

---

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

Versión: 2.3

12/05/2017

**RS.TE.CEL.MAN.0002**



Redsys · C/ Francisco Sancha, 12 · 28034 · Madrid · ESPAÑA

---

12/05/2017

## Autorizaciones y control de versión

Versión	Fecha	Afecta	Breve descripción del cambio
1.0	06/10/2015		Versión inicial del documento
1.1	29/10/2015		Se añade el detalle sobre la decodificación de la clave del comercio, previo al cálculo de la clave específica de la operación
1.2	04/11/2015		Se añade el código de anulación autorizada en la tabla de valores del Ds_Response
1.3	10/11/2015		Modificaciones del API Java
1.4	13/11/2015		Se añade todo lo relacionado con el API .NET
1.5	26/11/2015		Añadido punto 6 de entorno de pruebas
1.6	11/12/2015		Se añade información sobre el Pago por Referencia (Pago 1-Clic).
1.7	23/02/2016		Incorporación del error SIS0444
1.8	19/04/2016		Incorporación de los parámetros de la tarjeta en el apartado "Datos de la solicitud de pago"
1.9	23/05/2016		Incorporación de nuevos códigos de error en el "Glosario de errores"
2.0	30/05/2016		Modificación códigos de error
2.1	20/09/2016		Incorporación de nuevos códigos de error en el "Glosario de errores"
2.2	09/03/2017		Incorporación ejemplo wsdl estilo document/literal en notificación
2.3	12/05/2017		Se añade parámetro Ds_Card_Brand en respuestas/notificaciones

## ÍNDICE DE CONTENIDO

<b>1. Introducción .....</b>	<b>1</b>
1.1 Objetivo .....	1
1.2 Definiciones, siglas y abreviaturas .....	1
1.3 Referencias .....	1
<b>2. Descripción general del flujo .....</b>	<b>2</b>
2.1 Envío de petición al TPV Virtual .....	2
2.2 Recepción del resultado (Notificación on-line) .....	3
2.3 Retorno del control de la navegación del titular .....	3
<b>3. Formulario de envío de petición .....</b>	<b>4</b>
3.1 Identificar la versión de algoritmo de firma a utilizar .....	5
3.2 Montar la cadena de datos de la petición .....	5
3.3 Identificar la clave a utilizar para la firma .....	7
3.4 Firmar los datos de la petición .....	7
3.5 Utilización de librerías de ayuda .....	8
3.5.1 Librería PHP .....	8
3.5.2 Librería JAVA .....	10
3.5.3 Librería .NET .....	12
<b>4. Recepción de la notificación on-line .....</b>	<b>14</b>
4.1 Notificación Síncrona y Asíncrona .....	15
4.1.1 Librería PHP .....	15
4.1.2 Librería JAVA .....	17
4.1.3 Librería .NET .....	18
4.2 Notificación SOAP .....	20
4.2.1 Librería PHP .....	21
4.2.2 Librería JAVA .....	22
4.2.3 Librería .NET .....	24
<b>5. Retorno del control de la navegación .....</b>	<b>26</b>
5.1 Utilización de librerías de ayuda .....	26
5.1.1 Librería PHP .....	26

## Integración utilizando HMAC SHA256

5.1.2	Librería JAVA .....	28
5.1.3	Librería .NET .....	30
<b>6.</b>	<b>Entorno de pruebas .....</b>	<b>32</b>
<b>7.</b>	<b>Códigos de error.....</b>	<b>33</b>
7.1	Glosario de errores del SIS .....	34
<b>8.</b>	<b>ANEXOS.....</b>	<b>39</b>
8.1	Datos de la solicitud de pago .....	39
8.2	Datos de la notificación on-line.....	41
8.3	Notificación SOAP .....	45

## 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, JAVA y .NET) cuya utilización se presenta en detalle en esta guía y que están a su disposición en la siguiente dirección:**

[http://www.redsys.es/wps/portal/redsys/publica/areadeserviciosweb/de\\_scargaDeDocumentacionYEjecutables/](http://www.redsys.es/wps/portal/redsys/publica/areadeserviciosweb/de_scargaDeDocumentacionYEjecutables/)

