TPV-Virtual

Manual de Integración - Redirección

Versión: 4.1
Fecha: 24/09/2025
Referencia: RS.TE.CEL.MAN.0039

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

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.
3.1	16/01/2024	Punto 7 Punto 10	Eliminación de 3DSecure versión 1 Eliminación de tarjeta de prueba de 3DSecure versión 1
3.2	31/01/2024	Punto 10	Añadida tarjeta de pruebas DCC en libras
3.3	20/03/2025	Punto 6	Se sustituye la versión V1 de cifrado y firma de PSP por la nueva versión V2.
4.0	20/03/2025	Punto 3, 4 y Anexos	Nuevo Tipo de firma HMAC SHA512
4.1	24/09/2025	Todo el documento	Tipo de firma HMAC SHA512 V2

ÍNDICE

1. Introducción

1. Objetivo

Este documento tiene como propósito detallar los aspectos técnicos necesarios para que un comercio integre correctamente el TPV Virtual de Redsys mediante el método de conexión por redirección del navegador del cliente comprador.

Este tipo de integración permite trasladar la sesión del cliente al entorno seguro del TPV Virtual, donde se realiza la selección del medio de pago y la introducción de los datos de forma protegida, fuera del ámbito de responsabilidad del comercio. Además de ofrecer una implementación sencilla, este método proporciona mayor seguridad al permitir la autenticación del titular de la tarjeta mediante el protocolo EMV 3DS. Este protocolo autentica al comprador directamente con el banco emisor en el momento de la transacción, reforzando la protección de las compras.

Nota: La conexión requiere el uso de un sistema de firma basado en HMAC SHA-512, que garantiza la autenticación mutua entre el servidor del comercio y el TPV Virtual. El comercio puede desarrollar el cálculo de esta firma utilizando funciones estándar disponibles en los distintos entornos de desarrollo. No obstante, para facilitar el proceso, Redsys pone a disposición librerías específicas en PHP, Java y .NET, cuya implementación se describe detalladamente en esta guía. Estas librerías están disponibles en la siguiente dirección:

https://pagosonline.redsys.es/desarrolladores-inicio/integrate-con-nosotros/area-de-descargas-y-documentacion/#

2. Definiciones, siglas y abreviaturas

Término	Definición
SIS	Servidor Integrado de Redsys. Servidor del TPV Virtual encargado de gestionar las transacciones de pago.
SCA	Strong Customer Authentication. Autenticación reforzada del titular, exigida por la normativa PSD2 para aumentar la seguridad en los pagos electrónicos.

Término	Definición
Frictionless	Proceso de autenticación en el que no se requiere intervención del titular, permitiendo una experiencia de compra más fluida.
Challenge	Autenticación reforzada del titular mediante mecanismos como OTP (clave de un solo uso), contraseña estática, biometría, entre otros.
PSD2	Payment Services Directive 2. Regulación europea que establece requisitos para los servicios de pago digital, incluyendo la autenticación reforzada.
EMV 3DS	Nueva versión del protocolo 3DSecure, adoptado por el TPV Virtual para mejorar la seguridad y compatibilidad con la normativa PSD2.
MIT	Merchant Initiated Transaction. Transacciones iniciadas directamente por el comercio sin intervención del titular, como en el caso de pagos recurrentes.
COF	Credentials On File. Operativa que permite almacenar los datos de tarjeta del titular para su uso en futuras transacciones.
DCC	Dynamic Currency Conversion. Funcionalidad que permite al titular realizar el pago en su moneda local, en lugar de la moneda definida en el terminal.
PCI-DSS	Payment Card Industry Data Security Standard. Conjunto de estándares de seguridad desarrollado por las principales marcas de tarjetas para proteger los datos de los titulares durante el procesamiento, almacenamiento y transmisión de información sensible. El cumplimiento de PCI-DSS es obligatorio para todos los comercios que manejan datos de

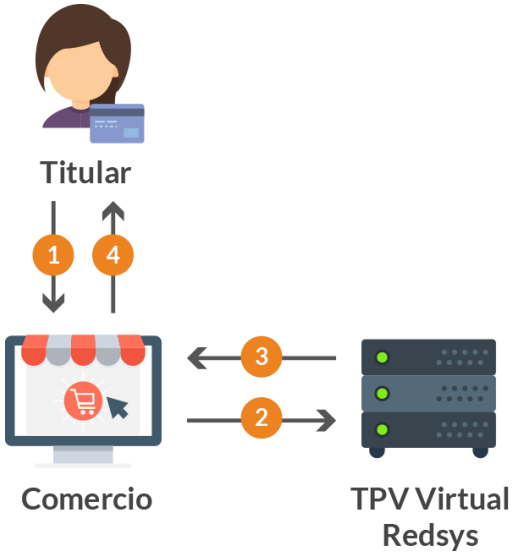
Término	Definición
	tarjetas, y garantiza prácticas seguras que reducen el riesgo de fraude y violaciones de seguridad.

