

TPV-Virtual

Manual de Integración - Redirección

Versión: 2.1

Fecha: 12/04/2019

Referencia: RS.TE.CEL.MAN.0039



Redsys, Servicios de Procesamiento, S.L. – c/ Francisco Sancha, 12 – 28034 Madrid (España)

www.redsys.es

Autorizaciones y control de versión

AUTOR: Redsys	VALIDADO POR: Comercio Electrónico	APROBADO POR: Redsys
Empresa: Redsys	Empresa: Redsys	Empresa: Redsys
Firma:	Firma:	Firma:
Fecha: 27/03/2019	Fecha: 27/03/2019	Fecha: 27/03/2019

Versión	Fecha	Afecta	Breve descripción del cambio
1.0	01/06/2018	TODO	Versión Inicial
1.1	24/09/2018	Punto 8.1	Código ASCII del Ds_Merchant_Order
1.2	07/11/2018	Punto 8.6	Se añade la opción de reintentar el pago
1.3	18/12/2018	Punto 8.6	Se modifica la opción de reintentar el pago
1.4	10/01/2019	Punto 8.1, 8.2 y 8.6	Se añade el campo Ds_Merchant_Paymethods, se añade una corrección sobre el envío del código de respuesta y se matizan los requisitos del comercio en los reintentos de pago
1.5	29/01/2019	Punto 8.1	Se añade en el parámetro Ds_Merchant_TransactionType el tipo de Anulación de Autorización
1.6	14/03/2019	Punto 8	Se rehacen los puntos 8.1, 8.2 y se eliminan los puntos de Monedas e Idiomas
2.0	27/03/2019	Varios puntos	Añadida la información sobre EMV3DS y PSD2 Añadida referencia a los nuevos documentos TPV-Virtual

			GuiaErroresSIS.xlsx y TPV-Virtual Parámetros Entrada-Salida.xlsx
2.1	12/04/2019	Entorno de pruebas Código de Error	Añadidas tarjetas de pruebas para EMV3DS La hoja de cálculo de los errores SIS, se modifica para incluirla en la hoja de cálculo Parámetros de entrada-salida

ÍNDICE

1. INTRODUCCIÓN	6
1.1 OBJETIVO	6
1.2 DEFINICIONES, SIGLAS Y ABREVIATURAS	6
1.3 REFERENCIAS	6
2. DESCRIPCIÓN GENERAL DEL FLUJO	7
2.1 ENVÍO DE PETICIÓN AL TPV VIRTUAL	7
2.2 RECEPCIÓN DEL RESULTADO (NOTIFICACIÓN ON-LINE)	8
2.3 RETORNO DEL CONTROL DE LA NAVEGACIÓN DEL TITULAR	8
3. FORMULARIO DE ENVÍO DE PETICIÓN	10
3.1 IDENTIFICAR LA VERSIÓN DE ALGORITMO DE FIRMA A UTILIZAR	11
3.2 MONTAR LA CADENA DE DATOS DE LA PETICIÓN	11
3.3 IDENTIFICAR LA CLAVE A UTILIZAR PARA LA FIRMA	12
3.4 FIRMAR LOS DATOS DE LA PETICIÓN	13
3.5 UTILIZACIÓN DE LIBRERÍAS DE AYUDA	13
3.5.1 LIBRERÍA PHP	13
3.5.2 LIBRERÍA JAVA	15
3.5.3 LIBRERÍA .NET	16
4. RECEPCIÓN DE LA NOTIFICACIÓN ON-LINE	18
4.1 NOTIFICACIÓN SÍNCRONA Y ASÍNCRONA	18
4.1.1 LIBRERÍA PHP	18
4.1.2 LIBRERÍA JAVA	20
4.1.3 LIBRERÍA .NET	21
5. RETORNO DEL CONTROL DE LA NAVEGACIÓN	24
5.1 UTILIZACIÓN DE LIBRERÍAS DE AYUDA	24
5.1.1 LIBRERÍA PHP	24
5.1.2 LIBRERÍA JAVA	26
5.1.3 LIBRERÍA .NET	27