### 1.2 Definiciones, siglas y abreviaturas

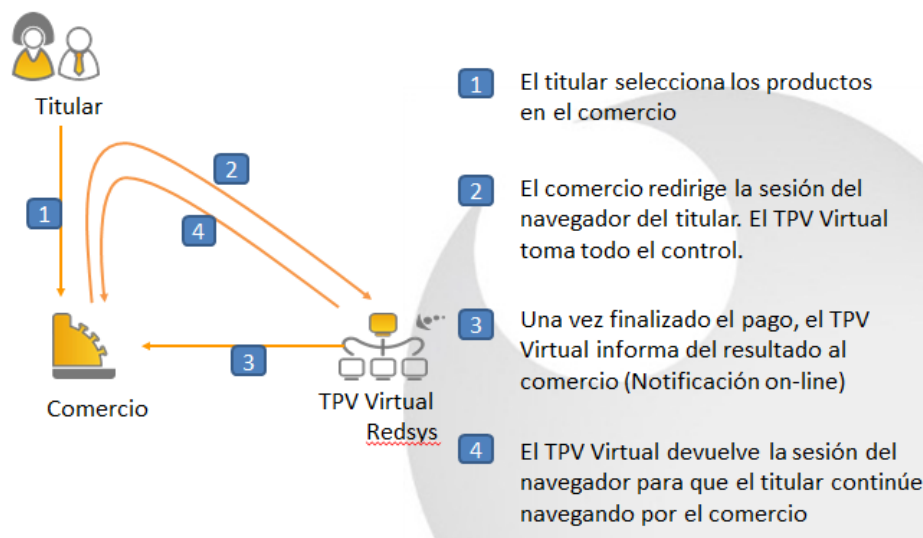
SIS. Servidor Integrado de Redsys (Servidor del TPV Virtual).

### 1.3 Referencias

- Documentación de Integración con el SIS
- Guía de comercios del SIS.

## 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 codificados en UTF-8 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
<b><a href="https://sis-t.redsys.es:25443/sis/realizarPago">https://sis-t.redsys.es:25443/sis/realizarPago</a></b>	Pruebas
<b><a href="https://sis.redsys.es/sis/realizarPago">https://sis.redsys.es/sis/realizarPago</a></b>	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 Módulo de Administración.

La notificación on-line consiste en un POST HTTP con la información del resultado codificada en UTF-8. 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 muestran diversos ejemplos del formulario de petición de pago:

#### 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="
eyJEU19NRVJDSEFOVF9BTU9VTIQiOiI5OTkiLCJEU19NRVJDSEFOVF9PukRFUiI6IjEYmZQ1
Njc4OTAiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NR
VJDSEFOVF9DVVJSRU5DWSi6Ijk3OCIsIkRTX01FukNIQU5UX1RSQU5TQUNUSU9OVFIQR
SI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCi6IjEiLCJEU19NRVJDSEFOVF9NRVJDSEFO
VVFVSTCI6Imh0dHA6XC9cL3d3dy5wcnVlYmEuY29tXC91cmxOb3RpZmljYWNPb24ucGhw
IiwiRfNFTUVSQ0hBTIRfVVJMT0siOiJodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsT0su
cGhwIiwiRfNFTUVSQ0hBTIRfVVJMS08iOiJodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJ
sS08ucGhwIn0="/>

  <input type="hidden" name="Ds_Signature"
value="PqV2+SF6asdasMjXasKJRTh3UIYya1hmU/igHkzhC+R="/>
</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="
eyJEU19NRVJDSEFOVF9BTU9VTIQiOiIxNDUiLCJEU19NRVJDSEFOVF9PukRFUiI6IjE0NDY
wNjg1ODEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NR
VJDSEFOVF9DVVJSRU5DWSi6Ijk3OCIsIkRTX01FukNIQU5UX1RSQU5TQUNUSU9OVFIQR
RSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCi6IjEiLCJEU19NRVJDSEFOVF9NRVJDSEF
OVVFVSTCI6Imh0dHA6XC9cL3d3dy5wcnVlYmEuY29tXC91cmxOb3RpZmljYWNPb24ucGhw
wIiwiRfNFTUVSQ0hBTIRfVVJMT0siOiJodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsT0
sucGhwIiwiRfNFTUVSQ0hBTIRfVVJMS08iOiJodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvd
XJsS08ucGhwIiwiRfNFTUVSQ0hBTIRfUEFOIjoIjNDU0ODgxMjA0OTQwMDAwNCIsIkRTX01
FukNIQU5UX0VYUeISWURBEUiOiIxNTEyIiwiRfNFTUVSQ0hBTIRfQ1ZWMiI6IjEYmZQ1"
/>

  <input type="hidden" name="Ds_Signature"
value="PqV2+SF6asdasMjXasKJRTh3UIYya1hmU/igHkzhC+R="/>
</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).

Los comercios que utilicen operativas especiales como el "Pago por referencia" (Pago 1-Clic), deberán incluir los parámetros específicos de su operativa como parte del objeto JSON.