3. Referencias

- Documentación de Integración con el SIS
- TPV-Virtual Guía SIS.
- TPV-Virtual Parámetros Entrada-Salida.xlsx
- Especificaciones COF ECOM
- https://pagosonline.redsys.es/desarrolladores-inicio

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 adquirir en el sitio web del comercio.
2. El **comercio** redirige la sesión del navegador del cliente a la URL del TPV Virtual de Redsys, donde el cliente introduce los datos de su tarjeta en un entorno seguro.
3. El **TPV Virtual** comunica al comercio el resultado de la operación.
4. **Redsys** devuelve la sesión del navegador del cliente al **comercio** para que continúe navegando en tu tienda web y pueda visualizar en esta misma tienda también el resultado final de la operación.

3. Envío de petición al Tpv virtual

El comercio envía al TPV Virtual los datos de la solicitud de pago codificados en UTF-8, a través del navegador del titular. Para ello, deberá preparar un formulario con los siguientes campos:

- **Ds_SignatureVersion**: Valor constante que indica la versión de firma que se está utilizando.
- **Ds_MerchantParameters**: Cadena en formato JSON que contiene todos los parámetros de la petición codificada en Base 64 URL-safe, para conocer el listado completo de parámetros disponibles, consulte el documento *TPV Virtual Parámetros Entrada-Salida.xlsx*.
- **Ds_Signature**: Firma digital de los datos enviados. Es el resultado del HMAC SHA512 de la cadena JSON codificada en Base 64 enviada en el parámetro anterior.

Este formulario debe enviarse a una de las siguientes URLs, dependiendo del entorno en el que se desee operar:

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

4. Montar la cadena de datos de la petición (Ds_MerchantParameters)

Para iniciar una solicitud de pago, el comercio debe generar una cadena en formato **JSON** que contenga todos los datos requeridos por el sistema. Los nombres de los parámetros incluidos en dicha cadena deben seguir una nomenclatura estandarizada, utilizando **mayúsculas** o el formato **CamelCase** (por ejemplo: DS_MERCHANT_AMOUNT o Ds_Merchant_Amount).

En aquellos casos en los que el comercio implemente operativas especiales, como el **Pago por Referencia** (también denominado **Pago 1-Clic**), será necesario incorporar los parámetros específicos correspondientes dentro del objeto JSON.

El conjunto completo de parámetros disponibles para incluir en una solicitud se encuentra detallado en el documento técnico **"TPV-Virtual Parámetros Entrada-Salida.xlsx"**.

Ejemplo de Objeto JSON Minificado (sin datos de tarjeta):

A continuación, se presentan un ejemplo del objeto JSON minificado utilizado en una petición de pago:

Ejemplo sin envío de datos de tarjeta:

{"DS_MERCHANT_AMOUNT":"999","DS_MERCHANT_ORDER":"1234567890","DS_MERCHANT_MERCHANTCODE":"999008881","DS_MERCHANT_CURRENCY":"978","DS_MERCHANT_TRANSACTIONTYPE":"0","DS_MER

CHANT_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"}

Una vez construida la cadena JSON con todos los campos requeridos, es imprescindible codificarla utilizando el formato **Base64 URL-safe**, asegurándose de que el resultado no contenga retornos de carro ni caracteres no válidos. Esta codificación garantiza la integridad de los datos durante su transmisión a través del navegador del cliente o comprador, evitando posibles alteraciones o corrupciones en el contenido original.

Ejemplo de Objeto JSON codificado en Base64 URL-safe (sin datos de tarjeta)::

eyJEU19NRVJDSEFOVF9BTU9VTlQiOiI5OTkiLCJEU19NRVJDSEFOVF9PUkRFUiI6IjEyMzQ1Njc4OTAiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9DVVJSRU5DWSI6Ijk3OCIsIkRTX01FUkNIQU5UX1RSQU5TQUNUSU9OVFlQRSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCI6IjEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVFVSTCI6Imh0dHA6XC9cL3d3dy5wcnVlYmEuY29tXC91cmxOb3RpZmljYWNpb24ucGhwIiwiRFNfTUVSQ0hBTlRfVVJMT0siOiJodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsT0sucGhwIiwiRFNfTUVSQ0hBTlRfVVJMS08iOiJodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsS08ucGhwIn0

La cadena resultante será el valor del parámetro **Ds_MerchantParameters**.

