TPV-Virtual

Manual de Integración - Redirección

Versión: 3.0
Fecha: 01/10/2020
Referencia: RS.TE.CEL.MAN.0039

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

www.redsys.es

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

Control de versión

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
2.2	02/07/2019	PSD2 y MIT	Operaciones COF y MIT
2.3	04/10/2019	Todo el documento	Se añaden adaptaciones para EVM3DS 2.2

2.4	12/11/2019		Se marcan como avances las características y especificaciones que estarán disponibles a futuro
2.5	16/12/2019	Punto 6,7,8,9 y 10	Añadido conexión PSP Modificación en Exención y tokenización Aclaración y ejemplos para funcionalidades de avances EMV3DS. Modificación tarjetas de pruebas para los avances EMV3DS.
2.6	16/03/2020	Punto 10	Se reestructura punto 10
3.0	01/10/2020	Todo el texto	Se elimina la marca “avance” en las funcionalidades afectadas por PSD2. Las librerías de ayuda a la integración se incluyen en Anexo 2. El punto 9 queda pendiente de especificaciones de las marcas. En el apartado de pruebas, se incluyen tarjetas de las distintas marcas.

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

ÍNDICE

1. INTRODUCCIÓN 6

1.1 OBJETIVO 6
1.2 DEFINICIONES, SIGLAS Y ABREVIATURAS 6
1.3 REFERENCIAS 7

2. DESCRIPCIÓN GENERAL DEL FLUJO 8

3. ENVÍO DE PETICIÓN AL TPV VIRTUAL 9

3.1 IDENTIFICAR LA VERSIÓN DE ALGORITMO DE FIRMA A UTILIZAR 10
3.2 MONTAR LA CADENA DE DATOS DE LA PETICIÓN 10
3.3 IDENTIFICAR LA CLAVE A UTILIZAR PARA LA FIRMA 12
3.4 FIRMAR LOS DATOS DE LA PETICIÓN 12

4. RECEPCIÓN DE LA NOTIFICACIÓN ON-LINE 13

5. RETORNO DEL CONTROL DE LA NAVEGACIÓN AL TITULAR 14

6. INTEGRACIÓN PARA PSPS 15

CONFIGURACIÓN 15
SOLICITUD Y RECEPCIÓN DE CLAVES 15
ENVÍO DE PETICIÓN AL TPV VIRTUAL 15
RECEPCIÓN DEL RESULTADO 15
EJEMPLO DE PETICIONES 16

7. PETICIONES AUTENTICADAS EMV3DS 17

EJEMPLO DE PAGO CON DATOS ADICIONALES EMV3DS 17

8. ADAPTACIONES PSD2 18

8.1 FUNCIONAMIENTO DE LAS SOLICITUDES DE EXENCIÓN A LA SCA (STRONG COSTUMER AUTHENTICATION) 19
PETICIÓN CON ENVIÓ DE EXENCIÓN 19
8.2 TRANSACCIONES INICIADAS POR EL COMERCIO (MIT- MERCHAT INITIATED TRANSACTION) 19

9. FUNCIONALIDADES AVANZADAS EMV3DS (3RI- OTA) 20

10. ENTORNO DE PRUEBAS 21

11. CÓDIGOS DE ERROR 23

12. ERRORES FRECUENTES 25

13. PREGUNTAS FRECUENTES 27

ANEXO 1 28

1.1 PARÁMETROS DE LA SOLICITUD 28
1.1.1 EJEMPLO DE PAGO/PREAUTORIZACIÓN 28
1.2 PARÁMETROS DE LA NOTIFICACIÓN ON-LINE 28
1.3 REINTENTOS DE PAGO 30

ANEXO 2 31

2.1 LIBRERÍAS DE AYUDA. FORMULARIO ENVÍO DE PETICIÓN 31
2.1.1 LIBRERÍA PHP 31
2.1.2 LIBRERÍA JAVA 32
2.1.3 LIBRERÍA .NET 34
2.2 LIBRERÍAS DE AYUDA. RECEPCIÓN ON-LINE 35
2.2.1 LIBRERÍA PHP 35
2.2.2 LIBRERÍA JAVA 36
2.2.3 LIBRERÍA .NET 38
2.3 LIBRERÍAS DE AYUDA. RETORNO DE CONTROL DE NAVEGACIÓN. 39
2.3.1 LIBRERÍA PHP 39
2.3.2 LIBRERÍA JAVA 41
2.3.3 LIBRERÍA .NET 43

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

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 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 ofrece la posibilidad de autenticar al titular mediante el protocolo 3DS, que permite autenticar al titular directamente con el banco emisor de su tarjeta en el momento de realizar la transacción, y que dota de mayor 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:

https://pagosonline.redsys.es/descargas.html

NOTA IMPORTANTE: Con motivo de la entrada en pleno vigor de la directiva de Europea de Pagos PSD2 a lo largo de 2020, se incluyen en esta guía algunas nuevas características y especificaciones técnicas que estarán disponibles a lo largo del 2020, para facilitar la preparación de los trabajos en aquellos casos de comercios que deseen incorporar ciertas posibilidades a su operativa de pago, especialmente en lo referente a la gestión de las autenticaciones y exenciones a la autenticación que la PSD2 contempla.".