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 muestran algunos ejemplos del objeto JSON de una petición:

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

```
{
  "DS_MERCHANT_AMOUNT": "145",
  "DS_MERCHANT_ORDER": "1446117555",
  "DS_MERCHANT_MERCHANTCODE": "999008881",
  "DS_MERCHANT_CURRENCY": "978",
  "DS_MERCHANT_TRANSACTIONTYPE": "0",
  "DS_MERCHANT_TERMINAL": "1",
  "DS_MERCHANT_MERCHANTURL": "http://www.prueba.com/UrlNotificacion.php",
  "DS_MERCHANT_URLOK": "http://www.prueba.com/UrlOK.php",
  "DS_MERCHANT_URLKO": "http://www.ban
csabadell.com/UrlKO.php"
}
```

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

```
{
  "DS_MERCHANT_AMOUNT": "145",
  "DS_MERCHANT_ORDER": "1446068581",
  "DS_MERCHANT_MERCHANTCODE": "999008881",
  "DS_MERCHANT_CURRENCY": "978",
  "DS_MERCHANT_TRANSACTIONTYPE": "0",
  "DS_MERCHANT_TERMINAL": "1",
  "DS_MERCHANT_MERCHANTURL": "http://www.prueba.com/UrlNotificacion.php",
  "DS_MERCHANT_URLOK": "http://www.prueba.com/UrlOK.php",
  "DS_MERCHANT_URLKO": "http://www.prueba.com/UrlKO.php",
  "DS_MERCHANT_PAN": "4548812049400004",
  "DS_MERCHANT_EXPIRYDATE": "1512",
  "DS_MERCHANT_CVV2": "123"
}
```

A continuación se muestran los objetos JSON que se acaban de mostrar codificados en BASE64:

eyJEU19NRVJDSEFOVF9BTU9VTIQiOiI5OTkiLCJEU19NRVJDSEFOVF9PUKRfUIi6IjEyMzQ1Njc4OTAiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9DVVJSRU5DWSi6Ijk3OCIsIkRXTX01FUKNIQU5UX1RSQU5TQUNUSU9OVFIQRSl6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCi6IjEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVFVSTCI6Imh0dHA6XC9cL3d3dy5wcnVlYmEuY29tXC91cmxOb3RpZmJlYWNPb24ucGhwIiwirFNFtUVSQ0hBTIRfVVJMT0siOiJodHRwOlwvXC93d3cucHJlZWJhLmNvbVwvdXJsT0su cGhwIiwirFNFtUVSQ0hBTIRfVVJMS08iOiJodHRwOlwvXC93d3cucHJlZWJhLmNvbVwvdXJsS08ucGhwInQ

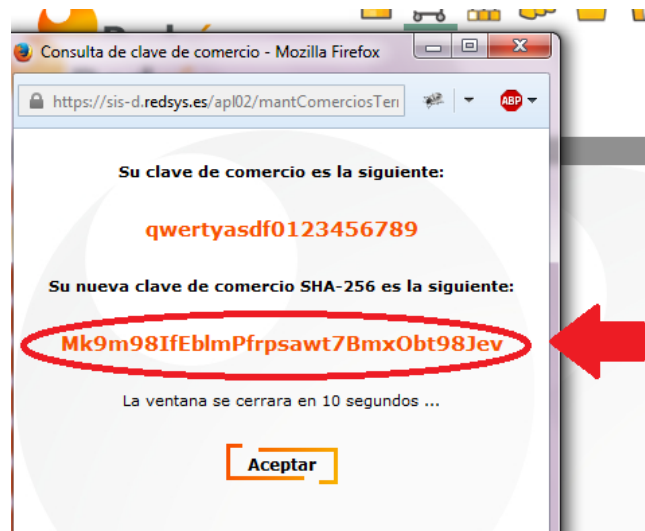
eyJEU19NRVJDSEFOVF9BTU9VTIQiOiIxNDUilCJEU19NRVJDSEFOVF9PUKRFuiI6IjE0NDYwNjg1ODEilCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEilCJEU19NRVJDSEFOVF9DVVJSRU5DWF5I6Ij3k3OCIsIkRTX01FUKNIQU5UX1RSQU5TQUNUSU9OVFIjRSI6IjAilCJEU19NRVJDSEFOVF9NRVJNSU5BTC16IjEilCJEU19NRVJDSEFOVF9NRVJDSEFOVFVSTC16Imh0dHA6XC9cL3d3dy5wcnVlYmEuY29tXC91cmxOb3R3pZmljYWNPb24ucGhwIiwIwIRnftUVVSQ0hBTIRfVJMT0siOiIodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsT0ScGhwIiwIiwIRnftUVVSQ0hBTIRfVJMS08iOiIodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsT0ScGhwIiwIiwIRnftUVVSQ0hBTIRfUEFOIjoiNDU0ODgxMjA0OTcwMDAwNCIsIkRTX01FUKNIQU5UX0VYUEIsWURBVEUiOiIxNTEyIiwIRnftUVVSQ0hBTIRfQ1ZWMi6IjEjYmY9

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 Redsys 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. Se puede obtener la clave accediendo al Módulo de Administración, opción Consulta datos de Comercio, en el apartado "Ver clave", tal y como se muestra en la siguiente imagen:



**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, 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.

**NOTA: La utilización de las librerías de ayuda proporcionadas por Redsys 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, JAVA y .NET para facilitar los desarrollos y la generación de los campos del formulario de pago. El uso de las librerías suministradas por Redsys 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 Redsys:

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:

## Integración utilizando HMAC SHA256

```
$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 obtenida del módulo de administración, 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>
```

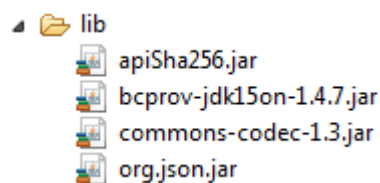
### 3.5.2 Librería JAVA

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

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

```
ApiMacSha256 apiMacSha256 = new ApiMacSha256();
```

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:

```
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();
```

## Integración utilizando HMAC SHA256

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 obtenida del módulo de administración, tal y como se muestra a continuación:

```
String claveModuloAdmin = "Mk9m98IfEblmPfrpsawt7Bmx0bt98Jev";
String signature = apiMacSha256.createMerchantSignature(claveModuloAdmin);
```

5. 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>
```



### 3.5.3 Librería .NET

A continuación se presentan los pasos que debe seguir un comercio para la utilización de la librería .NET proporcionada por Redsys:

1. Importar la librería RedsysAPI y Newronsoft.Json en su proyecto.
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:

```
// New instance of RedsysAPI
RedsysAPI r = new RedsysAPI();

// Fill Ds_MerchantParameters parameters
r.SetParameter("DS_MERCHANT_AMOUNT", amount);
r.SetParameter("DS_MERCHANT_ORDER", id);
r.SetParameter("DS_MERCHANT_MERCHANTCODE", fuc);
r.SetParameter("DS_MERCHANT_CURRENCY", currency);
r.SetParameter("DS_MERCHANT_TRANSACTIONTYPE", trans);
r.SetParameter("DS_MERCHANT_TERMINAL", terminal);
r.SetParameter("DS_MERCHANT_MERCHANTURL", url);
r.SetParameter("DS_MERCHANT_URLOK", urlOK);
r.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 parms = r.createMerchantParameters();
Ds_MerchantParameters.Value = parms;
```

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 obtenida del módulo de administración, tal y como se muestra a continuación:

```
string sig = r.createMerchantSignature(kc);
Ds_Signature.Value = sig;
```

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:



## Integración utilizando HMAC SHA256

```
<form action="https://sis-d.redsys.es:25443/sis/realizarPago"
method="post">
  <input runat="server" type="text" id="Ds_SignatureVersion"
    name="Ds_SignatureVersion" value="" />
  <input runat="server" size="100" type="text" id="Ds_MerchantParameters"
    name="Ds_MerchantParameters" value="" />
  <input runat="server" type="text" size="50" id="Ds_Signature"
    name="Ds_Signature" value="" />
  <input runat="server" 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 Redsys 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, JAVA y .NET 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 Redsys 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 Redsys:

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);
```

## Integración utilizando HMAC SHA256

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 obtenida del módulo de administración 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";
}
```

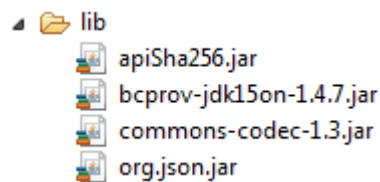
## 4.1.2 Librería JAVA

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

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

```
ApiMacSha256 apiMacSha256 = new ApiMacSha256();
```

3. 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");
```

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:

```
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.

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 obtenida del módulo de administración 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");
}
```

#### 4.1.3 Librería .NET

A continuación se presentan los pasos que debe seguir un comercio para la utilización de la librería .NET proporcionada por Redsys:

1. Importar la librería RedsysAPI y Newronsoft.Json en su proyecto.
2. Capturar los parámetros de la notificación on-line:

## Integración utilizando HMAC SHA256

```
// New instance of RedsysAPI
RedsysAPI r = new RedsysAPI();

// Obtain Ds_SignatureVersion using post
if (Request.Form["Ds_SignatureVersion"] != null)
{
    version = Request.Form["Ds_SignatureVersion"];
}

// Obtain Ds_MerchantParameters using post
if (Request.Form["Ds_MerchantParameters"] != null)
{
    data = Request.Form["Ds_MerchantParameters"];
}

// Obtain Ds_Signature using post
if (Request.Form["Ds_Signature"] != null)
{
    signatureReceived = Request.Form["Ds_Signature"];
}
```

6. Decodificar el parámetro **Ds\_MerchantParameters**. Para llevar a cabo la decodificación de este parámetro, se debe llamar a la función de la librería "decodeMerchantParameters()" que genera la cadena (tipo string) JSON de la respuesta, tal y como se muestra a continuación:

```
string deco = r.decodeMerchantParameters(data);
```

**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.

7. 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 obtenida del módulo de administración y el parámetro **Ds\_MerchantParameters** capturado, tal y como se muestra a continuación:

```
var kc = "Mk9m98IfEblmPfrpsawt7BmxObt98Jev";
string notif = r.createMerchantSignatureNotif(kc, data);
```

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:

```
string text = "";
if (notif.Equals(signatureReceived) && notif != "")
{
    text = "SIGNATURE OK";
}
else
{
    text = "SIGNATURE KO";
}
```

## 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 SincronizaciónSOAP 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, JAVA y .NET 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 Redsys 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 Redsys:

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 obtenida del módulo de administración 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.

## Integración utilizando HMAC SHA256

- 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 obtenida del módulo de administración, 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.

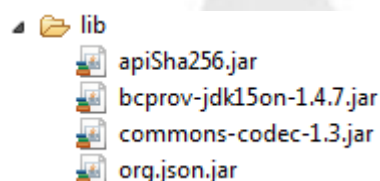
### 4.2.2 Librería JAVA

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

- 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:



## Integración utilizando HMAC SHA256

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

```
ApiMacSha256 apiMacSha256 = new ApiMacSha256();
```

- 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 obtenida del módulo de administración 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.

- 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 obtenida del módulo de administración, el valor del mensaje de respuesta y el número de pedido capturado, tal y como se muestra a continuación:

### Integración utilizando HMAC SHA256

```
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.

### 4.2.3 Librería .NET

A continuación se presentan los pasos que debe seguir un comercio para la utilización de la librería .NET proporcionada por Redsys:

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

```
RedsysAPISoap r = new RedsysAPISoap();
```

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 obtenida del módulo de administración y el valor del mensaje recibido en la notificación.

```
var kc = "sq7HjrU0BfKmC576ILgskD5srU870gJ7";
string signatureCalculate = r.createMerchantSignatureNotifSOAPRequest(kc, 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:

## Integración utilizando HMAC SHA256

```
signatureReceived = r.GetParameter(XML, "<Signature>", "</Signature>");

if (signatureCalculate == signatureReceived)
{
    res = "FIRMA OK";
}
else
{
    res = "FIRMA KO";
}
```

**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 obtenida del módulo de administración, el valor del mensaje de respuesta y el número de pedido capturado, tal y como se muestra a continuación:

```
string numOrder = r.GetOrderNotifSOAP(XML);
string respons = "<Response Ds_Version='0.0'><Ds_Response_Merchant>OK</Ds_Response_Merchant></Response>";
string signatureResponse = r.createSignatureNotifSOAPResponse(kc, respons, numOrder);
```

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 Redsys 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, JAVA y .NET 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 Redsys 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 Redsys:

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;
```

- Capturar los parámetros de la notificación on-line:

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

- 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:

```
$decoded = $miObj->decodeMerchantParameters($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.

- 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 obtenida del módulo de administración y el parámetro **Ds\_MerchantParameters** capturado, tal y como se muestra a continuación:

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

## Integración utilizando HMAC SHA256

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";
}
```

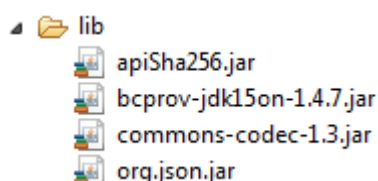
### 5.1.2 Librería JAVA

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

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

```
ApiMacSha256 apiMacSha256 = new ApiMacSha256();
```

3. 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");
```

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:

```
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.

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 obtenida del módulo de administración y el parámetro **Ds\_MerchantParameters** capturado, tal y como se muestra a continuación:

```
String claveModuloAdmin = "Mk9m98IfEblmPfrpsawt7BmxObt98Jev";
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");
}
```

### 5.1.3 Librería .NET

A continuación se presentan los pasos que debe seguir un comercio para la utilización de la librería .NET proporcionada por Redsys:

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

```
using RedsysAPIPrj;
```

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

```
RedsysAPI r = new RedsysAPI();
```

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

```
string version = Request.QueryString["Ds_SignatureVersion"];
string parms = Request.QueryString["Ds_MerchantParameters"];
string signatureRecibida = Request.QueryString["Ds_Signature"];
```

**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 obtenida del módulo de administración y el parámetro **Ds\_MerchantParameters** capturado, tal y como se muestra a continuación:

```
var kc = "sq7HjrU0BfKmc576ILgskD5srU870gJ7";

string signatureCalculada = r.createMerchantSignatureNotif(kc, parms);
```

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:

## Integración utilizando HMAC SHA256

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

## **6. Entorno de pruebas**

Existe un entorno de test que permite realizar las pruebas necesarias para verificar el correcto funcionamiento del sistema una vez realizada la integración, antes de hacer la implantación en el entorno real.

A continuación se proporcionarán las URL de acceso al portal de administración y la dirección del servicio para realizar las pruebas. Para obtener los datos de acceso, deberán dirigirse a su entidad bancaria para que ésta les proporcione los datos de acceso.

