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

SERVICIO TÉCNICO TPV VIRTUAL

Teléfono: 902 365 650 opción 2

TPVVirtual@bancsabadell.com

Atención especial para migración SHA1 a SHA2: lunes a viernes de 10h a 19h

Lista de modificaciones		
Fecha	**Versión**	**Cambios realizados**
05/11/2015	1.0	Creación del documento
11/11/2015	1.1	Modificación en los ejemplos de JAVA Añadida tabla de códigos de transacción

ÍNDICE DE CONTENIDO

1. Introducción

1.1 Objetivo

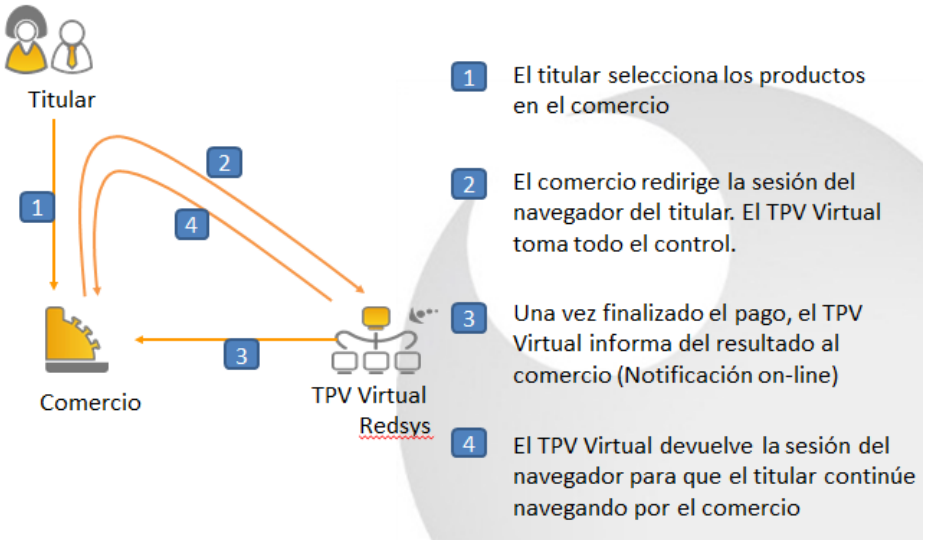
Este documento recoge los aspectos técnicos necesarios para que un comercio realice la integración con el TPV Virtual mediante conexión por Redirección del navegador del cliente comprador.

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

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

2. Descripción general del flujo

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



2.1 Envío de petición al TPV Virtual

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

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

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

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

2.2 Recepción del resultado (Notificación on-line)

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

La notificación on-line consiste en un POST HTTP con la información del resultado codificado 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 64y 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 64enviada en el parámetro anterior. **El comercio es responsable de validar el HMAC enviado por el TPV Virtual para asegurarse de la validez de la respuesta. Esta validación es necesaria para garantizar que los datos no han sido manipulados y que el origen es realmente el TPV Virtual.**