Nota: *Puede utilizar librerías de ayuda para la generación de este campo, ver Anexo*

5. Identificar la versión de algoritmo de firma a utilizar (Ds_SignatureVersion)

En cada transacción de pago, el comercio está obligado a especificar la versión exacta del algoritmo de firma empleado. Actualmente, se utiliza el identificador **HMAC_SHA512_V2** como referencia estándar para todas las solicitudes. En consecuencia, dicho valor debe asignarse al parámetro Ds_SignatureVersion, garantizando así la correcta interpretación y validación de la firma por parte del sistema receptor.

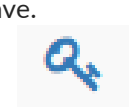
6. Firmar los datos de la petición (Ds_Signature)

6.1.1 Obtener Clave de Comercio

Pasos para obtener la clave del comercio:

1. Accede al Portal del TPV Virtual con tus credenciales.
2. Dirígete a la opción “Configuración del Comercio”.
3. Dependiendo del tipo de usuario, podrás encontrar la clave, en alguna de las siguientes 2 opciones:

   3.1 En el listado de terminales, en la columna de la derecha, busca el icono de una llave.



3.2 En el detalle del terminal, busca el botón “Ver Clave de Firma”

4. Haz clic en ese icono y se abrirá una ventana emergente que mostrará tu clave del terminal para la firma SHA-512.

Nota: *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.*

6.1.2 Generación de Clave de Firma

Para cada operación de pago, se debe generar una clave de firma única y específica. Esta clave se obtiene mediante un proceso de cifrado **AES en modo CBC (Cipher Block Chaining)**, utilizando como **vector de inicialización (IV)** un bloque de ceros.

El proceso de cifrado se realiza entre la **clave secreta del comercio** (en el apartado anterior) y el **número de pedido** correspondiente a la operación. El resultado de este cifrado constituye la clave de firma que se utilizará exclusivamente para esa transacción.

El proceso para obtener esta clave derivada es el siguiente:

1. Preparación de la clave del comercio:

   La clave proporcionada por Redsys (obtenida en el apartado anterior) debe tener una longitud de 16 caracteres.

   - Si la clave es más larga, se debe quedar con los 16 caracteres primeros.
   - Si la clave tiene menos de 16 caracteres, rellenarás por la derecha con ceros hasta completar los 16.

   Ejemplo de la clave del comercio con 16 caracteres:

   sq7HjrUOBfKmC576

2. Obtención de la clave de Firma:

Cifrar la clave del comercio en AES en modo CBC con los siguientes datos:

- Utilizar un vector de Inicialización todo a ceros.
- Diversificar con del Ds_Merchant_Order.

Ejemplo de la obtención de la clave de firma en BASE64:

Vector de Inicialización: 0000000000000000

Ds_Merchant_Order : 1234567890

Clave de Firma Obtenida: RWt3/IPTzYRMXsQtkiGRKg==

Esta cadena codificada es la clave derivada que se utilizará para firmar la operación.

Nota: *Usar un vector de ceros como IV no compromete la firma de la operación, ya que esto es sólo un paso accesorio. El verdadero algoritmo de firma es HMAC SHA-512.*

6.1.3 Firma HMAC SHA 512 V2

Deberás calcular el HMAC SHA-512 del valor del parámetro que vas a firmar (*Ds_MerchantParameters*) utilizando la clave que has obtenido en el paso anterior.

Ejemplo de Obtención del Ds_Signature

1. Valor *Ds_MerchantParameters* en Base64URL-Safe obtenido en este apartado:

   eyJEU19NRVJDSEFOVF9BTU9VTlQiOiI5OTkiLCJEU19NRVJDSEFOVF9PUkRFUiI6IjEyMzQ1Njc4OTAiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9DVVJSRU5DWSI6Ijk3OCIsIkRTX01FUkNIQU5UX1RSQU5TQUNUSU9OVFlQRSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCI6IjEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVFVSTCI6Imh0dHA6XC9cL3d3dy5wcnVlYmEuY29tXC91cmxOb3RpZmljYWNpb24ucGhwIiwiRFNfTUVSQ0hBTlRfVVJMS08iOiJodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsT0sucGhwIiwiRFNfTUVSQ0hBTlRfVVJMS08iOiJodHRwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsS08ucGhwIn0

2. Clave de Firma a utilizar obtenida en este apartado.

   RWt3/IPTzYRMXsQtkiGRKg==

3. Con estos datos calculamos el HMAC SHA512, lo codificamos en BASE54URL-Safe y será el valor Ds_Signature:

   Vjo02eSWq249IeZZp3R-ArFnGLhKY0OuzDDlx1BuVtZDC2yhczA7_11uZhsYzLZBCMFAz8u8uzGDX3AErHKmmw

Nota: *Puede utilizar librerías de ayuda para la generación de este campo, ver Anexo*