La URL para el envío de las órdenes de pago es la siguiente:

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

Adicionalmente, la URL para el acceso al módulo de administración es la siguiente:

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

\*El entorno de pruebas será idéntico al entorno real, con la única diferencia que los pagos realizados en este entorno no tendrán validez contable.

Desde Redsys se proporcionan unos datos genéricos de prueba para todos los clientes. Como ya se ha indicado, para obtener los datos de su comercio, deberá contactar con su entidad bancaria.

### **DATOS GENÉRICOS DE PRUEBA**

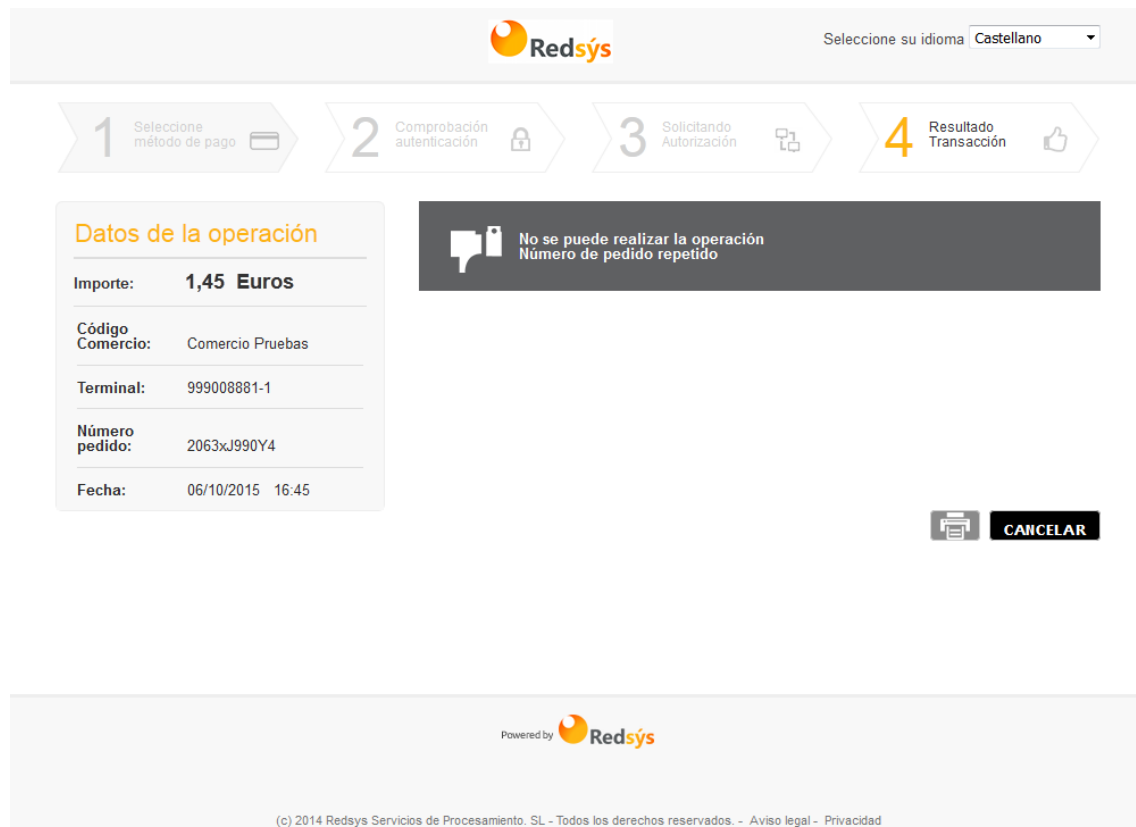
- Número de comercio (Ds\_Merchant\_MerchantCode): 999008881
- Terminal (Ds\_Merchant\_Terminal): 01
- Clave secreta: sq7HjrUOBfKmC576ILgskD5srU870gJ7
- Tarjeta aceptada:
  - Numeración: 4548 8120 4940 0004
  - Caducidad: 12/20
  - Código CVV2: 123
- Para compras seguras, en la que se requiere la autenticación del titular, el código de autenticación personal (CIP) es 123456.

## 7. 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




The screenshot shows the Redsys payment interface. At the top, there's a navigation bar with the Redsys logo and a language selector set to 'Castellano'. Below this is a progress bar with four steps: 1. Seleccione método de pago, 2. Comprobación autenticación, 3. Solicitando Autorización, and 4. Resultado Transacción. The fourth step is active. On the left, a box titled 'Datos de la operación' displays transaction details: Importe: 1,45 Euros; Código Comercio: Comercio Pruebas; Terminal: 999008881-1; Número pedido: 2063xJ990Y4; Fecha: 06/10/2015 16:45. On the right, a dark grey box with a speech bubble icon contains the error message: 'No se puede realizar la operación. Número de pedido repetido'. At the bottom right, there are buttons for 'Imprimir' and 'CANCELAR'. The footer includes the Redsys logo and copyright information: '(c) 2014 Redsys Servicios de Procesamiento. SL - Todos los derechos reservados. - Aviso legal - Privacidad'.

