



TPV-Virtual

Manual Integración REST

Versión: 4.0.1

Fecha: 17/10/2025

Referencia: RS.TE.CEL.MAN.0037



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	01/10/2018	TODO	Versión Inicial
1.1	17/12/2018	Anexo 7.7	3DSecure 2.0
1.1	25/01/2019	Anexo 7.9	Integración PSP
1.2	25/01/2019	Apartado 7.2 Peticiones de Confirmación/ Devolución	Añadido soporte para anulación de autorización, al enviar tipo 9 en TRANSACTIONTYPE
1.3	06.02.2019	Varios puntos	Se incluye código de respuesta en operaciones que requieren challenge: <i>Ds_response=9568</i> En los datos 3DSecure se indica siempre la versión de protocolo en el campo <i>protocolVersion</i> Se incluyen esquemas de secuencias para aclarar el flujo. Se añade el punto 7.8 Integración PSP's
1.4	11/02/2019	Varios puntos	Se incluyen todos los parámetros 3DSecure 1.0.2 y un ejemplo.
1.5	18/02/2019	Varios puntos	MPI Externo Mover el punto 7.7 al punto 5 Diferentes puntualizaciones en el apartado 5 de Autenticación 3DSecure Incluido apartado de timeout
2.0	27/03/2019	Varios puntos	Añadida la información sobre EMV3DS y PSD2. Añadida la información de transacciones con DCC. Añadida la información de Autenticaciones con DCC. Modificaciones del apartado Integración para PSP

			<p>Modificaciones del apartado ¿Operas por PUCE?</p> <p>Modificaciones del apartado MPI Externo EMV3DS</p> <p>Añadida referencia a los nuevos documentos TPV-Virtual GuiaErroresSIS.xlsx y TPV-Virtual Parámetros Entrada-Salida.xlsx</p>
2.1	12/04/2019	Puntos 5, 9, 10, 13	<p>Nota importante sobre integración EMV3DS</p> <p>Añadidas tarjetas de pruebas para EMV3DS</p> <p>Puntualización para operaciones PUCE</p> <p>La hoja de cálculo de los errores SIS, se modifica para incluirla en la hoja de cálculo Parámetros de entrada-salida</p>
2.2	25/07/2019	Puntos 5,8, 10	<p>Operaciones COF y MIT</p> <p>Código de error cambio formato</p> <p>Parámetro creq</p>
2.3	04/10/2019	Todo el documento	<p>Se añaden adaptaciones para EVM3DS 2.2</p>
2.4	12/11/2019	Puntos 5.1, 5.2, 8, 9 y 10.1.3	<p>Se marcan los avances para las funcionalidades EMV 3DS.</p> <p>Se reorganiza el punto 10.1.3. Flujo envío petición para PSP</p>
2.5	24/01/2020	Puntos 5,6,8,9,11	<p>PUNTOS 5, 6 añadido el parámetro Ds_Card_PSD2</p> <p>PUNTOS 5, MIT y tokenización</p> <p>Peticiones con EXENCIONES.</p> <p>Aclaración y ejemplos para funcionalidades de avances EMV3DS.</p> <p>Modificación tarjetas de pruebas para los avances EMV3DS.</p>
2.6	10/02/2020	Puntos 6,7 y 11	<p>Definición de nuevos parámetros opcionales: "cambioBCE" y</p>

			"porcentajeSobreBCE" para operaciones DCC con monedas de la Unión Europea
3.0	01/10/2020	Todo el documento	<p>Se elimina la marca "avance" en las funcionalidades afectadas por PSD2</p> <p>Se reestructuran los puntos en los que se explica el flujo de la petición REST al tpv virtual y de la petición con autenticación del titular.</p> <p>El punto 10 queda pendiente de especificaciones de las marcas.</p> <p>Se crea un Anexo con las librerías de ayuda.</p> <p>Se añaden tarjetas para realizar pruebas con todas las marcas de tarjetas.</p>
3.0.1	26/11/2020	Punto 6 Punto 6.6	<p>Se incluye nota IMPORTANTE: operación que inicia flujo en versión 2 y en el proceso de autenticación cambia el flujo a versión 1. Se añade tarjeta de prueba para este caso.</p> <p>Se incluyen ejemplos de peticiones 3DS v 1.0</p>
3.0.2	09/01/2024	Punto 6 Punto 7 Punto 12	<p>Eliminación de 3DSecure versión 1</p> <p>Se incluye texto de ejemplo para la operativa DCC BCE.</p> <p>Eliminación de casos de prueba de 3DSecure versión 1</p>
3.0.3	31/01/2024	Punto 12	Añadida tarjeta de pruebas DCC en Libras
3.0.4	04/11/2024	Punto 11.2	Sustituir "authenticationValue:" por "authenticationValue:"
3.0.5	12/02/2025	Punto 11.1.3	Cifrado de campos XpayDecodedData y TokenData para PSP
3.0.6	11/03/2025	Puntos 11.1, 11.1.2, 11.1.3, 11.1.4 y 11.1.5	Se sustituye la versión V1 de cifrado y firma de PSP por la nueva versión V2.