6.1.4 Datos de Envío

A continuación, se detalla cómo se debe estructurar y mostrar el formulario de pago utilizando los datos obtenidos en los apartados anteriores. Este formulario debe incorporar los parámetros codificados y firmados previamente, garantizando su integridad y autenticidad. La implementación debe asegurar que los datos se transmitan de forma segura y que el formulario sea compatible con el entorno del cliente/comprador:

Ejemplo de formulario de pago:

```
<form name="from" action="https://sis-t.redsys.es:25443/sis/realizarPago" method="POST">

<input type="hidden" name="Ds_SignatureVersion" value="HMAC_SHA512_V2"/>

<input type="hidden" name="Ds_MerchantParameters"
value="eyJEU19NRVJDSEFOVF9BTU9VTlQiOiI5OTkiLCJEU19NRVJDSEFOVF9PUkRFUiI6IjEyMzQ1Njc4OTAiLCJEU19NRVJDS
EFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9DVVJSRU5DWSI6Ijk3OCIsIkRTX01FUkNIQU5U
X1RSQU5TQUNUSU9OVFlQRSI6IjAiLCJEU19NRVJDSEFOVF9URVJNSU5BTCI6IjEiLCJEU19NRVJDSEFOVF9NRVJDSEFOVFVST
CI6Imh0dHA6XC9cL3d3dy5wcnVlYmEuY29tXC91cmxOb3RpZmljYWNpb24ucGhwIiwiRFNfTUVSQ0hBTlRfVVJMT0siOiJodH
RwOlwvXC93d3cucHJ1ZWJhLmNvbVwvdXJsT0sucGhwIiwiRFNfTUVSQ0hBTlRfVVJMS08iOiJodHRwOlwvXC93d3cucHJ1ZW
JhLmNvbVwvdXJsS08ucGhwIn0"/>

<input type="hidden" name="Ds_Signature" value="Vjo02eSWq249IeZZp3R-
ArFnGLhKY0OuzDDlx1BuVtZDC2yhczA7_11uZhsYzLZBCMFAz8u8uzGDX3AErHKmmw"/>

</form>
```

Ejemplo de Construcción del Objeto JSON Ds_MerchantParameters con **datos de tarjeta**:

```
{"DS_MERCHANT_AMOUNT":"145","DS_MERCHANT_ORDER":"1446068581","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"
,"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":"151
2","DS_MERCHANT_CVV2":"123"}
```

Nota: En el envío del número de tarjeta (DS_MERCHANT_PAN) debe incluirse la numeración completa. Los asteriscos utilizados en el ejemplo son únicamente ilustrativos.

Nota de Seguridad: Siempre que el comercio gestione datos sensibles de tarjeta, es obligatorio cumplir con los requisitos establecidos por el estándar PCI-DSS, que regula el tratamiento seguro de información de tarjetas de pago.

Ejemplo de Construcción del Objeto JSON Ds_MerchantParameters con **datos adicionales del protocolo EMV3DS para cumplimiento de PSD2**:

```
{"DS_MERCHANT_ORDER":"1552565870","DS_MERCHANT_MERCHANTCODE":"999008881","DS_MERCHAN
T_TERMINAL":"999","DS_MERCHANT_CURRENCY":"978","DS_MERCHANT_TRANSACTIONTYPE":"0","DS_ME
RCHANT_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"}}}
```

Nota: En el protocolo EMV3DS el comercio tiene la posibilidad de incluir, de forma opcional, información adicional relativa al titular de la tarjeta y a la transacción en la solicitud de pago. Estos datos permiten al **emisor de la tarjeta** realizar una evaluación más precisa del riesgo asociado a la operación, lo que facilita la toma de decisiones sobre la necesidad de aplicar **Autenticación Reforzada de Cliente (SCA)** conforme a la normativa PSD2.

Ejemplo de Construcción del Objeto JSON Ds_MerchantParameters con **envió de exención:**

*{"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**"}*

Nota: La normativa PSD2 establece una serie de excepciones, llamadas exenciones, para evitar la aplicación de **SCA** en ciertas transacciones, destinado a simplificar el proceso de pago en situaciones en las que el riesgo de fraude es bajo o se dan una serie de circunstancias. Para más información sobre las exenciones puede consultar en Ver información sobre normativa PSD2 .

4. Resultado de la petición

Una vez realizada la petición de pago desde el navegador del cliente en el sitio web del comercio, el servidor de la pasarela de pago toma el control del flujo de navegación. Durante este proceso, se presentan al cliente una serie de pantallas que permiten:

- Introducir los datos de su tarjeta.
- Autenticarse conforme a los requisitos de seguridad (por ejemplo, SCA).
- Visualizar los detalles de la operación, incluyendo el importe de la compra.

Al concluir el proceso, se producen dos acciones clave:

- Notificación técnica al comercio.
- Redirección del cliente al sitio web del comercio.