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

2.3 Retorno del control de la navegación del titular

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

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

3. Formulario de envío de petición

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

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="eyJEU19NRVJDSEFOVF9BTU9VTlQiOiIxNDUiLCJEU19NRVJDSEFOVF9PUkRFUiI6Ij
        E0NDYxMTc1NTUiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiIzMjcyMzQ2ODgiLCJE
        U19NRVJDSEFOVF9DVVJSRU5DWSI6Ijk3OCIsIkRTX01FUkNIQU5UX1RSQU5TQUNUSU9
        OVFlQRSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCI6IjEiLCJEU19NRVJDSEFOVF9NRVJ
        DSEFOVFVSTCI6Imh0dHA6XC9cL3d3dy5iYW5jc2FiYWRlbGwuY29tXC91cmxOb3RpZmljY
        WNpb24ucGhwIiwiRFNfTUVSQ0hBTlRfVVJMT0siOiJodHRwOlwvXC93d3cuYmFuY3NhYmF
        kZWxsLmNvbVwvdXJsT0sucGhwIiwiRFNfTUVSQ0hBTlRfVVJMS08iOiJodHRwOlwvXC93d3
        cuYmFuY3NhYmFkZWxsLmNvbVwvdXJsS08ucGhwIn0="/>
        <input type="hidden" name="Ds_Signature"
        value="QfLVUv4nF2Nw7jBAkw0w8H0eRlwh2E1w/ZlKHdA2Sq0="/>
</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="eyJEU19NRVJDSEFOVF9BTU9VTlQiOiIxNDUiLCJEU19NRVJDSEFOVF9PUkRFUiI6I
        jE0NDYwNjg1ODEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiIzMjcyMzQ2ODgiLCJ
        EU19NRVJDSEFOVF9DVVJSRU5DWSI6Ijk3OCIsIkRTX01FUkNIQU5UX1RSQU5TQUNUSU
        9OVFlQRSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCI6IjEiLCJEU19NRVJDSEFOVF9NR
        VJDSEFOVFVSTCI6Imh0dHA6XC9cL3d3dy5iYW5jc2FiYWRlbGwuY29tXC91cmxOb3RpZm
        ljYWNpb24ucGhwIiwiRFNfTUVSQ0hBTlRfVVJMT0siOiJodHRwOlwvXC93d3cuYmFuY3NhY
        mFkZWxsLmNvbVwvdXJsT0sucGhwIiwiRFNfTUVSQ0hBTlRfVVJMS08iOiJodHRwOlwvXC9
        3d3cuYmFuY3NhYmFkZWxsLmNvbVwvdXJsS08ucGhwIiwiRFNfTUVSQ0hBTlRfUEFOIjoiN
        DU0ODgxMjA0OTQwMDAwNCIsIkRTX01FUkNIQU5UX0VYUElSWURBVEUiOiIxNTEyIiwiRF
        NfTUVSQ0hBTlRfQ1ZWMiI6IjEyMyJ9"/>
        <input type="hidden" name="Ds_Signature"
        value="1KyJDBDUoD/62iK45MGFWcaB2WOPXcmqJrYOfzSx8hg="/>
</form>
```

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

3.1 Identificar la versión de algoritmo de firma a utilizar

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

3.2 Montar la cadena de datos de la petición

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

El nombre de cada parámetro debe indicarse en mayúsculas o con estructura "CamelCase" (Por ejemplo: DS_MERCHANT_AMOUNT o Ds_Merchant_Amount). La lista de parámetros que se pueden incluir en la petición se describe en el **Anexo 1** (Punto 8.1. Datos de la solicitud 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"}

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

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

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

eyJEU19NRVJDSEFOVF9BTU9VTlQiOiIxNDUiLCJEU19NRVJDSEFOVF9PUkRFUiI6IjE0NDY xMTc1NTUiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiIzMjcyMzQ2ODgiLCJEU19NR VJDSEFOVF9DVVJSRU5DWSI6Ijk3OCIsIkRTX01FUkNIQU5UX1RSQU5TQUNUSU9OVFlQR SI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCI6IjEiLCJEU19NRVJDSEFOVF9NRVJDSEFO VFVSTCI6Imh0dHA6XC9cL3d3dy5iYW5jc2FiYWRlbGwuY29tXC91cmxOb3RpZmljYWNpb 24ucGhwIiwiRFNfTUVSQ0hBTlRfVVJMT0siOiJodHRwOlwvXC93d3cuYmFuY3NhYmFkZWx sLmNvbVwvdXJsT0sucGhwIiwiRFNfTUVSQ0hBTlRfVVJMS08iOiJodHRwOlwvXC93d3cuYm FuY3NhYmFkZWxsLmNvbVwvdXJsS08ucGhwIn0=

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

eyJEU19NRVJDSEFOVF9BTU9VTlQiOiIxNDUiLCJEU19NRVJDSEFOVF9PUkRFUiI6IjE0NDY wNjg1ODEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiIzMjcyMzQ2ODgiLCJEU19NR VJDSEFOVF9DVVJSRU5DWSI6Ijk3OCIsIkRTX01FUkNIQU5UX1RSQU5TQUNUSU9OVFlQR SI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCI6IjEiLCJEU19NRVJDSEFOVF9NRVJDSEFO VFVSTCI6Imh0dHA6XC9cL3d3dy5iYW5jc2FiYWRlbGwuY29tXC91cmxOb3RpZmljYWNpb 24ucGhwIiwiRFNfTUVSQ0hBTlRfVVJMT0siOiJodHRwOlwvXC93d3cuYmFuY3NhYmFkZWx sLmNvbVwvdXJsT0sucGhwIiwiRFNfTUVSQ0hBTlRfVVJMS08iOiJodHRwOlwvXC93d3cuYm FuY3NhYmFkZWxsLmNvbVwvdXJsS08ucGhwIiwiRFNfTUVSQ0hBTlRfUEFOIjoiNDU0ODgx MjA0OTQwMDAwNCIsIkRTX01FUkNIQU5UX0VYUElSWURBVEUiOiIxNTEyIiwiRFNfTUVSQ 0hBTlRfQ1ZWMiI6IjEyMyJ9

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

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

3.3 Identificar la clave a utilizar para la firma

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

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

3.4 Firmar los datos de la petición

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

1. Se genera una clave específica por operación. Para obtener la clave derivada a utilizar en una operación se debe realizar un cifrado 3DES entre la clave del comercio, 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 Banco Sabadell para la generación de este campo, se expone en el apartado 3.5.

3.5 Utilización de librerías de ayuda

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

3.5.1 Librería PHP

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

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

```
include("./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);
$miObj->setParameter("DS_MERCHANT_TRANSACTIONTYPE",$transactioType);
$miObj->setParameter("DS_MERCHANT_TERMINAL",$terminal);
$miObj->setParameter("DS_MERCHANT_MERCHANTURL",$merchantURL);
$miObj->setParameter("DS_MERCHANT_URLOK",$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);
$miObj->setParameter("DS_MERCHANT_URLOK",$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);
```

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 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>
```

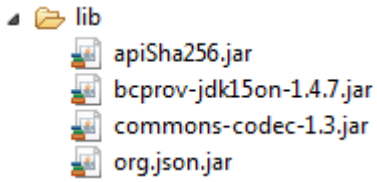
3.5.2 Librería JAVA

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

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

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

   El comercio debe incluir en la vía de construcción del proyectotodas 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:

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();
```

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:

```
String clave = "sq7HjrUOBfKmC576ILgskD5srU870gJ7";
String signature  = ApiMacSha256.createMerchantSignature(clave);
```

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>
```

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. **SíncronaSOAP con WSDL**. Igual a la SíncronaSOAP, pero en este caso el servidor SOAP que desarrolla el cliente se ajusta a las especificaciones de una WSDL que se describe en el Anexo 3(Notificación SOAP) del apartado Anexos del presente documento. Se recomienda este último tipo de notificación, que garantiza un entendimiento perfecto entre servidor y cliente.

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

4.1 Notificación Síncrona y Asíncrona

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

4.1.1 Librería PHP

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

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

   ```
   include("./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);
   ```

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.
}
```

Una vez se ha realizado la llamada a la función "createMerchantSignatureNotif()", 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.

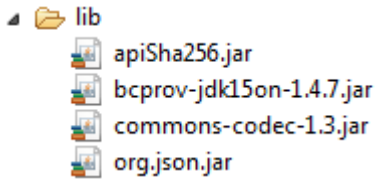
4.1.2 Librería JAVA

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

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

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

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



2. 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);
```

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:

```
String clave = "sq7HjrUOBfKmC576ILgskD5srU870gJ7";
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");
}
```

Una vez se ha realizado la llamada a la función "createMerchantSignatureNotif()", 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.

