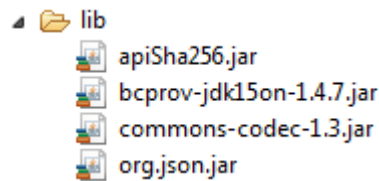
## 5.2.2 Librería JAVA

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 clave = "sq7HjrU0BfKmC576ILgskD5srU870gJ7";  
String signatureCalculada = ApiMacSha256.createMerchantSignatureNotifSOAPRequest(clave, 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 response = "<Response Ds_Version=\"0.0\">
                  <Ds_Reponse_Merchant>OK</Ds_Reponse_Merchant>
                </Response\";

String clave = "sq7HjrU0BfKmC576ILgskD5srU870gJ7";
String signatureCalculada = ApiMacSha256.createMerchantSignatureNotifSOAPRequest(clave, 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.

## 6. 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.

### 6.1 Utilización de librerías de ayuda

Una vez expuesta la nueva forma de acceso al SIS utilizando el sistema de firma basado en HMAC SHA256, este subapartado explica cómo se utilizan las librerías PHP y JAVA para la recepción de los parámetros del retorno de control de navegación.

#### 6.1.1 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("../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 = $miObj->decodeMerchantParameters($datos);
```

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:

```
$clave = 'sq7HjrU0BfKmc576ILgskD5srU870gJ7';  
$signatureCalculada = $miObj->createMerchantSignatureNotif($clave,$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)  
{  
    //FIRMA OK. Realizar tareas de servidor.  
}  
else  
{  
    //FIRMA KO. Error, firma inválida.  
}
```



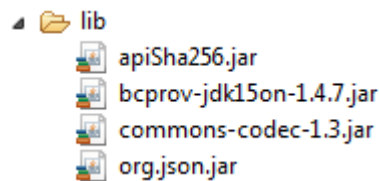
## 6.1.2 Librería JAVA

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 el 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 clave = "sq7HjrU0BfKmc576ILgskD5srU870gJ7";  
String signatureCalculada = ApiMacSha256.createMerchantSignatureNotif(clave, 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");  
}
```

## 7. Entorno de pruebas

---

El entorno de pruebas permite realizar las pruebas necesarias para verificar el correcto funcionamiento del sistema antes de la utilización en real del TPV Virtual del comercio. Dicho entorno es idéntico al real, pero sin que los pagos realizados tengan una validez contable.

Las claves del entorno de pruebas que le facilitamos a continuación son comunes para otros clientes de Banco Sabadell. También puede utilizar el entorno de pruebas de su comercio para realizar todas las pruebas que necesite. En caso de que no disponga de los datos de configuración, comuníquese con el Servicio Técnico en el teléfono 902 365 650 o el buzón [tpvvirtual@bancsabadell.com](mailto:tpvvirtual@bancsabadell.com).

Los parámetros del entorno de prueba son los que se describen a continuación.

### 1. URL para el envío de las órdenes de pago:

Entrada "realizarPago" (HTML):

<https://sis-t.redsys.es:25443/sis/realizarPago>

### 2. Número de comercio (Ds\_Merchant\_MerchantCode): 327234688

### 3. Clave secreta: qwertyasdf0123456789

### 4. Terminales (Ds\_Merchant\_Terminal):

- Terminal 001 - Para pagos en EUROS (Ds\_MerchantCurrency = 978) de comercios bajo protocolo CES (Comercio Electrónico Seguro -VERIFIED BY Visa y MasterCard SecureCode-)
- Terminal 002 - Para pagos en EUROS (Ds\_MerchantCurrency = 978) de comercios bajo protocolo No-CES (pagos considerados NO seguros)

### 5. Tarjeta aceptada:

- Numeración: 4548 8120 4940 0004
- Caducidad 12/17
- Código CVV2: 123.

En modo de compra segura (CES), en la que se requiera autenticación del comprador, el código de identificación personal (CIP) es: 123456

### 6. URL módulo de administración:

<https://sis-t.redsys.es:25443/canales/bsabadell>

## **7. Acceso al módulo de administración:**

- Para terminal 001 (CES):
  - Usuario: 327234688-001
  - Contraseña: 123456a
- Para terminal 002 (NO CES):
  - Usuario: 327234688-002
  - Contraseña: 123456a

## 8. Códigos de error asociados

Se han definido nuevos códigos de error propios de este nuevo acceso al SIS:

ERROR	DESCRIPCIÓN	MENSAJE (ANEXO VI)
SIS0007	Error al desmontar el XML de entrada o error producido al acceder mediante un sistema de firma antiguo teniendo configurado el tipo de clave HMAC SHA256	MSG0008
SIS0008	Error falta Ds_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	Error falta Ds_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