7. Tratamiento de la notificación HTTP POST

Una vez gestionada la transacción, el TPV Virtual puede informar al servidor del comercio del resultado mediante una notificación on-line, que consiste en una conexión directa desde Redsys al servidor del comercio.

Características de la Notificación On-line

- Es una función opcional, configurable desde el Portal de Administración del TPV Virtual.
- Permite al comercio recibir el resultado de la transacción en tiempo real, una vez que el cliente ha completado el proceso de pago.
- La notificación se envía mediante una solicitud HTTP POST a la URL especificada en el parámetro Ds_Merchant_MerchantURL de la petición de pago en el Ds_Merchant_Parameters. Este parámetro también es configurable en el Portal de administración y en ese caso no haría falta mandarlo en cada petición.
- La comunicación se realiza de forma paralela e independiente al flujo de navegación del cliente.

Contenido de la Notificación POST

La notificación incluye los siguientes campos:

- Ds_SignatureVersion: Versión del algoritmo de firma utilizado.
- Ds_MerchantParameters: Cadena JSON con los parámetros de respuesta, codificada en Base64 URL-safe y en formato UTF-8. La lista completa de parámetros está disponible en el documento “TPV-Virtual Parámetros Entrada-Salida.xlsx”.

- Ds_Signature: Firma generada a partir de la cadena codificada en Ds_MerchantParameters. El comercio debe validar esta firma para garantizar la integridad y autenticidad de los datos recibidos.

Tipos de Notificación

- Síncrona: El resultado se envía primero al comercio, y luego se muestra al cliente. Aunque el comercio no procese correctamente la notificación, el resultado de la operación se mantiene.
- Asíncrona: El resultado se comunica simultáneamente al comercio y al cliente. Al igual que en la modalidad síncrona, el resultado de la operación no se ve afectado por errores en el procesamiento por parte del comercio.

Resultado de la Operación

El resultado de la operación se informará mediante el parámetro Ds_Response contenido dentro del Ds_Merchant_Parameters.

Operaciones Autorizadas y su valor de Ds_Response:

0000 a 0099	**Transacción autorizada para pagos y preautorizaciones**
400	**Transacción autorizada para anulaciones**
900	**Transacción autorizada para devoluciones y confirmaciones**

Cualquier otro valor en el Ds_Repsonse indica que la operación no se ha completado correctamente, puede ver la lista completa de valores en Valores de Entrada y Salida en el apartado "**Códigos recibidos en el parámetro Ds_response"**

Nota: Asegúrate de que tu servidor esté preparado para recibir y procesar correctamente tanto la notificación POST y que las URL esté correctamente configurada y accesible.

8. Tratamiento del retorno de navegación

Una vez realizada la petición de pago y notificado al comercio el resultado, se muestra al comprador el resultado del pago. Redsys ofrece varias opciones para que el comercio elija la que se adapte a sus necesidades:

- **Mostrar Recibo**. Redsys muestra el recibo indicando al comprador como se ha finalizado su pago y desde aquí se redirigirá al comercio.
- **No Mostrar Recibo**. Redsys ofrece la opción de una redirección automática al comercio cuando el pago finalice.

Mostrar Recibo

Cuando la operación se complete en nuestro servidor, el titular podrá ver un cuadro de confirmación de la operación, denominado «Recibo Redsys», y que será muy parecido a este:

Una vez que el cliente ha realizado el proceso de pago, al pulsar sobre «Continuar», se le redirigirá de nuevo a tu tienda usando la URL comunicada como parámetro en la llamada inicial al TPV Virtual:

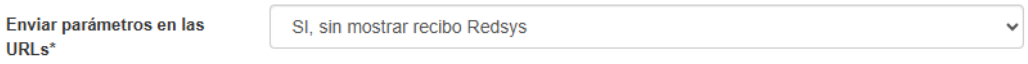
- Ds_Merchant_URLOK. Se utilizará este parámetro definido en el Ds_Merchant_Parameters de la petición cuando la operación haya finalizado correctamente.
- Ds_Merchant_URLKO. Se utilizará este parámetro definido en el Ds_Merchant_Parameters de la petición cuando la operación NO haya finalizado correctamente.

Opcionalmente se puede configurar para recibir los parámetros en estas URLs por GET, seleccionando en la configuración del comercio en el Portal de administración *Enviar Parámetros en las URLs= SI*.

1. **Nota**: Si no se envían estos parámetros se utilizarán los definidos en la Configuración del TPV Virtual en el Portal de administración.

2. **No Mostrar Recibo**

   Opcionalmente se podrá configurar en el Portal de administración del TPV Virtual de tal manera que no se muestre el «Recibo Redsys» y en ese caso se redirecciona directamente a la URL OK o la URL KO.