6.	<u>ADAPTACIONES PSD2 (SEPTIEMBRE 2019)</u>	<u>29</u>
6.1	TRANSACCIONES INICIADAS POR EL COMERCIO (MIT)	29
6.1.1	TRANSACCIONES MIT Y USO DE TOKENIZACIÓN (PAGO POR REFERENCIA)	30
7.	<u>ENTORNO DE PRUEBAS</u>	<u>31</u>
8.	<u>CÓDIGOS DE ERROR</u>	<u>33</u>
9.	<u>ANEXOS</u>	<u>35</u>
9.1	PARÁMETROS DE LA SOLICITUD	35
9.1.1	EJEMPLO DE PAGO/PREAUTORIZACIÓN	35
9.2	PARÁMETROS DE LA NOTIFICACIÓN ON-LINE	35
9.3	REINTENTOS DE PAGO	37
10.	<u>PETICIONES AUTENTICADAS EMV3DS</u>	<u>38</u>
10.1.1	EJEMPLO DE PAGO CON DATOS ADICIONALES EMV3DS	38
11.	<u>ERRORES FRECUENTES</u>	<u>39</u>
12.	<u>PREGUNTAS FRECUENTES</u>	<u>41</u>

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/descargaDeDocumentacionY Ejecutables/>

1.2 Definiciones, siglas y abreviaturas

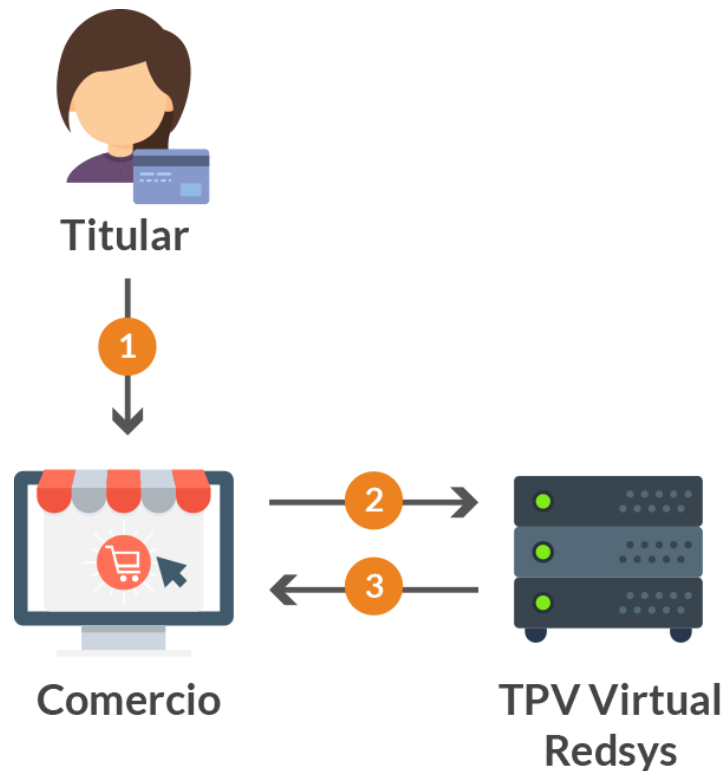
- SIS. Servidor Integrado de Redsys (Servidor del TPV Virtual).
- 3DSecure: Sistema de seguridad para los pagos online. En adelante EMV3DS
- EMV3DS: Siglas para identificar 3DSecure en la nueva versión del TPV-Virtual

1.3 Referencias

- Documentación de Integración con el SIS
- TPV-Virtual Guía SIS.
- TPV-Virtual GuiaErroresSIS.xlsx
- TPV-Virtual Parámetros Entrada-Salida.xlsx

2. Descripción general del flujo

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



1. El titular selecciona los productos que desea comprar en el comercio.
2. El comercio redirige la sesión del navegador del cliente a la URL de Redsys. En esta URL el cliente introduce los datos de tarjeta.
3. El TPV virtual informa al comercio del resultado de la operación y Redsys devuelve la sesión del navegador del cliente al comercio para que continúe navegando en su tienda web.

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 9.1 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 Módulo de Administración.

La notificación on-line puede consistir en:

- Un POST HTTP (Notificación Síncrona y Asíncrona) 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 9.2 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 3 del flujo el TPV Virtual, se mostrará al cliente un recibo con el resultado de la operación, siempre y cuando la configuración del comercio así lo especifique. Además, en caso de que la operación se deniegue, el cliente tendrá la opción de reintentar el pago con otra tarjeta u otro método de pago, siempre y cuando la configuración del comercio lo permita (la opción de reintentar el pago se describe en detalle en el Anexo 9.3).