<b>ERROR</b>	<b>DESCRIPCIÓN</b>	<b>MENSAJE (ANEXO VI)</b>
<b>SIS0054</b>	Error no existe operación sobre la que realizar la devolución	MSG0008
<b>SIS0055</b>	Error existe más de un pago con el mismo número de pedido	MSG0008
<b>SIS0056</b>	La operación sobre la que se desea devolver no está autorizada	MSG0008
<b>SIS0057</b>	El importe a devolver supera el permitido	MSG0008
<b>SIS0058</b>	Inconsistencia de datos, en la validación de una confirmación	MSG0008
<b>SIS0059</b>	Error no existe operación sobre la que realizar la confirmación	MSG0008
<b>SIS0060</b>	Ya existe una confirmación asociada a la preautorización	MSG0008
<b>SIS0061</b>	La preautorización sobre la que se desea confirmar no está autorizada	MSG0008
<b>SIS0062</b>	El importe a confirmar supera el permitido	MSG0008
<b>SIS0063</b>	Error. Número de tarjeta no disponible	MSG0008
<b>SIS0064</b>	Error. El número de tarjeta no puede tener más de 19 posiciones	MSG0008
<b>SIS0065</b>	Error. El número de tarjeta no es numérico	MSG0008
<b>SIS0066</b>	Error. Mes de caducidad no disponible	MSG0008
<b>SIS0067</b>	Error. El mes de la caducidad no es numérico	MSG0008
<b>SIS0068</b>	Error. El mes de la caducidad no es válido	MSG0008
<b>SIS0069</b>	Error. Año de caducidad no disponible	MSG0008
<b>SIS0070</b>	Error. El Año de la caducidad no es numérico	MSG0008
<b>SIS0071</b>	Tarjeta caducada	MSG0000
<b>SIS0072</b>	Operación no anulable	MSG0000
<b>SIS0074</b>	Error falta Ds_Merchant_Order	MSG0008
<b>SIS0075</b>	Error el Ds_Merchant_Order tiene menos de 4 posiciones o más de 12	MSG0008
<b>SIS0076</b>	Error el Ds_Merchant_Order no tiene las cuatro primeras posiciones numéricas	MSG0008
<b>SIS0077</b>	Error el Ds_Merchant_Order no tiene las cuatro primeras posiciones numéricas. No se utiliza	MSG0000
<b>SIS0078</b>	Método de pago no disponible	MSG0005
<b>SIS0079</b>	Error al realizar el pago con tarjeta	MSG0000
<b>SIS0081</b>	La sesión es nueva, se han perdido los datos almacenados	MSG0007
<b>SIS0084</b>	El valor de Ds_Merchant_Conciliation es nulo	MSG0008
<b>SIS0085</b>	El valor de Ds_Merchant_Conciliation no es numérico	MSG0008
<b>SIS0086</b>	El valor de Ds_Merchant_Conciliation no ocupa 6 posiciones	MSG0008
<b>SIS0089</b>	El valor de Ds_Merchant_ExpiryDate no ocupa 4 posiciones	MSG0008
<b>SIS0092</b>	El valor de Ds_Merchant_ExpiryDate es nulo	MSG0008
<b>SIS0093</b>	Tarjeta no encontrada en la tabla de rangos	MSG0006
<b>SIS0094</b>	La tarjeta no fue autenticada como 3D Secure	MSG0004
<b>SIS0097</b>	Valor del campo Ds_Merchant_CComercio no válido	MSG0008
<b>SIS0098</b>	Valor del campo Ds_Merchant_CVentana no válido	MSG0008
<b>SIS0112</b>	Error El tipo de transacción especificado en Ds_Merchant_Transaction_Type no esta permitido	MSG0008
<b>SIS0113</b>	Excepción producida en el servlet de operaciones	MSG0008