4.2 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 si, por lo que no tiene sentido rellenar el campo de notificación online, ya que no se tomará en cuenta.

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

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

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

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

4.2.1 Librería PHP

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

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

```
include("./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.

4. Una vez validada la firma, el comercio debe enviar la respuesta de la notificación. Esta respuesta está firmada y para llevar a cabo el cálculo de la firma primero se debe capturar el número de pedido del mensaje recibido en la notificación. Para obtener el número de pedido se debe llamar a la función de la librería "getOrderNotifSOAP()" con el valor del mensaje recibido en la notificación.
   Una vez obtenido el número de pedido, tan sólo falta calcular la firma que se enviará en la respuesta. Para realizar el cálculo de la firma se debe llamar a la función de la librería "createMerchantSignatureNotifSOAPResponse()" con la clave de comercio facilitada, el valor del mensaje de respuesta y el número de pedido capturado, tal y como se muestra a continuación:

```
$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.

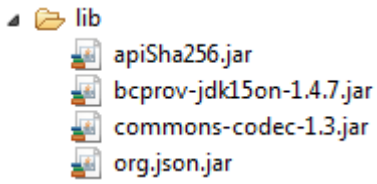
4.2.2 Librería JAVA

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

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

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

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



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

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

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.

```
String clave = "sq7HjrUOBfKmC576ILgskD5srU870gJ7";
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.

4. Una vez validada la firma, el comercio debe enviar la respuesta de la notificación. Esta respuesta está firmada y para llevar a cabo el cálculo de la firma primero se debe capturar el número de pedido del mensaje recibido en la notificación. Para obtener el número de pedido se debe llamar a la función de la librería "getOrderNotifSOAP()" con el valor del mensaje recibido en la notificación.
   Una vez obtenido el número de pedido, tan sólo falta calcular la firma que se enviará en la respuesta. Para realizar el cálculo de la firma se debe llamar a la función de la librería "createMerchantSignatureNotifSOAPResponse()" con la clave de comercio facilitada, el valor del mensaje de respuesta y el número de pedido capturado, tal y como se muestra a continuación:

```
String numPedido  = ApiMacSha256.getOrderNotifSOAP(XML);

String response = "<Response  Ds_Version="0.0">
                    <Ds_Reponse_Merchant>OK</Ds_Reponse_Merchant>
                  </Response";

String clave = "sq7HjrUOBfKmC576ILgskD5srU870gJ7";
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.

5. Retorno del control de la navegación

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

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

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

5.1 Utilización de librerías de ayuda

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

5.1.1 Librería PHP

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

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

```
include("./apiRedsys.php");
```

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

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

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

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

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

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

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

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 = '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.
}
```

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");
```

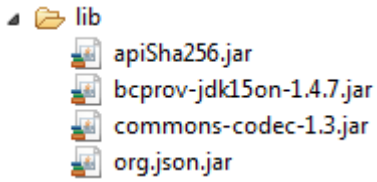
5.1.2 Librería JAVA

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. 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);
```

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:

```
String clave = "sq7HjrUOBfKmC576ILgskD5srU870gJ7";
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");
}
```

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.

6. 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

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: sq7HjrUOBfKmC576ILgskD5srU870gJ7

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

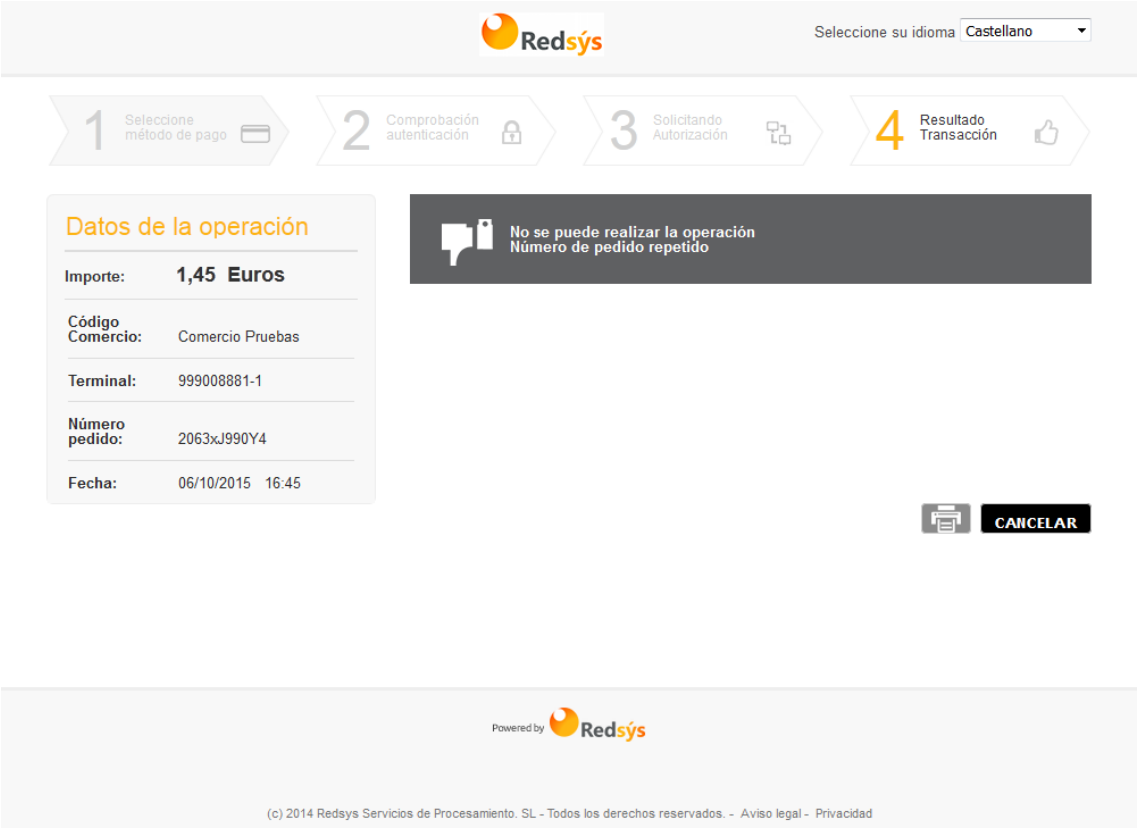
- Para terminal 002 (NO CES):
  - Usuario: 327234688-002
  - Contraseña: 123456a

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



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="tableresults">
202 <tbody></tbody>
```