Por último, en este paso se 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="
eyJEU19NRVJDSEFOVF9BTU9VTIQiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9PukRFUil6IjEyMzQ1Njc4OTAiLCJEU19NRV
JDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9DVVJSRU5DWSi6Ijk3OCIsIkRTX
01FukNIQU5UX1RSQU5TQUNUSU9OVFIQRSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCi6IjEiLCJEU19NRVJD
SEFOVF9NRVJDSEFOVFVSTCI6Imh0dHA6XC9cL3d3dy5wcnVlYmEuY29tXC91cmxOb3RpZmljYWNpb24ucGhwliw
iRfNjTUVSQ0hBTIRfVJMT0SiOiIodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsT0sucGhwliwiRfNjTUVSQ0h
BTIRfVJMS08iOiIodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsT0sucGhwliw0="/>

  <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="
eyJEU19NRVJDSEFOVF9BTU9VTIQiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9PukRFUil6IjE0NDYwNjg1ODEiLCJEU19NR
VJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9DVVJSRU5DWSi6Ijk3OCIsIkRT
X01FukNIQU5UX1RSQU5TQUNUSU9OVFIQRSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCi6IjEiLCJEU19NRVJD
SEFOVF9NRVJDSEFOVFVSTCI6Imh0dHA6XC9cL3d3dy5wcnVlYmEuY29tXC91cmxOb3RpZmljYWNpb24ucGhwli
wiRfNjTUVSQ0hBTIRfVJMT0SiOiIodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsT0sucGhwliwiRfNjTUVSQ0
hBTIRfVJMS08iOiIodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsT0sucGhwliwiRfNjTUVSQ0hBTIRfUEFOljo
iNDUOODgxMjA0OTQwMDAwNCIsIkRTX01FukNIQU5UX0VYUEISWURBEUiOiI0eXNlYmEwY29tXC91cmxOb3RpZmljYWNpb24ucGhwliw0="/>

  <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. 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 9.1.

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/ur/Notificacion.php",
  "DS_MERCHANT_URLOK": "http://www.prueba.com/ur/OK.php",
  "DS_MERCHANT_URLKO": "http://www.bancabadell.com/ur/KO.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/ur/Notificacion.php",
  "DS_MERCHANT_URLOK": "http://www.prueba.com/ur/OK.php",
  "DS_MERCHANT_URLKO": "http://www.prueba.com/ur/KO.php",
  "DS_MERCHANT_PAN": "454881 *****04",
  "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 muestran los objetos JSON que se acaban de mostrar codificados en BASE64:

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

eyJEU19NRVJDSEFOVF9BTU9VTIQioii5OTkiLCJEU19NRVJDSEFOVF9PUkRfUiil6jEyMzQ1Njc4OTAiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUIoii5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9DVVJSRU5DWSI6jk3OCIsIkRTX01FUkNIQU5UX1RSQU5TQUUSU9OVFIQRSi6jiAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCi6jEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVFSTCI6mh0dHA6XC9L3d3dy5wcnVlYmEuYy9tXC91cmxOb3RzZmljYWNPb24ucGhwliwifRNFjTVUSQohBTRIfvVMJTOSoiiodHRwOlwvXC93d3cucHJlZWJhbmNvbVwvdXJsS08ucGhwlnX0S0scGhwliwiRNFjTVUSQohBTRIfvVMJTOSoiiodHRwOlwvXC93d3cucHJlZWJhbmNvbVwvdXJsS08ucGhwlnX0

Ejemplo JSON codificado con envío de datos de tarjeta:

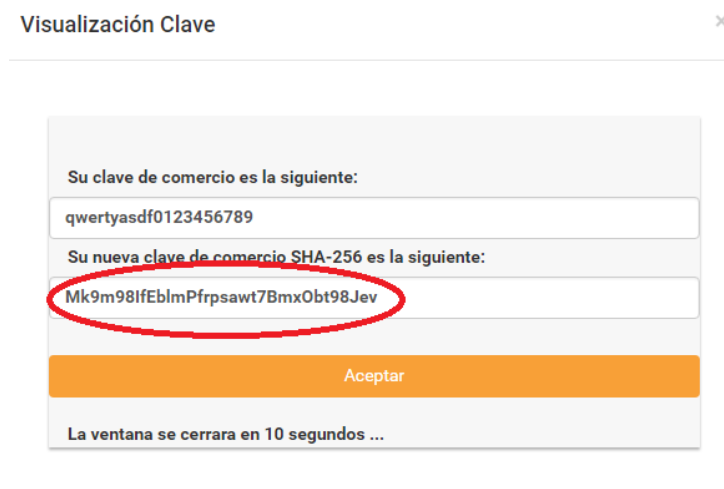
eyIEU19NRVJDSEFOVF9BTU9VTIQiOiIxNDUilCJEU19NRVJDSEFOVF9PUKRFUil6IjEONDYwNjg1ODEilCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEilCJEU19NRVJDSEFOVF9DVVJSRU5DWSI6Ijk3OCIsIkRTX01FukNIQU5UX1RSQU5TQUUNUSU9OVFIQRS6IjJailCJEU19NRVJDSEFOVF9URVJNSU5BTCI6IjEilCJEU19NRVJDSEFOVF9NRVJDSEFOVFVSTCI6ImhodHA6XC9cL3d3dy5wcnVlYmEuY29tXC9c1cmxOb3RpZmIjYWNpbn24ucGhwIiwifRNFtUVSQ0hBTIRfVJVMT0SiOiJodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsT0sucGhwliwiRfNFtUVSQ0hBTIRfVJMS0SiOiJodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsS08ucGhwliwiRfNFtUVSQ0hBTIRfUEFOIjo1NDU0OgVxMjA0OTQwMDAwNClslkRTX01FukNIQU5UX0VYUeISWURBVEUiOiIixNTEYliwiRfNFtUVSQ0hBTIRfQ1ZWMIl6IjEyMyJ9