4.0	20/03/2025	Puntos 3.3, 4 y Anexos	Nuevo Tipo de firma HMAC SHA512
4.0.1	17/10/2025	Puntos 11.1.2, 11.1.3 y 11.1.5	Nuevos ejemplos cifrado y firma de PSP

ÍNDICE

1. INTRODUCCIÓN	9
OBJETIVO	9
DEFINICIONES, SIGLAS Y ABREVIATURAS	9
REFERENCIAS	9
2. ¿QUÉ PERMITE LA INTEGRACIÓN CON INTERFAZ REST?	10
3. ESTRUCTURA DE UNA PETICIÓN REST	11
IDENTIFICAR LA VERSIÓN DE ALGORITMO DE FIRMA A UTILIZAR (Ds_SIGNATUREVERSION)	11
MONTAR LA CADENA DE DATOS DE LA PETICIÓN (Ds_MERCHANTPARAMETERS)	11
FIRMAR LOS DATOS DE LA PETICIÓN (Ds_SIGNATURE)	12
4. ESTRUCTURA DE RESPUESTA REST	15
5. TRANSACCIONES DIRECTAS (SIN AUTENTICACIÓN)	17
6. TRANSACCIONES CON AUTENTICACIÓN EMV 3DS	19
PASOS PARA REALIZAR UNA TRANSACCIÓN CON AUTENTICACIÓN	19
FLUJO AUTORIZACIÓN CON AUTENTICACIÓN EMV3DS FRICTIONLESS	20
FLUJO AUTORIZACIÓN CON AUTENTICACIÓN EMV3DS CHALLENGE	21
EJEMPLO DE PETICIONES PARA REALIZAR UNA TRANSACCIÓN CON AUTENTICACIÓN EMV3DS	23
6.5.1 INICIAR PETICIÓN	23
6.5.2 EJECUCIÓN DEL 3DSMETHOD	24
6.5.3 PETICIÓN DE AUTORIZACIÓN CON DATOS EMV3DS	25
6.5.4 EJECUCIÓN DEL CHALLENGE	27
6.5.5 CONFIRMACIÓN DE AUTORIZACIÓN EMV3DS POSTERIOR AL CHALLENGE	28
7. TRANSACCIONES CON DCC	29
8. TRANSACCIONES DCC CON AUTENTICACIÓN	32
9. ADAPTACIONES PSD2	35
EJEMPLOS DE PETICIONES CON EXENCIONES.	36
MENSAJE INICIA PETICIÓN (PARA CONOCER LAS EXENCIONES PERMITIDAS AL COMERCIO)	36

MENSAJE TRATA PETICIÓN (CON EMV3DS)	37
MENSAJE TRATA PETICIÓN (SIN EMV3DS)	37
EJEMPLO TRANSACCIÓN MIT (MERCHANT INITIATED TRANSACTION)	38

10. FUNCIONALIDADES AVANZADAS 3RI Y OTA **39**

11. OTRAS INTEGRACIONES REST: PSPS/ MPI EXTERNO/ PUCE **40**

INTEGRACIÓN PARA PSP	40
11.1.1 CONFIGURACIÓN	40
11.1.2 SOLICITUD Y RECEPCIÓN DE CLAVES	40
11.1.3 ENVÍO DE PETICIÓN AL TPV VIRTUAL	40
11.1.4 RECEPCIÓN DEL RESULTADO	41
11.1.5 EJEMPLO DE PETICIONES	41
MPI/3DSSERVER EXTERNO	43
PUCE (AUTENTICACIÓN)	44