1.2 Definiciones, siglas y abreviaturas

- **SIS**. Servidor Integrado de Redsys (Servidor del TPV Virtual).
- **SCA**. Strong Customer Authentication. Autenticación reforzada del titular.
- **Frictionless**. Autenticación sin intervención del titular.
- **Challenge**. Autenticación reforzada del titular (mediante OTP, contraseña estática, biometría, etc).

- **PSD2**. Payment Service Providers. Regulación europea en los servicios de pagos digitales.
- **3DSecure**: Sistema de seguridad para los pagos online. En adelante EMV3DS.
- **EMV3DS**: Siglas para identificar la nueva versión de 3DSecure en el TPV-Virtual.
- **MIT**. Merchant Initiated Transaction. Se refiere a las transacciones iniciadas directamente por el comercio sin que el titular esté presente como por ejemplo en el caso de pagos recurrentes.
- **COF**. Credentials On File. Se refiere a la operativa en la que se almacenan los datos de **tarjeta para futuros usos.**
- **DCC.** Dynamic Currency Conversion. Permite que el titular realice el pago en su propia moneda en lugar de la definida en el terminal.

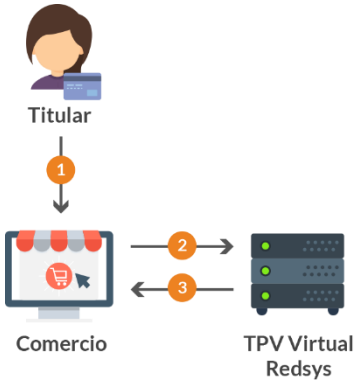
1.3 Referencias

- Documentación de Integración con el SIS
- TPV-Virtual Guía SIS.
- TPV-Virtual Parámetros Entrada-Salida.xlsx
- Especificaciones COF ECOM

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

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.

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

3. Envío de petición al Tpv virtual

El comercio envía 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. Ver *guía Tpv Virtual Parámetros Entrada-Salida. xlsx*, en la que 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

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="
        eyJEU19NRVJDSEFOVF9BTU9VTlQiOiI5OTkiLCJEU19NRVJDSEFOVF9PUkRFUiI6IjEyMzQ1Njc4OTAiLCJEU19NRV
        JDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9DVVJSRU5DWSI6Ijk3OCIsIkRTX
        01FUkNIQU5UX1RSQU5TQUNUSU9OVFlQRSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCI6IjEiLCJEU19NRVJDS
        EFOVF9NRVJDSEFOVFVSTCI6Imh0dHA6XC9cL3d3dy5wcnVlYmEuY29tXC91cmxOb3RpZmljYWNpb24ucGhwIiw
        iRFNfTUVSQ0hBTlRfVVJMT0siOiJodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsT0sucGhwIiwiRFNfTUVSQ0h
        BTlRfVVJMS08iOiJodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsS08ucGhwIn0="/>

        <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"/>
```

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

```
<input type="hidden" name="Ds_MerchantParameters" value="
eyJEU19NRVJDSEFOVF9BTU9VTlQiOiIxNDUiLCJEU19NRVJDSEFOVF9PUkRFUiI6IjE0NDYwNjg1ODEiLCJEU19NR
VJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9DVVJSRU5DWSI6Ijk3OCIsIkRT
X01FUkNIQU5UX1RSQU5TQUNUU9OVFlQRSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCI6IjEiLCJEU19NRVJD
SEFOVF9NRVJDSEFOVFVSTCI6Imh0dHA6XC9cL3d3dy5wcnVlYmEuY29tXC91cmxOb3RpZmljYWNpb24ucGhwIi
wiRFNfTUVSQ0hBTlRfVVJMT0siOiJodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsT0sucGhwIiwiRFNfTUVSQ0
hBTlRfVVJMS08iOiJodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsS08ucGhwIiwiRFNfTUVSQ0hBTlRfUEFOIjoi
NDU0ODgxMjA0OTQwMDAwNCIsIkRTX01FUkNIQU5UX0VYUElSWURBVEUiOiIxNTEyIiwiRFNfTUVSQ0hBTlRfQ
1ZWMiI6IjEyMyJ9"/>

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

</form>
```

Nota: es importante tener en cuenta que siempre que el comercio maneja datos de tarjeta, es necesario el cumplimiento de PCI-DSS

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.

El listado completo de parámetros que se pueden incluir en la petición están disponibles en el documento *"TPV-Virtual Parámetros Entrada-Salida.xlsx".*

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_MERCHANTC
ODE":"999008881","DS_MERCHANT_CURRENCY":"978","DS_MERCHANT_TRANSACTIONTYPE":"0","DS_MER
CHANT_TERMINAL":"1","DS_MERCHANT_MERCHANTURL":"http:\/\/www.prueba.com\/urlNotificacion.php"
```

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

,"DS_MERCHANT_URLOK":"http:\/\/www.prueba.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":"999008881","DS_MERCHANT_CURRENCY":"978","DS_MERCHANT_TRANSACTIONTYPE":"0","DS_MERCHANT_TERMINAL":"1","DS_MERCHANT_MERCHANTURL":"http:\/\/www.prueba.com\/urlNotificacion.php","DS_MERCHANT_URLOK":"http:\/\/www.prueba.com\/urlOK.php","DS_MERCHANT_URLKO":"http:\/\/www.prueba.com\/urlKO.php","DS_MERCHANT_PAN":"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:

eyJEU19NRVJDSEFOVF9BTU9VTlQiOiI5OTkiLCJEU19NRVJDSEFOVF9PUkRFUiI6IjEyMzQ1Njc4OTAiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9DVVJSRU5DWSI6Ijk3OCIsIkRTX01FUkNIQU5UX1RSQU5TQUNUSU9OVFlQRSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCI6IjEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVFVSTCI6Imh0dHA6XC9cL3d3dy5wcnVlYmEuY29tXC91cmxOb3RpZmljYWNpb24ucGhwIiwiRFNfTUVSQ0hBTlRfVVJMT0siOiJodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsT0sucGhwIiwiRFNfTUVSQ0hBTlRfVVJMS08iOiJodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsS08ucGhwIn0