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:

```
$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>

```

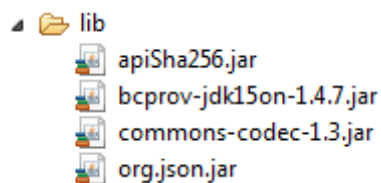
3.5.2 Librería JAVA

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();
```

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:

```
<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.

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;
```

- Capturar los parámetros de la notificación on-line:

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

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

Una vez se ha realizado la llamada a la función "decodeMerchantParameters()", se puede obtener el valor de cualquier parámetro que sea susceptible de incluirse en la notificación on-line (Anexo 9.2). 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.

- 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";
}
```

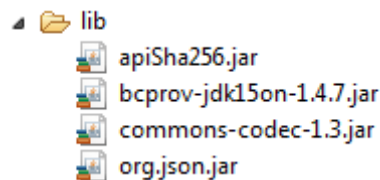
4.1.2 Librería JAVA

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 9.2). 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:

```
// 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"];
}
```

3. Decodificar el parámetro **Ds_MerchantParameters**. Para llevar a cabo la decodificación de este parámetro, se debe llamar a la función de la librería “decodeMerchantParameters()” 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.

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 = "Mk9m98IfEblmPfrpsawt7Bmx0bt98Jev";
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";  
}
```

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.

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

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

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

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

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

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

4. 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 9.2). Para llevar a cabo la obtención del valor de un parámetro se debe llamar a la función “getParameter()” de la librería con el nombre de parámetro, tal y como se muestra a continuación para obtener el código de respuesta:

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

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

5. Validar el parámetro **Ds_Signature**. Para llevar a cabo la validación de este parámetro se debe calcular la firma y compararla con el parámetro **Ds_Signature** capturado. Para ello se debe llamar a la función de la librería “createMerchantSignatureNotif()” con la clave 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);
```

6. 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";
}

```

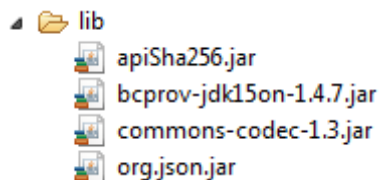
5.1.2 Librería JAVA

A continuación, se presentan los pasos que debe seguir un comercio para la utilización de la librería JAVA proporcionada por 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 del retorno de control de navegación:

```

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

```
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 9.2). Para llevar a cabo la obtención del valor de un parámetro se debe llamar a la función “getParameter()” de la librería con el nombre de parámetro, tal y como se muestra a continuación para obtener el código de respuesta:

```
String codigoRespuesta = apiMacSha256.getParameter("Ds_Response");
```

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

4. Validar el parámetro **Ds_Signature**. Para llevar a cabo la validación de este parámetro se debe calcular la firma y compararla con el parámetro **Ds_Signature** capturado. Para ello se debe llamar a la función de la librería "createMerchantSignatureNotif()" con la clave 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");
}
```

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:

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

6. Adaptaciones PSD2 (Septiembre 2019)

De acuerdo a la norma de PSD2 (entrada en vigor el 14 de septiembre de 2019) para la autenticación de todas las peticiones, se definen una serie de exenciones al uso de SCA que podrán venir marcadas en la petición. Para esta operativa se requiere activación por parte de la entidad (*más información sobre PSD2 en la guía de uso general del servicio "TPV Virtual Guía SIS.pdf"*).