### 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>

```



## 7.1 Glosario de errores del SIS

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

ERROR	DESCRIPCIÓN	MENSAJE (ANEXO VI)
<b>SIS0046</b>	El bin de la tarjeta no está dado de alta	MSG0002
<b>SIS0051</b>	Error número de pedido repetido	MSG0001
<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

ERROR	DESCRIPCIÓN	MENSAJE (ANEXO VI)
<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>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



ERROR	DESCRIPCIÓN	MENSAJE (ANEXO VI)
<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>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
<b>SIS0444</b>	Error producido al acceder mediante un sistema de firma antiguo teniendo configurado el tipo de clave HMAC SHA256	MSG0008
<b>SIS0448</b>	Error, la tarjeta de la operación es DINERS y el comercio no tiene el método de pago "Pago DINERS"	MSG0008
<b>SIS0449</b>	Error, el tipo de pago de la operación Ds_TransactionType (A) no está permitido para el comercio.	MSG0008

## Integración utilizando HMAC SHA256

ERROR	DESCRIPCIÓN	MENSAJE (ANEXO VI)
<b>SIS0450</b>	Error, el tipo de pago de la operación Ds_TransactionType (A) no está permitido para el comercio para tarjetas Amex.	MSG0008
<b>SIS0452</b>	Método de pago no disponible (Tarjeta 4B)	MSG0008
<b>SIS0453</b>	Error, la tarjeta de la operación es JCB y el comercio no tiene el método de pago "Pago JCB"	MSG0008
<b>SIS0454</b>	Error, la tarjeta de la operación es AMEX y el comercio no tiene el método de pago "Pago Amex"	MSG0008
<b>SIS0455</b>	Método de pago no disponible	MSG0008
<b>SIS0456</b>	Método de pago no seguro (Visa) no disponible	MSG0008
<b>SIS0457</b>	Método de pago no seguro (MasterCard) no disponible	MSG0008
<b>SIS0458</b>	Método de pago no seguro (MasterCard) no disponible	MSG0008
<b>SIS0459</b>	Método de pago no seguro (JCB) no disponible	MSG0008
<b>SIS0460</b>	Método de pago no seguro (Amex) no disponible	MSG0008
<b>SIS0461</b>	Método de pago no seguro (Amex) no disponible	MSG0008
<b>SIS0463</b>	Método de pago no disponible	MSG0008
<b>SIS0464</b>	Método de pago no seguro no disponible (MasterCard Comercial)	MSG0008
<b>SIS0465</b>	Método de pago no seguro no disponible	MSG0008
<b>SIS0469</b>	Error, no se ha superado el proceso de control de fraude.	MSG0008
<b>SIS0487</b>	Error, el comercio no tiene el método de pago Paygold	MSG0008
<b>SIS0488</b>	Error, el comercio no tiene el método de pago Pago Manual	MSG0008

## 8. ANEXOS

### 8.1 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.	<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.

## Integración utilizando HMAC SHA256

DATO	NOMBRE DEL DATO	Long. / Tipo	COMENTARIOS
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_UrKO</i>	250/A-N	Opcional: si se envía será utilizado como URLKO ignorando el configurado en el módulo de administración en caso de tenerlo
Identificación de comercio:	<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 considerarán 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.
Referencia	<i>Ds_Merchant_Identifier</i>	8/N	Opcional. Su uso es específico del pago por Referencia o Pago1-Clic.
Código de grupo	<i>Ds_Merchant_Group</i>	9/N	Opcional. Su uso es específico del pago por Referencia o Pago1-Clic.
Pago sin autenticación	<i>Ds_Merchant_DirectPayment</i>	4/N	Opcional. Su uso es específico del pago por Referencia o Pago1-Clic.
Tarjeta	<i>Ds_Merchant_Pan</i>	19/N	Opcional. Tarjeta. Su longitud depende del tipo de tarjeta.
Caducidad	<i>Ds_Merchant_ExpiryDate</i>	4/N	Opcional. Caducidad de la tarjeta. Su formato es AAMM, siendo AA los dos últimos dígitos del año y MM los dos dígitos del mes.
CVV2	<i>Ds_Merchant_CVV2</i>	3-4/N	Opcional. Código CVV2 de la tarjeta.

## 8.2 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_UrIOK`) o URL KO (`Ds_Merchant_UrIKO`) 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.

## Integración utilizando HMAC SHA256

DATO	NOMBRE DEL DATO	LONG/TIPO	COMENTARIOS
Pago Seguro	<i>Ds_SecurePayment</i>	1 / Núm.	0 – Si el pago <b>NO</b> es seguro 1 – Si el pago es seguro
Tipo de operación	<i>Ds_TransactionType</i>	1 / A-N	Tipo de operación que se envió en el formulario de pago
País del titular	Ds_Card_Country	3/Núm	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	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
Marca de Tarjeta	Ds_Card_Brand	2 / Núm	Opcional: No se debe hacer una validación cerrada sobre estos valores, pues pueden variar y/o añadirse nuevos. Valores posibles: 1 – VISA 2 –MASTERCARD 8 – AMEX 9 – JCB 22 – UPI 6 – DINERS 22 – CUP 7 - PRIVADA


Estos son los posibles valores del Ds\_Response o "Código de respuesta":

CÓDIGO	SIGNIFICADO
<b>0000 a 0099</b>	<b>Transacción autorizada para pagos y preautorizaciones</b>
<b>900</b>	<b>Transacción autorizada para devoluciones y confirmaciones</b>
<b>400</b>	<b>Transacción autorizada para anulaciones</b>
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 /



### 8.3 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)>
<!ELEMENT Ds_SecurePayment (#PCDATA)>
<!ELEMENT Ds_Amount (#PCDATA)>
<!ELEMENT Ds_Currency (#PCDATA)>
<!ELEMENT Ds_Order (#PCDATA)>
<!ELEMENT Ds_MerchantCode (#PCDATA)>
<!ELEMENT Ds_Terminal (#PCDATA)>
<!ELEMENT Ds_Response (#PCDATA)>
<!ELEMENT Ds_MerchantData (#PCDATA)>
<!ELEMENT Ds_Card_Type (#PCDATA)>
<!ELEMENT Ds_TransactionType (#PCDATA)>
<!ELEMENT Ds_ConsumerLanguage (#PCDATA)>
<!ELEMENT Ds_ErrorCode (#PCDATA)>
<!ELEMENT Ds_CardCountry (#PCDATA)>
<!ELEMENT Ds_AuthorisationCode (#PCDATA)>
<!ELEMENT Signature (#PCDATA)>
<!ELEMENT DS_Card_Type (#PCDATA)>
```