<b>ERROR</b>	<b>DESCRIPCIÓN</b>	<b>MENSAJE (ANEXO VI)</b>
<b>SIS0114</b>	Error, se ha llamado con un GET en lugar de un POST	MSG0000
<b>SIS0115</b>	Error no existe operación sobre la que realizar el pago de la cuota	MSG0008
<b>SIS0116</b>	La operación sobre la que se desea pagar una cuota no es una operación válida	MSG0008
<b>SIS0117</b>	La operación sobre la que se desea pagar una cuota no está autorizada	MSG0008
<b>SIS0118</b>	Se ha excedido el importe total de las cuotas	MSG0008
<b>SIS0119</b>	Valor del campo Ds_Merchant_DateFrequency no válido	MSG0008
<b>SIS0120</b>	Valor del campo Ds_Merchant_ChargeExpiryDate no válido	MSG0008
<b>SIS0121</b>	Valor del campo Ds_Merchant_SumTotal no válido	MSG0008
<b>SIS0122</b>	Valor del campo Ds_Merchant_DateFrequency o no Ds_Merchant_SumTotal tiene formato incorrecto	MSG0008
<b>SIS0123</b>	Se ha excedido la fecha tope para realizar transacciones	MSG0008
<b>SIS0124</b>	No ha transcurrido la frecuencia mínima en un pago recurrente sucesivo	MSG0008
<b>SIS0132</b>	La fecha de Confirmación de Autorización no puede superar en más de 7 días a la de Preautorización.	MSG0008
<b>SIS0133</b>	La fecha de Confirmación de Autenticación no puede superar en más de 45 días a la de Autenticación Previa.	MSG0008
<b>SIS0139</b>	Error el pago recurrente inicial está duplicado	MSG0008
<b>SIS0142</b>	Tiempo excedido para el pago	MSG0000
<b>SIS0197</b>	Error al obtener los datos de cesta de la compra en operación tipo pasarela	MSG0000
<b>SIS0198</b>	Error el importe supera el límite permitido para el comercio	MSG0000
<b>SIS0199</b>	Error el número de operaciones supera el límite permitido para el comercio	MSG0008
<b>SIS0200</b>	Error el importe acumulado supera el límite permitido para el comercio	MSG0008
<b>SIS0214</b>	El comercio no admite devoluciones	MSG0008
<b>SIS0216</b>	Error Ds_Merchant_CVV2 tiene más de 3 posiciones	MSG0008
<b>SIS0217</b>	Error de formato en Ds_Merchant_CVV2	MSG0008
<b>SIS0218</b>	El comercio no permite operaciones seguras por la entrada /operaciones	MSG0008
<b>SIS0219</b>	Error el número de operaciones de la tarjeta supera el límite permitido para el comercio	MSG0008
<b>SIS0220</b>	Error el importe acumulado de la tarjeta supera el límite permitido para el comercio	MSG0008
<b>SIS0221</b>	Error el CVV2 es obligatorio	MSG0008
<b>SIS0222</b>	Ya existe una anulación asociada a la preautorización	MSG0008
<b>SIS0223</b>	La preautorización que se desea anular no está autorizada	MSG0008
<b>SIS0224</b>	El comercio no permite anulaciones por no tener firma ampliada	MSG0008
<b>SIS0225</b>	Error no existe operación sobre la que realizar la anulación	MSG0008
<b>SIS0226</b>	Inconsistencia de datos, en la validación de una anulación	MSG0008
<b>SIS0227</b>	Valor del campo Ds_Merchant_TransactionDate no válido	MSG0008
<b>SIS0229</b>	No existe el código de pago aplazado solicitado	MSG0008
<b>SIS0252</b>	El comercio no permite el envío de tarjeta	MSG0008
<b>SIS0253</b>	La tarjeta no cumple el check-digit	MSG0006
<b>SIS0254</b>	El número de operaciones de la IP supera el límite permitido por el comercio	MSG0008

<b>ERROR</b>	<b>DESCRIPCIÓN</b>	<b>MENSAJE (ANEXO VI)</b>
<b>SIS0255</b>	El importe acumulado por la IP supera el límite permitido por el comercio	MSG0008
<b>SIS0256</b>	El comercio no puede realizar preautorizaciones	MSG0008
<b>SIS0257</b>	Esta tarjeta no permite operativa de preautorizaciones	MSG0008
<b>SIS0258</b>	Inconsistencia de datos, en la validación de una confirmación	MSG0008
<b>SIS0261</b>	Operación detenida por superar el control de restricciones en la entrada al SIS	MSG0008
<b>SIS0270</b>	El comercio no puede realizar autorizaciones en diferido	MSG0008
<b>SIS0274</b>	Tipo de operación desconocida o no permitida por esta entrada al SIS	MSG0008
<b>SIS0429</b>	Error en la versión enviada por el comercio en el parámetro Ds_SignatureVersion	MSG0008
<b>SIS0430</b>	Error al decodificar el parámetro Ds_MerchantParameters	MSG0008
<b>SIS0431</b>	Error del objeto JSON que se envía codificado en el parámetro Ds_MerchantParameters	MSG0008
<b>SIS0432</b>	Error FUC del comercio erróneo	MSG0008
<b>SIS0433</b>	Error Terminal del comercio erróneo	MSG0008
<b>SIS0434</b>	Error ausencia de número de pedido en la operación enviada por el comercio	MSG0008
<b>SIS0435</b>	Error en el cálculo de la firma	MSG0008



## 9. ANEXOS

### 9.1 Datos de la solicitud de pago

En la petición de pago se tendrán que enviar una serie de datos obligatorios. El comercio también podrá incluir datos adicionales según las necesidades de su operativa concreta.

Los datos imprescindibles para la gestión de la autorización están marcados como obligatorios en la tabla siguiente.