PARÁMETRO	VALORES POSIBLES
DS_MERCHANT_EXCEP_SCA	MIT , LWV, TRA , COR

- LWV : exención por bajo importe (hasta 30 €, con máx. 5 ops. o 100 € acumulado por tarjeta, estos contadores son controlados a nivel de entidad emisora de la tarjeta)
- TRA: exención por utilizarse un sistema de análisis de riesgo (y considerarse bajo riesgo) por parte del adquirente/comercio.
- MIT: operación iniciada por el comercio (sin estar asociada a una acción o evento del cliente) que están fuera del alcance de la PSD2. Este es el caso de las operativas de pagos de suscripciones, recurrentes..etc todas las que requieren el almacenamiento de las credenciales de pago del cliente (COF) o su equivalente mediante operativas de pagos programados tokenizados (uso funcionalidad "pago por referencia" en pagos iniciados por el comercio). Toda operativa de pago iniciada por el comercio (MIT) requiere que inicialmente cuando el cliente concede el permiso al comercio de uso de sus credenciales de pago, dicho "permiso o mandato" se haga mediante operación autenticada con SCA.
- COR: exención restringida a los casos de uso de un protocolo pago corporativo seguro

*Se deberá tener en cuenta que para las exenciones LWV, TRA y COR la primera opción será marcar la exención en el paso de la autenticación, para mejorar la experiencia de usuario. Esto permite que si el emisor no quiere aceptar la propuesta de exención y requiere SCA pueda solicitar la autenticación en el mismo momento sin necesidad de rechazar la operación (challenge required EMV3DS).

6.1 Transacciones iniciadas por el comercio (MIT)

Las transacciones iniciadas por el comercio sin que haya interacción posible con el cliente. Por ejemplo, pago mensual de un recibo o cuota de suscripción. Este tipo de exención requiere el marcaje de la operativa como COF (Credencial on File) de acuerdo al uso concreto que se esté haciendo de las credenciales almacenadas.

Sin embargo no todas las operativas en las que se utilizan datos de tarjeta/credenciales almacenadas (COF) son consideradas MIT. Por ejemplo, la operativa de pago en 1 clic, donde las credenciales del

cliente están almacenadas o tokenizadas (pago por referencia) con el objetivo de facilitar al máximo el momento del pago sin tener que solicitarlas de nuevo al cliente, no se puede considerar una transacción iniciada por el comercio. En tal caso según PSD2, y mientras no se aplique otra exención, se requiere el uso de autenticación reforzada (SCA).

No obstante, con PSD2 estando en vigor, toda operativa de pago iniciada por el comercio (MIT) requiere inicialmente una operación autenticada con SCA que es aquella en la que el cliente concede el permiso y acuerda con el comercio las condiciones para que se usen sus datos de pago para cargos posteriores de acuerdo a un servicio prestado continuado en el tiempo. Esta operativa debe también marcarse debidamente siguiendo la especificación Card-on-file (COF) para indicar que los datos de tarjeta se están almacenando para pagos posteriores.

NOTA: El listado completo de todos los parámetros de entrada del SIS se presenta en la hoja de cálculo "TPV-Virtual Parámetros Entrada-Salida.xlsx".

NOTA2: Acceder a la Guía COF para más información "Especificaciones COF ECom v1.1.pdf".

6.1.1 Transacciones MIT y uso de tokenización (pago por referencia)

En muchos casos se suele utilizar la tokenización de las credenciales de pago del cliente para que Redsys se encargue del almacenamiento seguro de los mismos y asegurar el cumplimiento de los estándares de seguridad de PCI DSS, con el objetivo de generar más tarde pagos iniciados por el comercio sin estar presente el titular de la tarjeta.

En estos casos, en la transacción inicial en la que se solicitar el token o referencia, bajo PSD2 se debe utilizar 3D Secure para aplicar autenticación reforzada y además se debe marcar adecuadamente mediante los parámetros COF el uso que se dará a la misma, de forma que en usos posteriores iniciados por el comercio con el token/referencia, el propio tpv virtual SIS incorpore de forma automática la información de marcaje de uso adecuada e información adicional requerida según la marca de la tarjeta (pej: id transacción original requerido para los pagos COF en Visa "DS_MERCHANT_COF_TXNID")

7. 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>

NOTA: 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): Aquí se deberá poner el número facilitado por su entidad (ejemplo 999008881)
- Terminal (Ds_Merchant_Terminal): Aquí se deberá poner el número facilitado por su entidad (ejemplo 01)
- Clave secreta: sq7HjrUOBfKmC576lLgskD5srU870gJ7

Tarjetas Autorizadas:

- Tarjeta autorizada 1
 - Numeración: **4548812049400004**
 - 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.