   Nota: Si no se envían estos parámetros se utilizarán los definidos en la Configuración del TPV Virtual.

   Nota Importante: Nunca utilices los datos enviados vía HTTP GET para validar tu pedido, y utiliza sólo los enviados en la notificación HTTP POST tal y como se ha explicado anteriormente; ya que el cliente podría cerrar la pantalla de pago una vez vea que la operación se ha realizado correctamente y nunca sabrías el resultado de la operación.

5. Entorno de pruebas

El comercio puede utilizar el entorno de pruebas (test) para realizar todas las verificaciones necesarias antes de proceder con la implantación en el entorno de producción. Este entorno permite simular transacciones sin impacto contable, facilitando la validación de la integración técnica.

En este apartado se proporcionan datos genéricos de prueba que pueden ser utilizados por cualquier cliente. Si el comercio desea realizar pruebas utilizando sus propios datos, deberá contactar con su entidad bancaria para obtener las credenciales específicas de acceso.

URLs de acceso al entorno de pruebas de Redsys:

Operativas por redirección (formulario de pago)
https://sis-t.redsys.es:25443/sis/realizarPago
Portal de administración del entorno de pruebas
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 del comercio: Aquí se debe utilizar la configurada por el comercio en el portal de administración (ejemplo: sq7HjrUOBfKmC576ILgskD5srU870gJ7)
- Tarjetas de Pruebas. La diferentes tarjetas y el comportamiento de ellas se pueden consultar en Tarjetas y entorno de pruebas

6. Errores en el proceso de integración

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).
- 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.

Error en el Ds_Merchant_Parameters (error SIS0430)

Este error ocurre porque en el parámetro Ds_Merchant_Parameters se ha encontrado algún error:

- El contenido debe estar codificado en formato Base64 URL-Safe, lo que implica que no deben aparecer caracteres como =, + o /. Estos caracteres son propios del Base64 estándar y no son válidos en este contexto.
- La cadena JSON original debe estar codificada en **UTF-8**.

Número de repetido (error SIS0051)

El error relacionado con la repetición de números de pedido suele producirse cuando en la integración del comercio reutiliza identificadores de pedido únicamente en casos de transacciones denegadas o autorizadas, pero no contempla aquellas operaciones que han quedado incompletas o interrumpidas. Esta situación puede generar conflictos en el sistema de la pasarela de pago. Para evitar este problema, se plantean dos alternativas:

- Solicitar configuración especial al servicio de soporte del TPV. Es posible solicitar que el TPV se configure para permitir la reutilización de números de pedido, con una única operación autorizada por día con el mismo número de pedido y sin límite para operaciones denegadas con el mismo identificador.
- Generar siempre números de pedido únicos. Se recomienda implementar una lógica que garantice la unicidad del número de pedido, no solo para transacciones autorizadas y denegadas, sino también para aquellas que no hayan finalizado correctamente. O que se hayan quedado en estado intermedio o sin respuesta tras un tiempo determinado.

Nota: Utilizar identificadores de pedido generados dinámicamente (por ejemplo, con marcas de tiempo o UUIDs) puede ayudar a evitar colisiones y facilitar el seguimiento de operaciones.

No Recepción de Respuesta "On-line" en el TPV Virtual

Si el comercio no está recibiendo la respuesta "on-line" tras una operación de compra, es importante verificar la configuración del parámetro Ds_Merchant_MerchantURL, ya que es en esta dirección donde el TPV Virtual envía la notificación HTTP con el resultado de la transacción.

Puntos clave a revisar en la integración:

- Verificar la URL de notificación (Ds_Merchant_MerchantURL): Asegúrate de que la URL esté correctamente definida en la petición de pago, este parámetro debe ir montando en el Ds_Merchant_Parameters
- El servidor del comercio debe estar preparado para recibir solicitudes HTTP POST.

Consultar el resultado en el Portal de Administración del TPV Virtual:

Si no se recibe la notificación, el comercio puede acceder al portal de administración y revisar en la sección "Notificaciones" o en el detalle de la operación "Consultas".

Nota: La respuesta del servidor sigue el protocolo estándar HTTP. Para interpretar errores genéricos (como códigos 4xx o 5xx), se puede consultar la documentación oficial de RFC 2616 - HTTP/1.1 Status Codes.

Nota: Si el comercio no dispone de un servidor correctamente configurado para recibir notificaciones HTTP, y necesita asistencia técnica personalizada, debe contactar directamente con su entidad bancaria para solicitar consultoría especializada.

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. Si el problema persiste se deberá revisar el email configurado en el Portal de Administración.