DATO	NOMBRE DEL DATO	Long. / Tipo	COMENTARIOS
Identificación de comercio: código FUC	<i>Ds_Merchant_MerchantCode</i>	9/N.	<b>Obligatorio.</b> Código FUC asignado al comercio.
Número de terminal	<i>Ds_Merchant_Terminal</i>	3/N.	<b>Obligatorio.</b> Número de terminal que le asignará su banco. Tres se considera su longitud máxima
Tipo de transacción	<i>Ds_Merchant_TransactionType</i>	1/ Num	<b>Obligatorio.</b> para el comercio para indicar qué tipo de transacción es. Los posibles valores son: 0 - <b>Autorización</b> 1 - <b>Preautorización</b> 2 - <b>Confirmación de preautorización</b> 3 - <b>Devolución Automática</b> 5 - <b>Transacción Recurrente</b> 6 - <b>Transacción Sucesiva</b> 7 - <b>Pre-autenticación</b> 8 - <b>Confirmación de pre-autenticación</b> 9 - <b>Anulación de Preautorización</b> O - <b>Autorización en diferido</b> P- <b>Confirmación de autorización en diferido</b> Q - <b>Anulación de autorización en diferido</b> R - <b>Cuota inicial diferido</b> S- <b>Cuota sucesiva diferido</b>
Importe	<i>Ds_Merchant_Amount</i>	12/ Núm.	<b>Obligatorio.</b> Para Euros las dos últimas posiciones se consideran decimales.
Moneda	<i>Ds_Merchant_Currency</i>	4/ Núm.	<b>Obligatorio.</b> 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.	<b>Obligatorio.</b> Los 4 primeros dígitos deben ser numéricos, para los dígitos restantes solo utilizar los siguientes caracteres ASCII Del 30 = <b>0</b> al 39 = <b>9</b> Del 65 = <b>A</b> al 90 = <b>Z</b> Del 97 = <b>a</b> al 122 = <b>z</b>
URL del comercio para la notificación "on-line"	<i>Ds_Merchant_MerchantURL</i>	250/A-N	Obligatorio <b>si</b> 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_UrIKO</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.

## 9.2 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 envía al comercio un POST independiente de la conexión con el navegador del comprador, y no tiene ningún reflejo en pantalla del mismo. Evidentemente, en el lado del comercio, deberá haber un proceso que recoja esta respuesta. Para ello tendrá que facilitar 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.

**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	<i>Ds_Date</i>	<i>dd/mm/yyyy</i>	Fecha de la transacción
Hora	<i>Ds_Hour</i>	<i>HH:mm</i>	Hora de la transacción
Importe	<i>Ds_Amount</i>	<i>12 / Núm.</i>	Mismo valor que en la petición.
Moneda	<i>Ds_Currency</i>	<i>4 / Núm.</i>	Mismo valor que en la petición. 4 se considera su longitud máxima.
Número de pedido	<i>Ds_Order</i>	<i>12 / A-N.</i>	Mismo valor que en la petición.
Identificación de comercio: código FUC	<i>Ds_MerchantCode</i>	<i>9 / N.</i>	Mismo valor que en la petición.
Terminal	<i>Ds_Terminal</i>	<i>3 / Núm.</i>	Número de terminal que le asignará su banco. 3 se considera su longitud máxima.
Código de respuesta	<i>Ds_Response</i>	<i>4 / Núm.</i>	Ver tabla siguiente (Posibles valores del <i>Ds_Response</i> ).
Datos del comercio	<i>Ds_MerchantData</i>	<i>1024 / A-N</i>	Información opcional enviada por el comercio en el formulario de pago.
Pago Seguro	<i>Ds_SecurePayment</i>	<i>1 / Núm.</i>	0 – Si el pago <b>NO</b> es seguro 1 – Si el pago es seguro
Tipo de operación	<i>Ds_TransactionType</i>	<i>1 / A-N</i>	Tipo de operación que se envió en el formulario de pago
País del titular	<i>Ds_Card_Country</i>	<i>3/Núm</i>	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:  <a href="http://unstats.un.org/unsd/methods/m49/m49alpha.htm">http://unstats.un.org/unsd/methods/m49/m49alpha.htm</a>
Código de autorización	<i>Ds_AuthorisationCode</i>	<i>6/ A-N</i>	Opcional: Código alfanumérico de autorización asignado a la aprobación de la transacción por la institución


DATO	NOMBRE DEL DATO	LONG/TIPO	COMENTARIOS
			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**

<b>Sesión / Fecha Totales</b>	<b>Fecha Hora</b>	<b>Tipo Operación Num. Pedido</b>	<b>Resultado NºAutorización o Cod.Respuesta</b>	<b>Importe</b>	<b>Neto Lote/Cajón</b>
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 /

## 9.3 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)>
<!ELEMENTS Signature (#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>
```