- Tarjeta autorizada 2 (Flujo Autenticación EMV3DS Sin solicitud de autenticación del titular)
 - Numeración: 4548814479727229
 - 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.
 - Esta tarjeta irá por el flujo EMV3DS, se podrán mandar datos adicionales en el parámetro DS_MERCHANT_EMV3D
- Tarjeta autorizada 3 (Flujo Autenticación EMV3DS Con solicitud de autenticación del titular)
 - Numeración: 4548817212493017
 - 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.
 - Esta tarjeta irá por el flujo EMV3DS, se podrán mandar datos adicionales en el parámetro DS_MERCHANT_EMV3D
 -

Tarjetas Denegada 190:

- Tarjeta denegada 1
 - Numeración: 5576440022788500
 - 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.
- Tarjeta denegada 2 (Flujo Autenticación EMV3DS Sin solicitud de autenticación del titular)
 - Numeración: 4907277775205123
 - 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.
 - Esta tarjeta irá por el flujo EMV3DS, se podrán mandar datos adicionales en el parámetro DS_MERCHANT_EMV3D

- Tarjeta denegada 3 (Flujo Autenticación EMV3DS Con solicitud de autenticación del titular)
 - Numeración: **4907271141151707**
 - 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.
 - Esta tarjeta irá por el flujo EMV3DS, se podrán mandar datos adicionales en el parámetro DS_MERCHANT_EMV3D

8. Códigos de error

En este apartado se muestra la manera de informar los posibles errores que se pueden producir en el proceso de integración.

NOTA: El listado completo de todos los errores del SIS se presenta en la hoja de cálculo "TPV-Virtual Parámetros Entrada-Salida.xlsx".

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

Página de resultado de la operación


Seleccione su idioma Castellano

1 Seleccione método de pago 

2 Comprobación autenticación 

3 Solicitando Autorización 

4 Resultado Transacción 

Datos de la operación

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



No se puede realizar la operación
Número de pedido repetido


CANCELAR

Powered by  Redsys

(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>

```



9. ANEXOS

9.1 Parámetros de la solicitud

En la petición de pago hacia el TPV Virtual SIS se tendrán que enviar una serie de datos obligatorios y otros opcionales, que irán en función del tipo de operación y operativa que se desee realizar.

NOTA: El listado completo de todos los parámetros de entrada del SIS se presenta en la hoja de cálculo "TPV-Virtual Parámetros Entrada-Salida.xlsx".

9.1.1 Ejemplo de pago/preautorización

A continuación, se muestra un ejemplo del parámetro Ds_Merchant_Parameters, previo a ser codificado en Base 64:

```
{
  "DS_MERCHANT_ORDER": "1552565870",
  "DS_MERCHANT_MERCHANTCODE": "999008881",
  "DS_MERCHANT_TERMINAL": "999",
  "DS_MERCHANT_CURRENCY": "978",
  "DS_MERCHANT_TRANSACTIONTYPE": "0",
  "DS_MERCHANT_AMOUNT": "1000",
  "DS_MERCHANT_MERCHANTURL": "http://www.prueba.com/urNotificacion.php",
  "DS_MERCHANT_URLOK": "http://www.prueba.com/urLOK.php",
  "DS_MERCHANT_URLKO": "http://www.bancabadell.com/urIKO.php"
}
```

* DS_MERCHANT_TRANSACTIONTYPE: "0" para PAGO

* DS_MERCHANT_TRANSACTIONTYPE: "1" para PREAUTORIZACIÓN

9.2 Parámetros de la notificación on-line

La notificación on-line 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.

El resultado de la operación se informará mediante el parámetro Ds_Response o "Código de respuesta". Además, se informará dicho código de respuesta en la consulta de operaciones, siempre y cuando la operación no está autorizada, tal y como se muestra en la siguiente imagen:

Fecha	Tipo operación	Número de pedido	Resultado operación y código	Importe
29/06/2018 10:01:44	Autorización	290618100053	Autorizada 101311 	1,00 EUR
29/06/2018 10:46:55	Autorización	5674	Sin Finalizar 9998	1,45 EUR
29/06/2018 10:54:06	Autorización	7907vPBMh	Sin Finalizar 9998	1,45 EUR

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.

NOTA: Los datos notificados también 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.

NOTA2: El listado completo de todos los parámetros de salida del SIS se presenta en la hoja de cálculo “TPV-Virtual Parámetros Entrada-Salida.xlsx”.

9.3 Reintentos de pago

La opción de reintentar el pago ofrece al titular la posibilidad de intentar realizar el pago, con otra tarjeta o con otro método de pago, cuando la operación ha sido denegada por el emisor o por error en la autenticación del titular (Error = 184). No se ofrecerá al titular esta opción de reintento del pago, en caso de que la operación se deniegue por aplicación de reglas de fraude, por un error técnico en el proceso de autenticación o por cualquier otro tipo de error.

Para que al cliente le aparezca esta opción, el comercio debe tener una configuración específica en el Módulo de Administración del TPV-Virtual (Permitir Repetir Pedido = SI, con reintentos) y cumplir los siguientes requisitos:

- El comercio debe ser capaz de recibir varias notificaciones asociadas al mismo número de pedido y sólo tener en cuenta la primera notificación que identifique que la operación está autorizada, o en caso de no recibir ninguna notificación de operación autorizada, deberá tener en cuenta la última notificación que reciba sobre el número de pedido en cuestión.
- Actualmente la opción de reintentar no es compatible con los módulos de pago facilitados por Redsys para las tiendas virtuales de Prestashop, Magento, etc. En módulos de pago externos a Redsys esta opción podría no funcionar adecuadamente, provocando un error en la actualización de los pedidos en la tienda virtual.

En la siguiente imagen se muestra un ejemplo del recibo que se muestra al cliente, en el que se le ofrece la opción de reintentar:

The screenshot displays a payment interface with the Redsys logo at the top. A progress bar at the top indicates four steps: 1. Seleccione método de pago, 2. Solicitando datos adicionales, 3. Autenticando, and 4. Resultado de la transacción. The main content area is divided into two sections. On the left, under the heading 'Datos de la operación', there is a table with the following details:

Importe	1,45 €
Comercio	Comercio Pruebas (ESPAÑA)
Terminal	999008881-1
Número pedido	1662VT
Fecha	07/11/2018 09:30
Descripción producto	Viajes & Ocio

On the right, a dark grey box contains a speech bubble icon and the text: 'No se puede realizar la operación. Transacción denegada por su entidad'. Below this, it shows 'Número Tarjeta: *****0003' and 'La operación ha sido denegada.' followed by the question '¿Quiere reintentar el pago con otra tarjeta o método de pago?'. At the bottom of this section are two buttons: 'REINTENTAR' and 'CANCELAR'. A printer icon is located at the bottom right of the main content area. The footer of the interface includes the text 'Powered by Redsys' and '(c) 2014 Redsys Servicios de Procesamiento. SL - Todos los derechos reservados.'

10. Peticiones Autenticadas EMV3DS

En esta integración, sin necesidad de desarrollo específico pasarás a autenticar los pagos con la nueva normativa para 3D Secure (EMV3DS).

Adicionalmente se podrán mandar más parámetros para conocimiento de las marcas y los emisores y así mejorar los ratios de autenticaciones sin intervención del titular de la tarjeta. Para este uso se podrá añadir el parámetro de entrada DS_MERCHANT_EMV3DS tipo JSON en las peticiones de pago y preautorización.

Para más información sobre este parámetro de entrada acceder a la hoja de cálculo “TPV-Virtual Parámetros Entrada-Salida.xlsx”.

10.1.1 Ejemplo de pago con datos adicionales EMV3DS

A continuación, se muestra un ejemplo del parámetro Ds_Merchant_Parameters, previo a ser codificado en Base 64:

```
{
  "DS_MERCHANT_ORDER": "1552565870",
  "DS_MERCHANT_MERCHANTCODE": "999008881",
  "DS_MERCHANT_TERMINAL": "999",
  "DS_MERCHANT_CURRENCY": "978",
  "DS_MERCHANT_TRANSACTIONTYPE": "0",
  "DS_MERCHANT_AMOUNT": "1000",
  "DS_MERCHANT_MERCHANTURL": "http://www.prueba.com/urlNotificacion.php",
  "DS_MERCHANT_URLOK": "http://www.prueba.com/urlOK.php",
  "DS_MERCHANT_URLKO": "http://www.bancsabadell.com/urlKO.php",
  "DS_MERCHANT_EMV3DS": {
    "shipAddrCountry": "840",
    "shipAddrCity": "Ship City Name",
    "shipAddrState": "CO",
    "shipAddrLine3": "Ship Address Line 3",
    "shipAddrLine2": "Ship Address Line 2",
    "shipAddrLine1": "Ship Address Line 1",
    "shipAddrPostCode": "Ship Post Code",
    "cardholderName": "Cardholder Name",
    "email": "example@example.com",
    "mobilePhone": {"cc": "123", "subscriber": "123456789"}
  }
}
```

11. Errores frecuentes

No recibo la respuesta “on-line” de las compras con el TPV Virtual

El TPV Virtual enviará la notificación a la dirección que el comercio haya enviado en el campo Ds_Merchant_MerchantURL de la petición de pago.

En la pestaña “Notificaciones” pueden consultarse todas las notificaciones que se han enviado al comercio, tanto las aceptadas como aquellas en las que el servidor del comercio haya devuelto un error.

Puesto que la respuesta de los servidores sigue un protocolo internacional, el significado concreto de los errores genéricos puede consultarse en multitud de páginas de internet. Por ejemplo:

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

Si el comercio no es capaz de instalarse correctamente un servidor que le permita recibir las notificaciones HTTP y desea solicitar consultoría directa y personalizada, debe ponerse en contacto con su entidad.

No recibo el email de confirmación de las compras con el TPV Virtual.

Ocurre con frecuencia que las notificaciones que envía TPV Virtual sí que llegan al comercio, pero entran en la bandeja de Correo No Deseado. Se debe revisar esta bandeja.

No recibo los parámetros con el resultado de las operaciones de la compra en mis URLs de OK y KO.

Las URL OK y KO nunca deben utilizarse para recibir los parámetros con el resultado de las operaciones. Son URL's que deben utilizarse únicamente para redirigir del titular a la web del comercio con fines comerciales o para que el comercio muestre el recibo al titular, si está configurado para ello.

Error de firma

Cuando hay un error de firma el comercio ha de verificar:

- Que los datos que se han utilizado para hacer la firma son iguales a los que se envían en el formulario, teniendo en cuenta, que cualquier modificación del valor o formato de un campo posterior al cálculo de la firma, hace que ésta sea incorrecta.
- Que la clave secreta empleada por el comercio coincide con la clave que tiene cargada el comercio en el módulo de administración (apartado comercios).
- Se debe revisar que los comercios no están enviando espacios en blanco en la firma. Si la petición se hace mediante cURL o mediante el navegador Safari, puede que se conviertan los símbolos “+” en espacio en blanco. Para que esto no ocurra se deben sustituir los símbolos “+” de la firma por “%2B” (Valor URL encoded).
- Si el comercio no consigue localizar qué parámetro es el erróneo, debe contactar con el Centro de Atención al Cliente de Redsys, o con el departamento de Soporte a la Integración de Redsys, si su entidad le ha facilitado el contacto.

Tengo en mi comercio denegaciones por número de repetido, pero no tengo constancia de haberlos repetido.

Esto ocurre habitualmente porque la plataforma del comercio está generando números de pedido repetido únicamente cuando recibe denegaciones o autorizaciones, pero los está repitiendo cuando las transacciones se quedan a medias. Ante esto hay dos opciones:

- Solicitar al servicio de Soporte que el TPV se configure para que pueda repetir números de pedidos. Máximo de una operación autorizada al día y sin límite para las denegadas.
- Generar siempre números de pedido distintos, no solo para las operaciones autorizadas y denegadas, sino para aquellas que no hayan finalizado trascurrido un tiempo.

Necesito hacer una devolución de una operación, pero no me aparece la opción de devolución en el módulo de administración.

Se debe a que el usuario con el que se está accediendo a Canales no tiene permiso para hacer devoluciones. Si necesita este permiso debe ponerse en contacto con su entidad.

12. Preguntas Frecuentes

Soy un comercio y necesito conocer la clave de encriptación de mi TPV Virtual

Para ver la clave del TPV Virtual hay que seguir los siguientes pasos:

Acceda a su módulo de administración de su TPV virtual.

Seleccione la opción "comercio" y pulse "ver clave"

Introduzca su contraseña de su usuario del TPV virtual y pulse aceptar. -Tendrá acceso a ver la clave del comercio durante 10 segundos.

Mi usuario de comercio de acceso al módulo de administración del Canales está bloqueado. ¿Cómo puedo desbloquearlo?

Bajo las casillas de usuario y contraseña existe un link de "He olvidado mi contraseña". Tras pulsarlo deberá escribir su usuario y confirmar la dirección de envío de la nueva contraseña.

Quiero modificar las URLs de OK y KO de mi comercio.

Las URLs de OK y KO son unas direcciones de Internet definidas por el comercio, a las que es posible redirigir al titular una vez aparece la pantalla del recibo de la compra. Existen dos formas de definir estas URLs:

- Si están configuradas en la herramienta de administración.
- Si las envía el comercio en el formulario de pago en los campos Ds_Merchant_UrlOK y Ds_Merchant_UrlKO, será el propio comercio el que deberá modificarlas en el formulario de pago que envía.