7.1 Glosario de errores del SIS

ERROR	DESCRIPCIÓN
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
SIS0008	Error falta Ds_Merchant_MerchantCode
SIS0009	Error de formato en Ds_Merchant_MerchantCode
SIS0010	Error falta Ds_Merchant_Terminal
SIS0011	Error de formato en Ds_Merchant_Terminal
SIS0014	Error de formato en Ds_Merchant_Order
SIS0015	Error falta Ds_Merchant_Currency
SIS0016	Error de formato en Ds_Merchant_Currency
SIS0017	Error no se admiten operaciones en pesetas
SIS0018	Error falta Ds_Merchant_Amount
SIS0019	Error de formato en Ds_Merchant_Amount
SIS0020	Error falta Ds_Merchant_MerchantSignature
SIS0021	Error la Ds_Merchant_MerchantSignature viene vacía
SIS0022	Error de formato en Ds_Merchant_TransactionType
SIS0023	Error Ds_Merchant_TransactionType desconocido
SIS0024	Error Ds_Merchant_ConsumerLanguage tiene más de 3 posiciones
SIS0025	Error de formato en Ds_Merchant_ConsumerLanguage
SIS0026	Error No existe el comercio / terminal enviado
SIS0027	Error Moneda enviada por el comercio es diferente a la que tiene
SIS0028	Error Comercio / terminal está dado de baja
SIS0030	Error en un pago con tarjeta ha llegado un tipo de operación que no es ni pago ni preautorización
SIS0031	Método de pago no definido
SIS0033	Error en un pago con móvil ha llegado un tipo de operación que no es ni pago ni preautorización
SIS0034	Error de acceso a la Base de Datos
SIS0037	El número de teléfono no es válido
SIS0038	Error en java
SIS0040	Error el comercio / terminal no tiene ningún método de pago asignado
SIS0041	Error en el cálculo de la HASH de datos del comercio.
SIS0042	La firma enviada no es correcta
SIS0043	Error al realizar la notificación on-line
SIS0046	El bin de la tarjeta no está dado de alta
SIS0051	Error número de pedido repetido
SIS0054	Error no existe operación sobre la que realizar la devolución
SIS0055	Error existe más de un pago con el mismo número de pedido
SIS0056	La operación sobre la que se desea devolver no está autorizada
SIS0057	El importe a devolver supera el permitido
SIS0058	Inconsistencia de datos, en la validación de una confirmación
SIS0059	Error no existe operación sobre la que realizar la confirmación
SIS0060	Ya existe una confirmación asociada a la preautorización
SIS0061	La preautorización sobre la que se desea confirmar no está autorizada
SIS0062	El importe a confirmar supera el permitido
SIS0063	Error. Número de tarjeta no disponible
SIS0064	Error. El número de tarjeta no puede tener más de 19 posiciones
SIS0065	Error. El número de tarjeta no es numérico
SIS0066	Error. Mes de caducidad no disponible
SIS0067	Error. El mes de la caducidad no es numérico
SIS0068	Error. El mes de la caducidad no es válido
SIS0069	Error. Año de caducidad no disponible
SIS0070	Error. El Año de la caducidad no es numérico
SIS0071	Tarjeta caducada

SIS0072	Operación no anulable
SIS0074	Error falta Ds_Merchant_Order
SIS0075	Error el Ds_Merchant_Order tiene menos de 4 posiciones o más de 12
SIS0076	Error el Ds_Merchant_Order no tiene las cuatro primeras posiciones numéricas
SIS0077	Error el Ds_Merchant_Order no tiene las cuatro primeras posiciones numéricas. No se utiliza
SIS0078	Método de pago no disponible
SIS0079	Error al realizar el pago con tarjeta
SIS0081	La sesión es nueva, se han perdido los datos almacenados
SIS0084	El valor de Ds_Merchant_Conciliation es nulo
SIS0085	El valor de Ds_Merchant_Conciliation no es numérico
SIS0086	El valor de Ds_Merchant_Conciliation no ocupa 6 posiciones
SIS0089	El valor de Ds_Merchant_ExpiryDate no ocupa 4 posiciones
SIS0092	El valor de Ds_Merchant_ExpiryDate es nulo
SIS0093	Tarjeta no encontrada en la tabla de rangos
SIS0094	La tarjeta no fue autenticada como 3D Secure
SIS0097	Valor del campo Ds_Merchant_CComercio no válido
SIS0098	Valor del campo Ds_Merchant_CVentana no válido
SIS0112	Error El tipo de transacción especificado en
SIS0113	Excepción producida en el servlet de operaciones
SIS0114	Error, se ha llamado con un GET en lugar de un POST
SIS0115	Error no existe operación sobre la que realizar el pago de la cuota
SIS0116	La operación sobre la que se desea pagar una cuota no es una operación válida
SIS0117	La operación sobre la que se desea pagar una cuota no está autorizada
SIS0118	Se ha excedido el importe total de las cuotas
SIS0119	Valor del campo Ds_Merchant_DateFrecuency no válido
SIS0120	Valor del campo Ds_Merchant_ChargeExpiryDate no válido
SIS0121	Valor del campo Ds_Merchant_SumTotal no válido
SIS0122	Valor del campo Ds_Merchant_DateFrecuency o no Ds_Merchant_SumTotal tiene formato incorrecto
SIS0123	Se ha excedido la fecha tope para realizar transacciones
SIS0124	No ha transcurrido la frecuencia mínima en un pago recurrente sucesivo
SIS0126	Operación denegada para evitar duplicidades.
SIS0132	La fecha de Confirmación de Autorización no puede superar en más de 7 días a la de Preautorización.
SIS0133	La fecha de Confirmación de Autenticación no puede superar en más de 45 días a la de Autenticación Previa.
SIS0139	Error el pago recurrente inicial está duplicado
SIS0142	Tiempo excedido para el pago
SIS0197	Error al obtener los datos de cesta de la compra en operación tipo pasarela
SIS0198	Error el importe supera el límite permitido para el comercio
SIS0199	Error el número de operaciones supera el límite permitido para el comercio
SIS0200	Error el importe acumulado supera el límite permitido para el comercio
SIS0214	El comercio no admite devoluciones
SIS0216	Error Ds_Merchant_CVV2 tiene más de 3 posiciones
SIS0217	Error de formato en Ds_Merchant_CVV2
SIS0218	El comercio no permite operaciones seguras por la entrada /operaciones
SIS0219	Error el número de operaciones de la tarjeta supera el límite permitido para el comercio
SIS0220	Error el importe acumulado de la tarjeta supera el límite permitido para el comercio
SIS0221	Error el CVV2 es obligatorio
SIS0222	Ya existe una anulación asociada a la preautorización
SIS0223	La preautorización que se desea anular no está autorizada
SIS0224	El comercio no permite anulaciones por no tener firma ampliada
SIS0225	Error no existe operación sobre la que realizar la anulación
SIS0226	Inconsistencia de datos, en la validación de una anulación
SIS0227	Valor del campo Ds_Merchant_TransactionDate no válido
SIS0229	No existe el código de pago aplazado solicitado
SIS0252	El comercio no permite el envío de tarjeta