Ejemplo JSON codificado con envío de datos de tarjeta:

eyJEU19NRVJDSEFOVF9BTU9VTlQiOiIxNDUiLCJEU19NRVJDSEFOVF9PUkRFUiI6IjE0NDYwNjg1ODEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9DVVJSRU5DWSI6Ijk3OCIsIkRTX01FUkNIQU5UX1RSQU5TQUNUSU9OVFlQRSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCI6IjEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVFVSTCI6Imh0dHA6XC9cL3d3dy5wcnVlYmEuY29tXC91cmxOb3RpZmljYWNpb24ucGhwIiwiRFNfTUVSQ0hBTlRfVVJMT0siOiJodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsT0sucGhwIiwiRFNfTUVSQ0hBTlRfVVJMS08iOiJodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsS08ucGhwIn0hBTlRfUEFOIjoiNDU0ODgxMjA0OTQwMDAwNCIsIkRTX01FUkNIQU5UX0VYUElSWURBVEUiOiIxNTEyIiwiRFNfTUVSQ0hBTlRfQ1ZWMiI6IjEyMyJ9

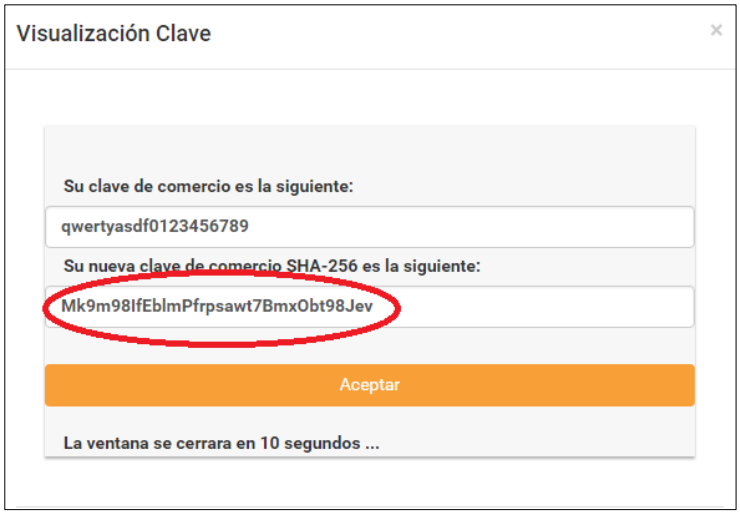
La cadena resultante de la codificación en BASE64 será el valor del parámetro **Ds_MerchantParameters**, tal y como se puede observar en el ejemplo de formulario mostrado al inicio del apartado 3.

NOTA: Puede utilizar librerías de ayuda para la generación de este campo, ver Anexo 2- punto 2.1

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

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 Portal de Administración del Tpv virtual, 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: Puede utilizar librerías de ayuda para la generación de este campo, ver Anexo 2-punto 2.1

4. Recepción de la notificación on-line

Una vez gestionada la transacción, el TPV Virtual puede informar al servidor del comercio del resultado de la misma, mediante una conexión directa al servidor del comercio (paso 3 del flujo descrito en el punto 2). Esta 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 de pago en el TPV Virtual. El Tpv Virtual envía la notificación on-line con el resultado de la transacción a la URL informada por el comercio en la petición de pago, en el parámetro *Ds_Merchant_MerchantURL*

Para que el comercio puede recibir esta notificación on-line tiene que configurarlo en el Portal de Administración del Tpv Virtual.

El comercio que esté configurado para recibir esta notificación, debe capturar **y validar todos los parámetros junto a la firma** de la notificación on-line, previamente a cualquier ejecución en su servidor.

La notificación on-line consiste en un POST HTTP (Notificación Síncrona o Asíncrona) con la información del resultado del pago, 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. Ver *guía Tpv Virtual Parámetros Entrada-Salida. xlsx*, en la que 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.

El TPV Virtual cuenta con diferentes tipos de notificación:

1. **Síncrona.** Implica que el resultado de la compra primero se envía al comercio y a continuación se muestra al cliente. Aunque el comercio no procese correctamente la notificación, el resultado de la operación se mantiene.

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

2. **Asíncrona.** Implica que el resultado de la autorización se comunica a la vez al comercio y al cliente. Aunque el comercio no procese correctamente la notificación, el resultado de la operación se mantiene.

NOTA: Puede utilizar librerías de ayuda para facilitar los desarrollos para la recepción de los parámetros de la notificación on-line y validación de la firma, ver Anexo 2 – punto 2.2

5. Retorno del control de la navegación al titular

Una vez que el cliente ha realizado el proceso de pago en el TPV Virtual, se redirige la navegación hacia a la tienda web (paso 3 del flujo descrito en el punto 2):

- En este momento 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 1- punto 1-3).
- 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.