12. ENTORNO DE PRUEBAS **45**

13. EJEMPLOS DE TIPOS DE OPERACIÓN MÁS HABITUALES **49**

PETICIÓN DE PAGO/PREAUTORIZACIÓN (CON ENVÍO DE DATOS DE TARJETA SIN AUTENTICACIÓN)	49
PETICIÓN DE CONFIRMACIÓN/DEVOLUCIÓN/ANULACIÓN	49
PETICIÓN DE TOKENIZACIÓN (PAGO POR REFERENCIA - PAGO 1-CLIC)	49
PETICIÓN DE PAGO CON TOKENIZACIÓN (PAGO POR REFERENCIA - PAGO 1-CLIC)	50

14. TIMEOUT **51**

15. ERRORES FRECUENTES **52**

16. PREGUNTAS FRECUENTES **53**

ANEXO 1 **54**

1. LIBRERÍAS DE AYUDA PARA EL CÁLCULO DE LA FIRMA	54
1.1 LIBRERÍA PHP	54
1.2 LIBRERÍA JAVA	55
1.3 LIBRERÍA .NET	56
2. LIBRERÍAS DE AYUDA RESPUESTA PETICIÓN DE PAGO	57
2.1 LIBRERÍA PHP	57
2.2 LIBRERÍA JAVA	59
2.3 LIBRERÍA .NET	60

ANEXO 2 **62**

PETICIÓN CON FIRMA EN HMAC SHA-256 **62**

1. Introducción

Objetivo

Este documento recoge los aspectos técnicos necesarios para que un comercio realice la integración con el TPV Virtual utilizando un interfaz REST.

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.

Referencias

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

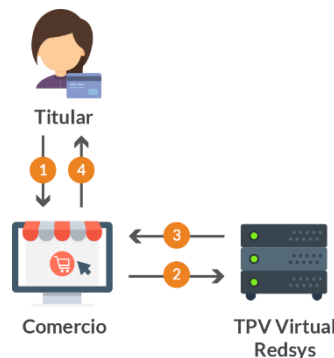
2. ¿Qué permite la integración con interfaz REST?

Esta forma de conexión permite a los comercios tener integrado el TPV Virtual dentro de su propia aplicación Web, así todo el proceso de pago se realiza en la misma web del comercio y el cliente no abandona en ningún momento este entorno web. Este modo de conexión también ofrece la posibilidad de autenticar al titular mediante el protocolo EM3DS, que dota de mayor seguridad a las compras.

Es importante tener en cuenta que en este tipo de integración es el comercio el que recoge los datos de tarjeta del cliente para procesar el pago y, por tanto, tiene afectación en su cumplimiento de PCI-DSS.

Los comercios también pueden utilizar este modo de conexión para integrar el TPV Virtual con su backoffice y realizar operaciones asociadas como devoluciones o confirmaciones de Preautorización. En este caso no es necesario que el comercio maneje los datos de tarjeta, por lo que si únicamente utiliza la integración REST para este tipo de operaciones no tendría afectación de cara a cumplimiento de PCI-DSS.

El esquema básico de un pago mediante integración REST sería el siguiente:



1. El titular selecciona los productos que desea comprar e introduce los datos de tarjeta en un formulario mostrado por el comercio.
2. El comercio envía los datos al pago al TPV virtual.
3. Una vez realizado el pago, el TPV virtual informa del resultado de la operación al comercio.
4. El comercio devuelve la información del resultado del pago al titular.

3. Estructura de una petición REST

El comercio envía una petición (POST) mediante la interfaz REST al Tpv Virtual. En esta petición se envía un JSON que incluye los siguientes parámetros:

- **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.
- **Ds_Signature:** Firma de los datos enviados. Es el resultado de la firma calculada siguiendo el algoritmo indicado en el parámetro Ds_SignatureVersion.

NOTA: recuerde que para realizar la petición REST debe enviar la cabecera **Content-Type** con el siguiente valor **"application/json"**.

NOTA: la conexión requiere del uso de un sistema de firma basado en HMAC SHA512, 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) . Ver Anexos. Estas librerías también están disponibles en la siguiente dirección:

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

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

En la petición se debe identificar la versión de algoritmo que se está utilizando para la firma. Actualmente los comercios deben utilizar el valor **HMAC_SHA512_V1** para identificar la versión de todas las peticiones, por lo que este será el valor del parámetro **Ds_SignatureVersion**.