SIS0253	La tarjeta no cumple el check-digit
SIS0254	El número de operaciones de la IP supera el límite permitido por el comercio
SIS0255	El importe acumulado por la IP supera el límite permitido por el comercio
SIS0256	El comercio no puede realizar preautorizaciones
SIS0257	Esta tarjeta no permite operativa de preautorizaciones
SIS0258	Inconsistencia de datos, en la validación de una confirmación
SIS0261	Operación detenida por superar el control de restricciones en la entrada al SIS
SIS0270	El comercio no puede realizar autorizaciones en diferido
SIS0274	Tipo de operación desconocida o no permitida por esta entrada al SIS
SIS0295	Se ha denegado una operación que fue enviada en el mismo minuto para evitar duplic.
SIS0319	El comercio no pertenece al grupo especificado en Ds_Merchant_Group
SIS0321	La referencia indicada en Ds_Merchant_Identifier no está asociada al comercio
SIS0322	Error de formato en Ds_Merchant_Group
SIS0429	Error en la versión enviada por el comercio en el parámetro Ds_SignatureVersion
SIS0430	Error al decodificar el parámetro Ds_MerchantParameters
SIS0431	Error del objeto JSON que se envía codificado en el parámetro Ds_MerchantParameters
SIS0432	Error FUC del comercio erróneo
SIS0433	Error Terminal del comercio erróneo
SIS0434	Error ausencia de número de pedido en la operación enviada por el comercio
SIS0435	Error en el cálculo de la firma

7.2 Códigos de transacción

A) CODIGOS PARA TRANSACCIONES APROBADAS

CÓDIGO	TÍTULO	DESCRIPCIÓN
000	TRANSACCION APROBADA	Transacción autorizada por el banco emisor de la tarjeta
001	TRANSACCION APROBADA PREVIA IDENTIFICACION DE TITULAR	Código exclusivo para transacciones Verified by Visa o MasterCard SecureCode. La transacción ha sido autorizada y, además, el banco emisor nos informa que ha autenticado correctamente la identidad del titular de la tarjeta.
002 - 099	TRANSACCION APROBADA	Transacción autorizada por el banco emisor.

B) CODIGOS PARA TRANSACCIONES DENEGADAS

b.1. Transacciones denegadas por motivos genéricos (SIN RETIRADA DE TARJETA)

CÓDIGO	TÍTULO	DESCRIPCIÓN
101	TARJETA CADUCADA	Transacción denegada porque la fecha de caducidad de la tarjeta que se ha informado en el pago, es anterior a la actualmente vigente.
102	TARJETA BLOQUEDA TRANSITORIAMENTE O BAJO SOSPECHA DE FRAUDE	Tarjeta bloqueada transitoriamente por el banco emisor o bajo sospecha de fraude.
104	OPERACIÓN NO PERMITIDA	Operación no permitida para ese tipo de tarjeta.
106	NUM. INTENTOS EXCEDIDO	Excedido el número de intentos con PIN erróneo.
107	CONTACTAR CON EL EMISOR	El banco emisor no permite una autorización automática. Es necesario contactar telefónicamente con su centro autorizador para obtener una aprobación Manual.
109	IDENTIFICACIÓN INVALIDA DEL COMERCIO O TERMINAL	Denegada porque el comercio no está correctamente dado de alta en los sistemas internacionales de tarjetas.
110	IMPORTE INVALIDO	El importe de la transacción es inusual para el tipo de comercio que solicita la autorización de pago.
114	TARJETA NO SOPORTA EL TIPO DE OPERACIÓN SOLICITADO	Operación no permitida para ese tipo de tarjeta.