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 Portal de administración del TPV Virtual. Se pueden disponer de URLs de retorno distintas según el resultado de la transacción (URL OK y URL KO).

Opcionalmente, el TPV Virtual puede incluir los mismos campos de la notificación on-line en las URLs de retorno. En este caso, el comercio debe capturar y validar dichos parámetros previo a cualquier ejecución en su servidor. Para activar esta opción debe configurar en el Portal de administración el campo *Parámetros en las URLs*= SI.

NOTA: Puede utilizar librerías de ayuda para facilitar la captura y validación del retorno de control de la navegación, ver Anexo 2 -punto 2.3.

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

6. Integración para PSPs

Si eres un agregador de comercio o PSP hay una integración específica para que, de esta forma, con clave secreta para PSP puedan operar en nombre de los comercios a nivel de terminal.

Para estas peticiones los parámetros a enviar son los mismos que en la petición habitual de comercios, pero estos tendrán una versión de firma y firma en ANSI X9.19.

No se definen flujos específicos para PSP. Los parámetros de entrada y salida así como los códigos de error se podrán ver en la guía *"TPV-Virtual Parámetros Entrada-Salida.xlsx.*

NOTA: para realizar esta integración se requiere activación por parte de la Entidad.

Configuración

Los comercios-terminales deberán estar asociados y configurados para poder enviar peticiones desde un PSP. Esta configuración debe solicitársela el comercio a su entidad.

Solicitud y recepción de claves

Se podrán recibir dos claves privadas con el protocolo establecido por Redsys.

- Clave para realizar cifrado 3DES del campo DS_MERCHANT_PAN (en caso de ser necesario).
- Clave para firmar la petición (DS_MERCHANTPARAMETERS) según la normal X9.19

Envío de petición al TPV Virtual

Al igual que en la petición de pago enviada por un comercio, el PSP tiene los campos siguientes variando como hemos comentado antes la firma y su versión de firma.

- Ds_SignatureVersion: Constante que indica la versión de firma que se está utilizando. Para esta integración de PSP el valor a utilizar será **T25V1.**
- Ds_MerchantParameters: Cadena en formato JSON con todos los parámetros de la petición codificada en Base 64 y sin retornos de carro (En la sección de anexos 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 Mac X9.19 de la cadena JSON codificada en Base 64 enviada en el parámetro anterior. El valor a enviar en el campo Ds_Signature se obtiene convirtiendo a hexadecimal la Mac obtenida en el paso anterior y extrayendo los 4 primeros bytes (8 caracteres) comenzando por la izquierda del resultado.

Recepción del resultado

La recepción del resultado será firmada de la misma forma que la petición de envío, según la norma ANSI x9.19

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

Ejemplo de peticiones

Cadena en JSON

*{"DS_MERCHANT_AMOUNT":"145","DS_MERCHANT_ORDER":"1446068581","DS_MERCHANT_MERCHANTCODE":"999008881","DS_MERCHANT_CURRENCY":"978","DS_MERCHANT_TRANSACTIONTYPE":"0","DS_MERCHANT_TERMINAL":"1","DS_MERCHANT_MERCHANTURL":"http:\/\/www.prueba.com\/urlNotificacion.php","DS_MERCHANT_PAN":"454881********04","DS_MERCHANT_EXPIRYDATE":"1512","DS_MERCHANT_CVV2":"123"}*

Ciframos en 3DES el parámetro DS_MERCHANT_PAN (si se incluyen datos de tarjeta)

Número de tarjeta	*454881********04*
Número de tarjeta hexadecimal cifrado en 3DES modo CBC sin vector, clave F180E06B7A89A88F4A2A52C8EC1C5D1C	*377f34989cf0ae79857a70aa45f3e4c3*

Cadena en JSON con PAN cifrado:

{"DS_MERCHANT_AMOUNT":"145","DS_MERCHANT_ORDER":"1446068581","DS_MERCHANT_MERCHANTCODE":"999008881","DS_MERCHANT_CURRENCY":"978","DS_MERCHANT_TRANSACTIONTYPE":"0","DS_MERCHANT_TERMINAL":"1","DS_MERCHANT_MERCHANTURL":"http:\/\/www.prueba.com\/urlNotificacion.php","DS_MERCHANT_PAN":"377f34989cf0ae79857a70aa45f3e4c3","DS_MERCHANT_EXPIRYDATE":"1512","DS_MERCHANT_CVV2":"123"}

A continuación, se muestra el objeto JSON codificado en BASE64, campo DS_MERCHANTPARAMETERS:

eyJEU19NRVJDSEFOVF9BTU9VTlQiOiIxNDUiLCJEU19NRVJDSEFOVF9PUkRFUiI6IjE0NDYwNjg1ODEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9DVVJSRU5DWSI6Ijk3OCIsIkRTX01FUkNIQU5UX1RSQU5TQUNUSU9OVFlQRSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCI6IjEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVFVSTCI6Imh0dHA6XC9cL3d3dy5wcnVlYmEuY29tXC91cmxOb3RpZmljYWNpb24ucGhwIiwiRFNfTUVSQ0hBTlRfUEFOIjoiMzc3ZjM0OTg5Y2YwYWU3OTg1N2E3MGFhNDVmM2U0YzMiLCJEU19NRVJDSEFOVF9FWFBJUllEQVRFIjoiMTUxMiIsIkRTX01FUkNIQU5UX0NWVjIiOiIxMjMifQ==

Firmar los datos de la petición

Sobre toda la cadena obtenida en el paso anterior (DS_MERCHANTPARAMETERS) se calcula la firma completa en base a la norma x9.19.