## Integración utilizando HMAC SHA256

Ejemplo:

Sea el siguiente mensaje:

```
<Message>
  <Request Ds_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)>
<!ELEMENT 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>
  <Request Ds_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>I3gacbQMEvUYN59YiHkiml-crEMwFAeogI1jLBDFiw=</Signature>
</Message>
```

Mensaje de respuesta desde el comercio al SIS:

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

### **WSDL para el servicio InotificacionSIS estilo rpc/encode**

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:

## Integración utilizando HMAC SHA256

```
<?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>
```

### **WSDL para el servicio InotificacionSIS estilo document/literal**

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:

```
<wsdl:definitions                                xmlns:ns1="http://schemas.xmlsoap.org/soap/http"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:tns="http://notificador.webservice.sis.redsys.es"
xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"      name="RedClsNotificacionSISService"
targetNamespace="http://notificador.webservice.sis.redsys.es">
  <wsdl:types>
    <xs:schema                                xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://notificador.webservice.sis.redsys.es" version="1.0">
      <xs:element name="notificacion">
        <xs:complexType>
          <xs:sequence>
            <xs:element form="qualified" name="datoEntrada" nillable="true" type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="notificacionResponse">
        <xs:complexType>
          <xs:sequence>
            <xs:element form="qualified" name="notificacionReturn" nillable="true"
type="xs:string"/>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:schema>
  </wsdl:types>
  <wsdl:message name="procesaNotificacionSIS">
    <wsdl:part element="tns:notificacion" name="parameters"/>
  </wsdl:message>
  <wsdl:message name="procesaNotificacionSISResponse">
    <wsdl:part element="tns:notificacionResponse" name="parameters"/>
  </wsdl:message>
  <wsdl:portType name="RedClsNotificacionSIS">
    <wsdl:operation name="procesaNotificacionSIS">
      <wsdl:input message="tns:procesaNotificacionSIS" name="procesaNotificacionSIS"/>
      <wsdl:output message="tns:procesaNotificacionSISResponse"
name="procesaNotificacionSISResponse"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding                                name="RedClsNotificacionSISServiceSoapBinding"
type="tns:RedClsNotificacionSIS">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="procesaNotificacionSIS">
      <soap:operation soapAction="" style="document"/>
      <wsdl:input name="procesaNotificacionSIS">
        <soap:body use="literal"/>
      </wsdl:input>
      <wsdl:output name="procesaNotificacionSISResponse">
        <soap:body use="literal"/>
      </wsdl:output>
    </wsdl:operation>
  </wsdl:binding>
</wsdl:definitions>
```

## Integración utilizando HMAC SHA256

```
</wsdl:binding>
<wsdl:service name="RedClsNotificacionSISService">
  <wsdl:port binding="tns:RedClsNotificacionSISServiceSoapBinding"
name="RedClsNotificacionSISPort">
    <soap:address location="https://host:port/services/RedClsNotificacionSISPort"/>
  </wsdl:port>
</wsdl:service>
</wsdl:definitions>
```