116	DISPONIBLE INSUFICIENTE	El titular no dispone de suficiente crédito.
118	TARJETA NO REGISTRADA	Tarjeta inexistente o no dada de alta por banco emisor. (Error SIS0078)
119	DENEGACION SIN ESPECIFICAR EL MOTIVO	Transacción denegada por el banco emisor pero sin que este dé detalles acerca del motivo.
125	TARJETA NO EFECTIVA	Tarjeta inexistente o no dada de alta por banco emisor.
129	ERROR CVV2/CVC2	El código CVV2/CVC2 (los tres dígitos del reverso de la tarjeta) informado por el comprador es erróneo.
167	CONTACTAR CON EL EMISOR: SOSPECHA DE FRAUDE	Debido a una sospecha de que la transacción es fraudulenta el banco emisor no permite una autorización automática. Es necesario contactar telefónicamente con su centro autorizador para obtener una aprobación manual.
180	TARJETA AJENA AL SERVICIO	Operación no permitida para ese tipo de tarjeta.
181-182	TARJETA CON RESTRICCIONES DE DEBITO O CREDITO	Tarjeta bloqueada transitoriamente por el banco emisor.
184	ERROR EN AUTENTICACION	Código exclusivo para transacciones Verified by Visa o MasterCard SecureCode. Transacción denegada por autenticación errónea.
190	DENEGACION SIN ESPECIFICAR EL MOTIVO	Transacción denegada por el banco emisor pero sin que este dé detalles acerca del motivo.
191	FECHA DE CADUCIDAD ERRONEA	Transacción denegada porque la fecha de caducidad de la tarjeta que se ha informado en el pago, no se corresponde con la actualmente vigente.

b.2. Transacciones denegadas por motivos en los que el banco emisor de la tarjeta considera que existen indicios de fraude. (CON RETIRADA DE TARJETA)

CÓDIGO	TÍTULO	DESCRIPCIÓN
201	TARJETA CADUCADA	Transacción denegada porque la fecha de caducidad de la tarjeta que se ha informado en el pago, es anterior a la actualmente vigente. Además, el banco emisor considera que la tarjeta está en una situación de posible fraude.
202	TARJETA BLOQUEDA TRANSITORIAMENTE O BAJO SOSPECHA DE FRAUDE	Tarjeta bloqueada transitoriamente por el banco emisor o bajo sospecha de fraude. Además, el banco emisor considera que la tarjeta está en una situación de posible fraude.
204	OPERACION NO PERMITIDA	Operación no permitida para ese tipo de tarjeta. Además, el banco emisor considera que la tarjeta está en una situación de posible fraude.
207	CONTACTAR CON EL EMISOR	El banco emisor no permite una autorización automática. Es necesario contactar telefónicamente con su centro autorizador para obtener una aprobación manual. Además, el banco emisor considera que la tarjeta está en una situación de posible fraude.
208 – 209	TARJETA PERDIDA O ROBADA	Tarjeta bloqueada por el banco emisor debido a que el titular le ha manifestado que le ha sido robada o perdida. Además, el banco emisor considera que la tarjeta está en una situación de posible fraude.
280	ERROR CVV2/CVC2	Código exclusivo para transacciones en las que se solicita el código de 3 dígitos CVV2 (tarj.Visa) o CVC2 (tarj.MasterCard) del reverso de la tarjeta. El código CVV2/CVC2 informado por el comprador es erróneo. Además, el banco emisor considera que la tarjeta está en una situación de posible fraude.
290	DENEGACION SIN ESPECIFICAR EL MOTIVO	Transacción denegada por el banco emisor pero sin que este dé detalles acerca del motivo. Además, el banco emisor considera que la tarjeta está en una situación de posible fraude.

C) CODIGOS REFERIDOS A ANULACIONES O DEVOLUCIONES (Ds_Merchant_TransactionType = 3) SOLICITADAS POR EL COMERCIO

CÓDIGO	TÍTULO	DESCRIPCIÓN
400	ANULACION ACEPTADA	Transacción de anulación o retrocesión parcial aceptada por el banco emisor.
480	NO SE ENCUENTRA LA OPERACIÓN ORIGINAL O TIME-OUT EXCEDIDO	La anulación o retrocesión parcial no ha sido aceptada porque no se ha localizado la operación original, o bien, porque el banco emisor no ha dado respuesta dentro del time-out predefinido.
481	ANULACION ACEPTADA	Transacción de anulación o retrocesión parcial aceptada por el banco emisor. No obstante, la respuesta del banco emisor se ha recibido con mucha demora, fuera del time-out predefinido.

D) CODIGOS REFERIDOS A CONCILIACIONES DE PRE-AUTORIZACIONES O PRE-AUTENTICACIONES (Ds_Merchant_TransactionType = 2, 8, O o R)

CÓDIGO	TÍTULO	DESCRIPCIÓN
500	CONCILIACION ACEPTADA	La transacción de conciliación ha sido aceptada por el banco emisor.
501 – 503	NO ENCONTRADA LA OPERACION ORIGINAL O TIME-OUT EXCEDIDO	La conciliación no ha sido aceptada porque no se ha localizado la operación original, o bien, porque el banco emisor no ha dado respuesta dentro del time-out predefinido.
9928	ANULACIÓN DE PREAUTORITZACIÓN REALIZADA POR EL SISTEMA	El sistema ha anulado la preautorización diferida al haber pasado más de 72 horas.
9929	ANULACIÓN DE PREAUTORITZACIÓN REALIZADA POR EL COMERCIO	La anulación de la preautorización ha sido aceptada