Obtenemos la MAC 5D823A402DE70705 con la clave de cifrado 269289DA6EAD0B20928C8F2F2F6BC752 y el DS_MERCHANTPARAMETERS.

El campo DS_SIGNATURE será cogiendo los 4 primeros bytes (8 caracteres) iniciando por la izquierda del resultado del MAC del paso anterior ***5D823A40***

Formar el mensaje de la petición

- *Ds_SignatureVersion: T25V1*
- *Ds_MerchantParameters: eyJEU19NRVJDSEFOVF9BTU9VTlQiOiIxNDUiLCJEU19NRVJDSEFOVF9PUkRFUiI6IjE0NDYwNjg1ODEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9DVVJSRU5DWSI6Ijk3OCIsIkRTX01FUkNIQU5UX1RSQU5TQUNUSU9OVFlQRSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCI6IjEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVFVSTCI6Imh0dHA6XC9cL3d3dy5wcnVlYmEuY29tXC91cmxOb3RpZmljYWNpb24ucGhwIiwiRFNfTUVSQ0hBTlRfUEFOIjoiMzc3ZjM0OTg5Y2YwYWU3OTg1N2E3MGFhNDVmM2U0YzMiLCJEU19NRVJDSEFOVF9FWFBJUllEQVRFIjoiMTUxMiIsIkRTX01FUkNIQU5UX0NWVjIiOiIxMjMifQ==*
- *Ds_Signature: 5D823A40*

7. Peticiones Autenticadas EMV3DS

Este tipo de integración permite realizar pagos con autenticación del titular mediante el protocolo 3DSecure definido por las marcas. Este tipo de integración está preparada para utilizar las diferentes versiones de dicho protocolo, tanto la versión 3DS v1 como la EMV 3DS 2.

Con el fin de mejorar la experiencia del usuario en el proceso de autenticación, se pueden añadir en la petición de pago los parámetros que han definido las marcas para la versión EMV3DS v2, que aportan datos adicionales del titular de la tarjeta y del dispositivo que está utilizando. En este caso el comercio tiene que añadir en la petición de pago o preatorización el parámetro de entrada DS_MERCHANT_EMV3DS tipo JSON. Para más información sobre este parámetro de entrada acceder al documento "*TPV-Virtual Parámetros Entrada-Salida.xlsx*".