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

Es importante hacer un buen uso de las URLs de redirección definidas como Ds_Merchant_UrlOK y Ds_Merchant_UrlKO, están no deben utilizarse para recibir los parámetros con el resultado de la operación. Estas URLs están destinadas exclusivamente a:

- Redirigir al titular de la tarjeta de vuelta al sitio web del comercio tras completar el proceso de pago.
- Mostrar información comercial o un recibo de compra, si el comercio lo tiene configurado.

7. Preguntas Frecuentes

Soy un comercio y necesito conocer la clave de comercio para la encriptación de las peticiones

Puedes ver la información de cómo obtener esta clave en el apartado Obtener Clave de Comercio

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, deberá acceder al apartado de "Comercio" y establecer unas nuevas URLs.
- Si el comercio envía las URLs en cada transacción, debe incluirlas en los campos Ds_Merchant_UrlOK y Ds_Merchant_UrlKO dentro del objeto Ds_Merchant_Parameters. En este caso, cualquier modificación debe realizarse directamente en el formulario de pago que genera el comercio.

ANEXO 1: Apis de Firma

1. Librerías de ayuda para el cálculo de la firma

En los apartados anteriores se ha descrito la forma de acceso al SIS utilizando la entrada REST y el sistema de firma basado en HMAC SHA512. En este apartado se explica como 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.

1.1.1 Librería PHP

A continuación, se presentan los pasos que debe seguir un comercio para la utilización de la librería PHP proporcionada por Redsys, compatible con los algoritmos *HMAC SHA-512_V2*.

HMAC_SHA512_V2

1. Importar el fichero principal de la librería *signature.php* donde haya sido ubicado en el proyecto, en este caso, estos archivos están ubicados en la carpeta *signatureUtils*, por lo que habría que importarlo tal y como se muestra a continuación:

```
// Para API HMAC SHA 512
include_once 'signatureUtils/signature.php';
include_once 'signatureUtils/utils.php';
```

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

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:

```
$data = array(
    "DS_MERCHANT_AMOUNT" => $amount,
    "DS_MERCHANT_ORDER" => $id,
    "DS_MERCHANT_MERCHANTCODE" => $fuc,
    "DS_MERCHANT_CURRENCY" => $moneda,
    "DS_MERCHANT_TRANSACTIONTYPE" => $trans,
    "DS_MERCHANT_TERMINAL" => $terminal,
    "DS_MERCHANT_MERCHANTURL" => $url,
    "DS_MERCHANT_URLOK" => $urlOK,
    "DS_MERCHANT_URLKO" => $urlKO,
);
```

Por último, para calcular el parámetro Ds_MerchantParameters, debemos convertir este array en un json y codificarlo en Base64UrlSafe. Para esto podemos utilizar el

siguiente código. Tanto la clase estática Utils como la clase estática Signature se importan con la línea del punto 1.

```
$params = Utils::base64_url_encode_safe(json_encode($data));
```

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, los parámetros obtenidos en el punto anterior, y el número de pedido de la petición (DS_MERCHANT_ORDER), tal y como se muestra a continuación:

```
$kc = "sq7HjrUOBfKmC576ILgskD5srU870gJ7";
$signature = Signature::createMerchantSignature($kc, $params, $id);
```

4. Una vez obtenidos los valores de los parámetros Ds_MerchantParameters y Ds_Signature, se debe realizar la petición REST con dichos valores y el parámetro Ds_SignatureVersion.

```
$merchantParameters = array(
    "Ds_MerchantParameters" => $params,
    "Ds_Signature" => $signature,
    "Ds_SignatureType" => "HMAC_SHA512_V2",
);
```

1.1.2 1.2 Librería JAVA

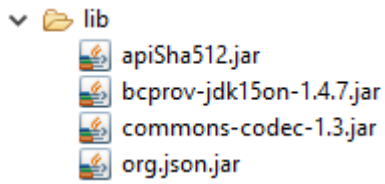
A continuación, se presentan los pasos que debe seguir un comercio para la utilización de la librería JAVA proporcionada por Redsys. Compatible con los algoritmos HMAC SHA-512:

HMAC SHA-512

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

```
import sis.redsys.api.Signature;
import sis.redsys.api.Utils;
```

El comercio debe incluir en la vía de construcción del proyecto todas las librerías(JARs).