E) CODIGOS DE ERROR ENVIADOS POR LA PROPIA PLATAFORMA DE PAGOS DE BANCO SABADELL

CÓDIGO	TÍTULO	DESCRIPCIÓN
904	COMERCIO NO REGISTRADO EN EL FUC	Hay un problema en la configuración del código de comercio. Contactar con Banco Sabadell para solucionarlo.
909	ERROR DE SISTEMA	Error en la estabilidad de la plataforma de pagos de Banco Sabadell o en la de los sistemas de intercambio de Visa o MasterCard.
912	EMISOR NO DISPONIBLE	El centro autorizador del banco emisor no está operativo en estos momentos.
913	TRANSMISION DUPLICADA	Se ha procesado recientemente una transacción con el mismo número de pedido (Ds_Merchant_Order).
916	IMPORTE DEMASIADO PEQUEÑO	No es posible operar con este importe.
928	TIME-OUT EXCEDIDO	El banco emisor no da respuesta a la petición de autorización dentro del time-out predefinido.
940	TRANSACCION ANULADA ANTERIORMENTE	Se está solicitando una anulación o retrocesión parcial de una transacción que con anterioridad ya fue anulada.
941	TRANSACCION DE AUTORIZACION YA ANULADA POR UNA ANULACION ANTERIOR	Se está solicitando la confirmación de una transacción con un número de pedido (Ds_Merchant_Order) que se corresponde a una operación anulada anteriormente.
942	TRANSACCION DE AUTORIZACION ORIGINAL DENEGADA	Se está solicitando la confirmación de una transacción con un número de pedido (Ds_Merchant_Order) que se corresponde a una operación denegada.
943	DATOS DE LA TRANSACCION ORIGINAL DISTINTOS	Se está solicitando una confirmación errónea.
944	SESION ERRONEA	Se está solicitando la apertura de una tercera sesión. En el proceso de pago solo está permitido tener abiertas dos sesiones (la actual y la anterior pendiente de cierre).
945	TRANSMISION DUPLICADA	Se ha procesado recientemente una transacción con el mismo número de pedido (Ds_Merchant_Order).
946	OPERACION A ANULAR EN PROCESO	Se ha solicitada la anulación o retrocesión parcial de una transacción original que todavía está en proceso y pendiente de respuesta.

947	TRANSMISION DUPLICADA EN PROCESO	Se está intentando procesar una transacción con el mismo número de pedido (Ds_Merchant_Order) de otra que todavía está pendiente de respuesta.
949	TERMINAL INOPERATIVO	El número de comercio (Ds_Merchant_MerchantCode) o el de terminal (Ds_Merchant_Terminal) no están dados de alta o no son operativos.
950	DEVOLUCION NO PERMITIDA	La devolución no está permitida por regulación
965	VIOLACIÓN NORMATIVA	Violación de la Normativa de Visa o Mastercard
9064	LONGITUD TARJETA INCORRECTA	Nº posiciones de la tarjeta incorrecta
9078	NO EXISTE METODO DE PAGO	Los tipos de pago definidos para el terminal (Ds_Merchant_Terminal) por el que se procesa la transacción, no permiten pagar con el tipo de tarjeta informado.
9093	TARJETA NO EXISTE	Tarjeta inexistente.
9094	DENEGACION DE LOS EMISORES	Operación denegada por parte de los emisoras internacionales
9104	OPER. SEGURA NO ES POSIBLE	Comercio con autenticación obligatoria y titular sin clave de compra segura
9126	OPERACIÓN DENEGADA PARA EVITAR DUPLICIDADES	La operación se ha denegado para evitar duplicidades.
9142	TIEMPO LÍMITE DE PAGO SUPERADO	El titular de la tarjeta no se ha autenticado durante el tiempo máximo permitido.
9218	NO SE PUEDEN HACER OPERACIONES SEGURAS	La entrada Operaciones no permite operaciones Seguras
9253	CHECK-DIGIT ERRONEO	Tarjeta no cumple con el check-digit (posición 16 del número de tarjeta calculada según algoritmo de Luhn).
9256	PREAUTORITZACIONES NO HABILITADAS	La tarjeta no puede hacer Preautorizaciones
9261	LÍMITE OPERATIVO EXCEDIDO	La transacción excede el límite operativo establecido por Banco Sabadell
9912	EMISOR NO DISPONIBLE	El centro autorizador del banco emisor no está operativo en estos momentos.

9913	ERROR EN CONFIRMACION	Error en la confirmación que el comercio envía al TPV Virtual (solo aplicable en la opción de sincronización SOAP)
9914	CONFIRMACION “KO”	Confirmación “KO” del comercio (solo aplicable en la opción de sincronización SOAP)
9915	PAGO CANCELADO	El usuario ha cancelado el pago
9928	AUTORIZACIÓN EN DIFERIDO ANULADA	Anulación de autorización en diferido realizada por el SIS (proceso batch)
9929	AUTORIZACIÓN EN DIFERIDO ANULADA	Anulación de autorización en diferido realizada por el comercio
9997	TRANSACCIÓN SIMULTÁNEA	En el TPV Virtual se está procesando de forma simultánea otra operación con la misma tarjeta.
9994	ESTADO OPERACIÓN: SOLICITADA	La operación está en espera de que el usuario seleccione una tarjeta en el wallet correspondiente (IUPAY!)
9998	ESTADO OPERACIÓN: SOLICITADA	Estado temporal mientras la operación se procesa. Cuando la operación termine este código cambiará.
9999	ESTADO OPERACIÓN: AUTENTICANDO	Estado temporal mientras el TPV realiza la autenticación del titular. Una vez finalizado este proceso el TPV asignará un nuevo código a la operación.