Ejemplo de pago con datos adicionales EMV3DS

A continuación, se muestra un ejemplo del parámetro Ds_MerchantParameters, 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"}
 }
}
```


La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

8. Adaptaciones PSD2

De acuerdo a la normativa PSD2 (entrada en vigor el 14 de septiembre de 2019), directiva europea que tiene como objetivo mejorar la seguridad y reforzar la autenticación del cliente en las operaciones de comercio electrónico. Como norma básica se exige la autenticación del titular en todas las operaciones (SCA), sin embargo, también se define la posibilidad de que el comercio, en la petición de pago solicite una exención para evitar dicha autenticación. Para solicitar una exención el comercio deberá incluir el siguiente parámetro en sus peticiones:

PARÁMETRO	VALORES POSIBLES
DS_MERCHANT_EXCEP_SCA	LWV, TRA, MIT, COR, ATD

- LWV (Low value transaction): 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 (Análisis de riesgo de la operación): esta exención se basa en un análisis de riesgo de la operación (y considerarse bajo riesgo) por parte del adquirente/comercio.
- MIT (Merchant Initiated Transaction): operación iniciada por el comercio (sin estar asociada a una acción o evento del cliente, es decir, sin que haya interacción posible con el cliente), están fuera del alcance de la PSD2. Este es el caso de las operativas de pagos de subscripciones, 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 la operación inicial, cuando el cliente concede el permiso al comercio de uso de sus credenciales de pago, se haga mediante operación autenticada con SCA.

  NOTA: Para enviar operaciones de tipo MIT se recomienda utilizar la integración por REST.
- COR (Secure Corporate Payment): exención restringida a los casos de uso de un protocolo pago corporativo seguro.
- ATD : exención de autenticación delegada. Autenticación Delegada es un programa específico de las marcas. (para más información, consultar con la documentación de las marcas sobre este tema)

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

NOTA: 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).

8.1 Funcionamiento de las solicitudes de exención a la SCA (Strong Costumer Authentication)

Si bien un comercio, a través del servicio de pago que le ofrece su entidad bancaria, puede solicitar que un pago se acoja a alguna de las exenciones a SCA previstas, la aceptación final de dicha solicitud de "no autenticar" al titular de una tarjeta depende, en última instancia, de la entidad emisora de la tarjeta. Esta solicitud de exención en una transacción, no garantiza la aceptación de la misma por parte del banco emisor de la tarjeta. En caso de no aceptación, se va a requerir que se proceda a autenticar la transacción.

Por esta razón, el tpv virtual, acogiéndose a las *best practices* del sector, y para asegurar la mejor experiencia de pago al usuario, priorizará siempre iniciar el flujo de autenticación con el emisor (vía EMV3DS, si el emisor lo soporta) indicando al emisor en dicha solicitud la preferencia del comercio de no autenticar en base a la exención solicitada. De esta forma se mejora la usabilidad y, en caso de que el emisor no acepte la exención, se autentique en el mismo momento al usuario. Si el emisor accede a aplicar la exención solicitada, se cierra el flujo de autenticación sin requerir ni mostrar ninguna pantalla o acción al titular, siendo un proceso transparente para el usuario.

Petición con envió de exención

Una vez conocidas las exenciones que puede utilizar, el comercio deberá añadir en la transacción el parámetro **DS_MERCHANT_EXCEP_SCA** con una de las exenciones permitidas, tal y como se indica a continuación:

Ejemplo:

*{"DS_MERCHANT_AMOUNT":"145","DS_MERCHANT_ORDER":"1446117555","DS_MERCHANT_MERCHANTCODE":"999008881","DS_MERCHANT_CURRENCY":"978","DS_MERCHANT_TRANSACTIONTYPE":"0","DS_MERCHANT_TERMINAL":"1","DS_MERCHANT_MERCHANTURL":"http:\/\/www.prueba.com\/urlNotificacion.php","DS_MERCHANT_URLOK":"http:\/\/www.prueba.com\/urlOK.php","DS_MERCHANT_URLKO":"http:\/\/www.bancsabadell.com\/urlKO.php",**"DS_MERCHANT_EXCEP_SCA":"TRA"}*

8.2 Transacciones iniciadas por el comercio (MIT- Merchat Initiated Transaction)

Una transacción MIT es aquella que es iniciada por el propio comercio sin que haya interacción posible con el cliente. Por ejemplo, pago mensual de un recibo o cuota de subscripción. Este tipo de

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

operaciones, al no estar el cliente presente y no ser posible su autenticación, no requerirán de autenticación del titular (SCA).

Para identificar correctamente este tipo de transacción, el comercio debe incluir en la petición de pago, el parámetro **DS_MERCHANT_EXCEP_SCA** con el valor **MIT** y, además, enviar el parámetro **DS_MERCHANT_DIRECTPAYMENT** con el valor **true**.

Estas transacciones MIT, pueden estar asociadas a una petición inicial de pago (**Operación COF inicia**l) en la que el titular está presente y concede el permiso al comercio para que use sus datos de pago en cargos posteriores, de acuerdo a un servicio prestado de forma continuada en el tiempo. Esta operación inicial deberá ser autenticada con SCA y debe marcarse siguiendo las especificaciones COF. La operación MIT también requiere que se marque correctamente el indicador de COF, de acuerdo al uso concreto que se esté haciendo de las credenciales almacenadas.

Hay que tener en cuenta que no todas las operativas en las que se utilizan datos de tarjeta/credenciales almacenadas (COF) pueden ser consideradas MIT. Por ejemplo, la operativa de **pago en 1 clic**, donde las credenciales del cliente están almacenadas o tokenizadas (pago por referencia), **NO** se pueden considerar transacciones iniciadas por el comercio las peticiones de pago que se realizan utilizando las credenciales almacenadas ya que el titular está presente y por lo tanto puede autenticarse. En este caso, según la normativa PSD2, y mientras no se aplique otra exención, se requiere el uso de autenticación reforzada (SCA).

NOTA 1: Se recomienda que el comercio utilice la integración REST para este tipo de transacciones puesto que en las transacciones MIT no hay interacción directa con el titular.

NOTA 2: Para más información sobre especificaciones de Credentials on File (COF) ver Guía Especificaciones COF Ecom.

NOTA 3: El listado completo de todos los parámetros de entrada del SIS está disponible en el doc. "TPV-Virtual Parámetros Entrada-Salida.xlsx".

9. Funcionalidades Avanzadas EMV3DS (3RI- OTA)

En cuando a este punto de operaciones R3I, las especificaciones todavía no están cerradas por parte de las marcas. Este punto se suprime, temporalmente, hasta que dispongamos de los requisitos de las marcas que nos permitan implementar la solución definitiva para este tipo de transacciones.

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

10. Entorno de pruebas

El comercio puede utilizar el entorno de test para realizar las pruebas que necesite para verificar el correcto funcionamiento de su integración antes de hacer la implantación en el entorno real.

En esta guía se facilitan datos genéricos de prueba que pueden ser utilizados por cualquier cliente, si el comercio está interesado en realizar estas pruebas con los datos de su comercio, deberá dirigirse a su Entidad bancaría para que le facilite los datos de acceso.

Las URLs de acceso al entorno de pruebas son:

URL petición de pago
https://sis-t.redsys.es:25443/sis/realizarPago
URL acceso al Portal de Administración
https://sis-t.redsys.es:25443/canales/portal

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.

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 de firma: sq7HjrUOBfKmC576ILgskD5srU870gJ7

Tarjeta de pruebas	Descripción
4548812049400004 Caducidad: no se valida CVV2: distinto de 999	Tarjeta que permite realizar operaciones con autenticación en versión 1.0.2
4548814479727229 Caducidad: no se valida CVV2: distinto de 999	EMV3DS 2.1 autenticación Frictionless
4548817212493017 Caducidad: no se valida CVV2: distinto de 999	EMV3DS 2.1 autenticación Challenge

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

Operaciones denegadas	
Cualquier operación con valor de CVV2 = 999 o importe terminado en 96	
Pruebas PSD2	
4548815324058868 Caducidad: no se valida CVV2: distinto de 999	EMV3DS 2.2 *Si se envía una exención se realizará autenticación Frictionless. Si no se envía exenciones pedirá Challenge*
4548815374025114 Caducidad: no se valida CVV2: distinto de 999	EMV3DS 2.2 *Si se envía MIT devolverá Frictionless, en cualquier otro caso pedirá Challenge*
4548817212493017 Caducidad: no se valida CVV2: distinto de 999	Tarjeta con soft decline. Denegación 195 si la autorización no va como autenticada.
Pruebas DCC	
4137360000000006 Caducidad: no se valida CVV2: distinto de 999	Tarjeta Visa con DCC y 3DS V1
5424180805648190 Caducidad: no se valida CVV2: distinto de 999	Tarjeta Master con DCC y EMV3DS 2 Frictionless
4117731234567891 Caducidad: no se valida CVV2: distinto de 999	Tarjeta Visa con DCC y EMV3DS 2 Challenge
5409960031405146 Caducidad: no se valida CVV2: distinto de 999	Tarjeta Master con moneda Korona Noruega y EMV3DS 2 Challenge
Otras marcas	
36849800000018 Caducidad: no se valida CVV2: distinto de 999	Tarjeta Diners
36849800000000 Caducidad: no se valida CVV2: distinto de 999	Tarjeta Diners
376674000000008 Caducidad: no se valida CVV2: distinto de 999	Tarjeta Amex
376674000000016 Caducidad: no se valida CVV2: distinto de 999	Tarjeta Amex
3587870000000001	Tarjeta JCB

Caducidad: no se valida CVV2: distinto de 999	
3587870000000019 Caducidad: no se valida CVV2: distinto de 999	Tarjeta JCB

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

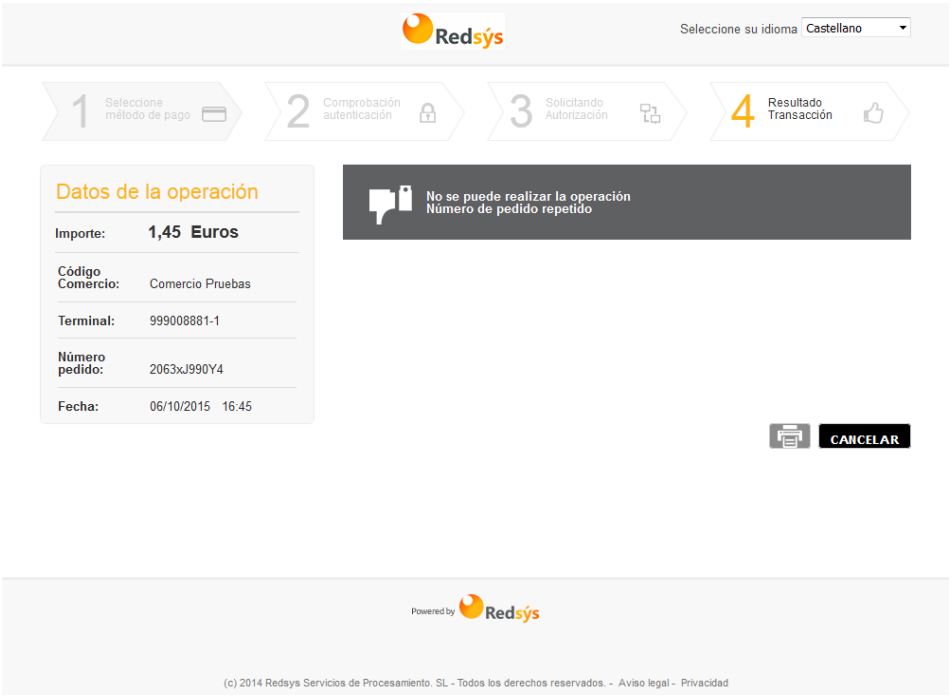
11. Códigos de error

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 códigos de error del SIS está disponible en el documento "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:



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>
```