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:

```
JSONObject data = new JSONObject();
data.put("DS_MERCHANT_AMOUNT", amount);
data.put("DS_MERCHANT_ORDER", id);
data.put("DS_MERCHANT_MERCHANTCODE", fuc);
data.put("DS_MERCHANT_CURRENCY", moneda);
data.put("DS_MERCHANT_TRANSACTIONTYPE", trans);
data.put("DS_MERCHANT_TERMINAL", terminal);
data.put("DS_MERCHANT_MERCHANTURL", url);
data.put("DS_MERCHANT_URLOK", urlOK);
data.put("DS_MERCHANT_URLKO", urlKO);
```

Por último, para calcular el parámetro Ds_MerchantParameters, debemos convertir este JSONObject en un String y codificarlo en Base64UrlSafe. Para esto podemos utilizar el siguiente código.

```
final String params =
Utils.encodeB64UrlSafeString(data.toString());
```

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, los parámetros obtenidos en el punto anterior, y el número de pedido de la petición (DS_MERCHANT_ORDER), tal y como se muestra a continuación:

```
String key = "sq7HjrUOBfKmC576ILgskD5srU870gJ7";
final String signature =
Signature.createMerchantSignature(key, params, id);
```

4. Una vez obtenidos los valores de los parámetros Ds_MerchantParameters y Ds_Signature, se debe realizar la petición REST con dichos valores y el parámetro Ds_SignatureVersion.

```
JSONObject merchantParameters = new JSONObject();
merchantParameters.put("Ds_MerchantParameters", params);
merchantParameters.put("Ds_Signature", signature);
merchantParameters.put("Ds_SignatureVersion",
"HMAC_SHA512_V2");
```

1.1.3 1.3 Librería .NET

Debe importar la librería de ayuda para API HMAC-SHA512 para seguir estos pasos.

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);
```

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 la petición REST con dichos valores y el parámetro Ds_SignatureVersion.

2. Librerías de ayuda respuesta petición de pago

En los apartados anteriores se ha descrito la forma de acceso al SIS utilizando conexión REST. 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 en la respuesta del servicio REST.

El uso de las librerías suministradas por Redsys es opcional, si bien simplifican los desarrollos a realizar por el comercio.

2.1.1 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. Esta notificación de respuesta se recibirá firmada con el mismo algoritmo con el que originalmente fue firmada la petición original. Compatible con los algoritmos HMAC SHA-512:

HMAC SHA-512

1. Importar el fichero principal de la librería *apiRedys512.php* donde haya sido ubicado en el proyecto, en este caso, estos archivos están ubicados en la carpetas *redsysHMAC512_API_PHP_7.0.0*, por lo que habría que importarlo tal y como se muestra a continuación:

```
// Para API HMAC SHA 512
include_once 'redsysHMAC512_API_PHP_7.0.0/apiRedsys512.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, en este caso llamada *RedsysAPI512*:

```
$miObj = new RedsysAPI512; // Para API HMAC SHA 512
```

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

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

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

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

   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 **¡Error! No se encuentra el origen de la referencia.**). Para l levar 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 = "sq7HjrUOBfKmC576ILgskD5srU870gJ7";
$firmaCalculada = $miObj->createMerchantSignatureNotif($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($firmaCalculada === $firmaRecibida) {
   echo "FIRMA OK. Realizar tareas en el servidor";
} else {
   echo "FIRMA KO. Error, firma inválida";
}
```

2.1.2 2.2 Librería JAVA

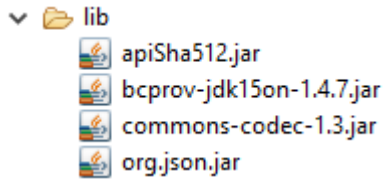
A continuación, se presentan los pasos que debe seguir un comercio para la utilización de la librería JAVA proporcionada por Redsys. Compatible con los algoritmos *HMAC SHA-512*:

HMAC SHA-512

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

```
<%@page import="sis.redsys.api.ApiMacSha512"%> // Para API HMAC SHA 512
```

El comercio debe incluir en la vía de construcción del proyecto todas las librerías(JARs) del tipo de firma que haya elegido.



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

```
ApiMacSha512 miObj = new ApiMacSha512(); // Para API HMAC SHA 512
```

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

```
String version = request.getParameter("Ds_SignatureVersion");
String params = request.getParameter("Ds_MerchantParameters");
String firmaRecibida = 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 = 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 **¡Error! No se encuentra el origen de la referencia.**). Para l levar 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 = 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:

```
String claveModuloAdmin = "sq7HjrUOBfKmC576ILgskD5srU870gJ7";
String firmaCalculada =
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(firmaCalculada.equals(firmaRecibida)) {
        System.out.println("FIRMA OK. Realizar tareas en el servidor");
} else {
```

```
            System.out.println("FIRMA KO. Error, firma inválida");
        }
```

2.1.3 2.3 Librería .NET

Debe importar o bien la librería de ayuda para API HMAC-SHA265 o la de HMAC-SHA512 para seguir estos pasos. Todos los pasos son comunes entre ambas pero solo puedes tener importadas una de ellas a la vez a diferencia del resto de lenguajes, donde se pueden importar al mismo tiempo sin problemas.

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:

```
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.

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";
}
```