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

DATO	NOMBRE DEL DATO	Long. / Tipo	COMENTARIOS
Identificación de comercio: denominación comercial	*Ds_Merchant_MerchantName*	*25/A-N*	Opcional: será el nombre del comercio que aparecerá en el ticket del cliente (opcional).
Idioma del titular	*Ds_Merchant_ConsumerLanguage*	*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)	Ds_Merchant_SumTotal	12/N.	Obligatorio si se trabaja con pagos recurrentes. Representa la suma total de los importes de las cuotas. Las dos últimas posiciones se consideran decimales.
Datos del comercio	*Ds_Merchant_MerchantData*	*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	*Ds_Merchant_DateFrecuency*	*5/ N*	Frecuencia en días para las transacciones recurrentes y recurrentes diferidas (obligatorio para recurrentes)
Fecha límite	*Ds_Merchant_ChargeExpiryDate*	*10/ A-N*	Formato yyyy-MM-dd fecha límite para las transacciones Recurrentes (Obligatorio para recurrentes y recurrentes diferidas)
Código de Autorización	*Ds_Merchant_AuthorisationCode*	*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	*Ds_Merchant_TransactionDate*	*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.
Identificador	*Ds_Merchant_Identifier*	Max. 40 / A-N	El valor del campo es obligatorio para el primer pago. Para segundo pago y sucesivos, el valor será el identificador que el Banco ha facilitado en el mensaje de respuesta del primer pago.
Grupo de comercios	*Ds_Merchant_Group*	*Max. 9/N*	Opcional. Permite asociar una referencia a un conjunto de comercios.
Pantallas adicionales	*Ds_Merchant_DirectPayment*	-	Opcional. Este parámetro funciona como un flag que indica s hay que mostrar pantallas adicionales (DCC, Fraccionamiento, Autenticación, etc.)
Método de Pago	*Ds_Merchant_PayMethod*	*1 / A-N*	Solo para comercios que permiten el pago con IUPAY y que hayan insertado el botón IUPAY en su página web. C - Pago con Tarjeta O - Pago mediante IUPAY

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 varías posibilidades en función de la necesidad del comercio. Tanto la notificación HTTP como la notificación por mail tienen exactamente el mismo formato.

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

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

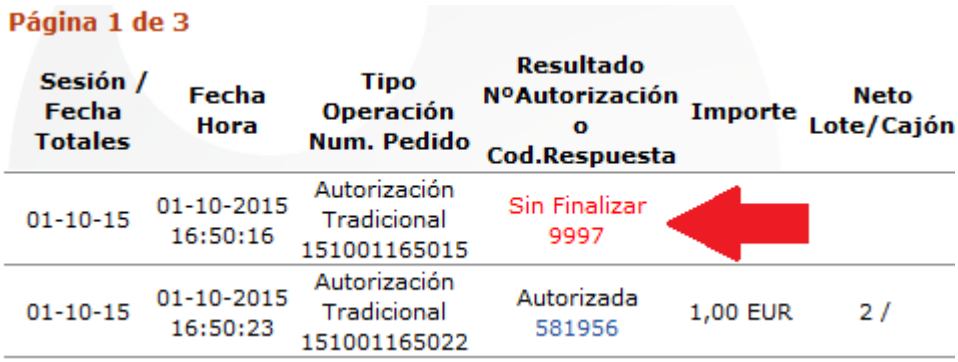
DATO	NOMBRE DEL DATO	LONG/TIPO	COMENTARIOS
Fecha	*Ds_Date*	*dd/mm/yyyy*	Fecha de la transacción
Hora	*Ds_Hour*	*HH:mm*	Hora de la transacción
Importe	*Ds_Amount*	*12 / Núm.*	Mismo valor que en la petición.
Moneda	*Ds_Currency*	*4 / Núm.*	Mismo valor que en la petición. 4 se considera su longitud máxima.
Número de pedido	*Ds_Order*	*12 / A-N.*	Mismo valor que en la petición.
Identificación de comercio: código FUC	*Ds_MerchantCode*	*9 / N.*	Mismo valor que en la petición.
Terminal	*Ds_Terminal*	*3 / Núm.*	Número de terminal que le asignará su banco. 3 se considera su longitud máxima.
Código de respuesta	*Ds_Response*	*4 / Núm.*	Ver tabla siguiente (Posibles valores del Ds_Response).
Datos del comercio	*Ds_MerchantData*	*1024 / A-N*	Información opcional enviada por el comercio en el formulario de pago.
Pago Seguro	*Ds_SecurePayment*	*1 / Núm.*	0 – Si el pago **NO** es seguro 1 – Si el pago es seguro
Tipo de operación	*Ds_TransactionType*	*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: http://unstats.un.org/unsd/methods/

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

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

CÓDIGO	SIGNIFICADO
101	Tarjeta caducada
102	Tarjeta en excepción transitoria o bajo sospecha de fraude
106	Intentos de PIN excedidos
125	Tarjeta no efectiva
129	Código de seguridad (CVV2/CVC2) incorrecto
180	Tarjeta ajena al servicio
184	Error en la autenticación del titular
190	Denegación del emisor sin especificar motivo
191	Fecha de caducidad errónea
202	Tarjeta en excepción transitoria o bajo sospecha de fraude con retirada de tarjeta
904	Comercio no registrado en FUC
909	Error de sistema
913	Pedido repetido
944	Sesión Incorrecta
950	Operación de devolución no permitida
9912/912	Emisor no disponible
9064	Número de posiciones de la tarjeta incorrecto
9078	Tipo de operación no permitida para esa tarjeta
9093	Tarjeta no existente
9094	Rechazo servidores internacionales
9104	Comercio con "titular seguro" y titular sin clave de compra segura
9218	El comercio no permite op. seguras por entrada /operaciones
9253	Tarjeta no cumple el check-digit
9256	El comercio no puede realizar preautorizaciones
9257	Esta tarjeta no permite operativa de preautorizaciones
9261	Operación detenida por superar el control de restricciones en la entrada al SIS
9913	Error en la confirmación que el comercio envía al TPV Virtual (solo aplicable en la opción de sincronización SOAP)
9914	Confirmación "KO" del comercio (solo aplicable en la opción de sincronización SOAP)
9915	A petición del usuario se ha cancelado el pago
9928	Anulación de autorización en diferido realizada por el SIS (proceso batch)
9929	Anulación de autorización en diferido realizada por el comercio
9997	Se está procesando otra transacción en SIS con la misma tarjeta
9998	Operación en proceso de solicitud de datos de tarjeta
9999	Operación que ha sido redirigida al emisor a autenticar

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



Página 1 de 3

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

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)>
```

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>
<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-crEMwFAeogI1jlLBDFiw=</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

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.wsdlxmlns:xs="http://www.w3.org/2
001/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>
```