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

12. 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.

El comercio también puede consultar el resultado de la notificación en el Portal de administración del Tpv virtual, en la opción del menú "Notificaciones", o bien en el detalle de la operación en la pantalla de consulta de operaciones de este Portal de administración

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 la operación. 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 (error SIS0042)

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 Portal de administración (apartado comercios).
- Se debe revisar que 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).

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

- 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 o dirigirse al servicio de soporte de su Entidad.

Tengo en mi comercio denegaciones por número de repetido (SIS0051), 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 al Portal de administración no tiene permiso para hacer devoluciones. Para activar este permiso al usuario tiene que solicitarlo a su Entidad bancaria.

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

13. 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 Portal de administración 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 el Portal 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.

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

ANEXO 1

1.1 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 del SIS está disponible en el documento "TPV-Virtual Parámetros Entrada-Salida.xlsx".

1.1.1 Ejemplo de pago/preautorización

A continuación, se muestra un ejemplo del parámetro Ds_MerchantParameters, 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_TRANSACTIONTYPE":"0" para PAGO
* DS_MERCHANT_TRANSACTIONTYPE":"1" para PREAUTORIZACIÓN

1.2 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

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

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 del SIS está disponible en el documento "TPV-Virtual Parámetros Entrada-Salida.xlsx".

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

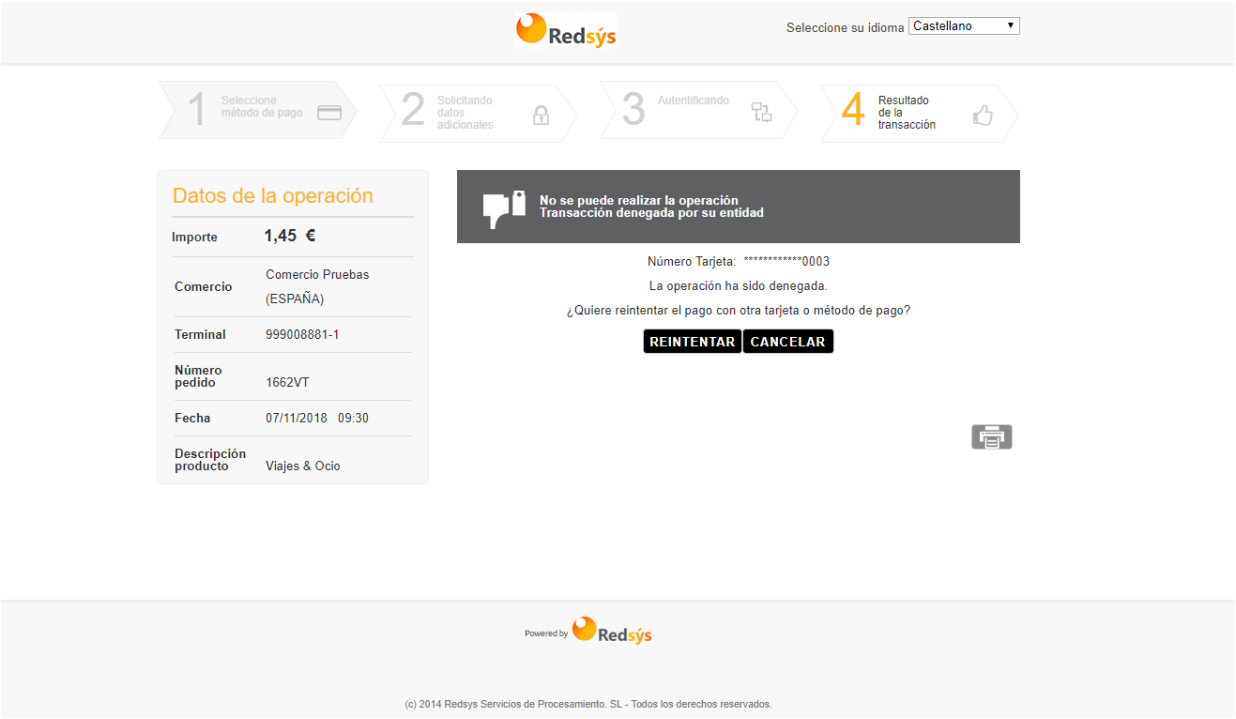
1.3 Reintentos de pago

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:



La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

ANEXO 2

2.1 Librerías de ayuda. Formulario envío de petición

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.

2.1.1 Librería PHP

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

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

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

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

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

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

3. 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();
```

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

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>
```

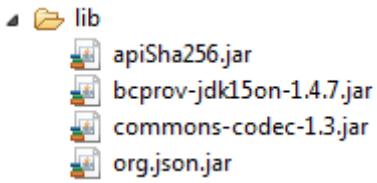
2.1.2 Librería JAVA

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:



La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

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 = "Mk9m98IfEblmPfrpsawt7BmxObt98Jev";
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>
```

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

2.1.3 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. 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 RedysAPI
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>
```

2.2 Librerías de ayuda. Recepción on-line

2.2.1 Librería PHP

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

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

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

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

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

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

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

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

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

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

Una vez se ha realizado la llamada a la función “decodeMerchantParameters()”, se puede obtener el valor de cualquier parámetro que sea susceptible de incluirse en la notificación

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

on-line (Anexo 1.2). 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.

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

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

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

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

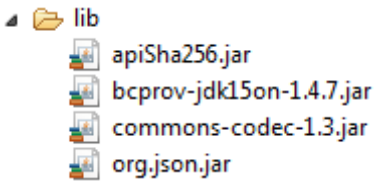
2.2.2 Librería JAVA

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 1.2). 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:

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

```
String claveModuloAdmin = "Mk9m98IfEblmPfrpsawt7BmxObt98Jev";
String signatureCalculada = apiMacSha256.createMerchantSignatureNotif(claveModuloAdmin,
                                                                       params);
```

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

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

2.2.3 Librería .NET

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

1. Importar la librería 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"];
}

// Obtan 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()”

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

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

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

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

2.3 Librerías de ayuda. Retorno de control de navegación.

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.

2.3.1 Librería PHP

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

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

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:

```
$decodec = $miObj->decodeMerchantParameters($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 1.2). 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";
}
```

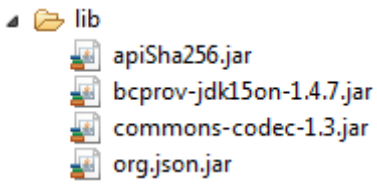
2.3.2 Librería JAVA

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");
```

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

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 1.2). 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 = "Mk9m98IfEblmPfrpsawt7BmxObt98Jev";
String signatureCalculada = apiMacSha256.createMerchantSignatureNotif(claveModuloAdmin,
                                                                      params);
```

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

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

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.

2.3.3 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 = "sq7HjrUOBfKmC576ILgskD5srU870gJ7";
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";
}
```

La propiedad intelectual de este documento pertenece a Redsys. Queda prohibida su reproducción, venta o cesión a terceros.