NOTA: Teniendo en cuenta que recomendamos la versión de firma con HMAC SHA512, también está disponible la versión de firma HMAC SHA256, si se desea utilizar esta firma Ver Anexo 2.

Montar la cadena de datos de la petición (Ds_MerchantParameters)

En la petición se deben enviar todos los parámetros necesarios para identificar el tipo de operación que se desea realizar. Todos los parámetros se montan en formato JSON, y el nombre de cada parámetro deberá indicarse en mayúsculas o con estructura "CamelCase" (Por ejemplo: DS_MERCHANT_AMOUNT o Ds_Merchant_Amount).

A continuación, se muestra un ejemplo del objeto JSON con los campos de una petición de devolución (DS_MERCHANT_TRANSACTIONTYPE = 3) antes de codificarlo en Base 64. Los ejemplos que mostraremos en este documento estarán representados en este formato para simplificar su comprensión:

```
{ "DS_MERCHANT_MERCHANTCODE": "999008881", "DS_MERCHANT_TERMINAL": "1", "DS_MERCHANT_ORDER": "06080232580", "DS_MERCHANT_AMOUNT": "100", "DS_MERCHANT_CURRENCY": "978", "DS_MERCHANT_TRANSACTIONTYPE": "3" }
```

Una vez montada la cadena JSON con todos los campos, es necesario codificarla en Base 64, sin retornos de carro, para asegurarnos de que se mantiene constante y no es alterada en el proceso de envío.

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

```
eyJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9URVJNSU5BTCi6IjEiLCJEU19NRVJDSEFOVF9PUkRFUiI6IjA2MDgwMjMyNTgwIiwiaWRFNfTUVSQ0hBTlRfQU1PVU5UIjoimTAwIiwiaWRFNfTUVSQ0hBTlRfQ1VSUkVOQ1kiOiI5NzgiLCJEU19NRVJDSEFOVF9UUkFOU0FDVlPTlRZUEUiOiIzIn0=
```

Esta cadena resultante de la codificación en BASE64 será el valor del parámetro **Ds_MerchantParameters**.

NOTA1: Para utilizar las librerías de ayuda ver Anexos

NOTA2: El listado completo de parámetros de entrada del Tpv Virtual (SIS) está disponible en el documento "TPV-Virtual Parámetros Entrada-Salida.xlsx".

Firmar los datos de la petición (Ds_Signature)

Para calcular la firma es necesario utilizar una clave específica para cada terminal. Para obtener esta clave tiene que acceder al Portal de Administración, opción Consulta datos de Comercio, en el apartado "Ver clave" puede consultar esta 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.** Además, esta misma clave de comercio sirve tanto para el algoritmo HMAC SHA512 como para el HMAC SHA256.

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

Para los ejemplos usaremos el siguiente string como Ds_MerchantParameters:

```
{"DS_MERCHANT_AMOUNT":"999","DS_MERCHANT_ORDER":"1234567890","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_URLLOK":"http://www.prueba.com/urlOK.php","DS_MERCHANT_URLKO":"http://www.prueba.com/urlKO.php"}
```

Por lo que el número de pedido, que es importante en este proceso, será **1234567890**

Compatible con los algoritmos HMAC SHA-512:

HMAC SHA-512:

1. Se genera una clave específica por operación. Para obtener la clave derivada a utilizar en una operación, se debe realizar un cifrado AES CBC con vector de inicialización todo a ceros entre la clave del comercio (la cual debe ser previamente procesada para que la longitud de esta sea de 16 caracteres, rellenando con 0 a la derecha si esta longitud es inferior) y el valor del número de pedido de la operación (Ds_Merchant_Order), además, este resultado debe ser devuelto codificado en BASE 64.

Clave de la operación procesada para ser de 16 caracteres:

sq7HjrUOBfKmC576

Clave tras ser cifrada con AES y diversificada con el número de pedido, que es **1234567890** en nuestro ejemplo:

RWt3/IPTzYRMXsQtKiGRKg==

2. Se calcula el HMAC SHA512 del valor del parámetro **Ds_MerchantParameters** y la clave obtenida en el paso anterior.
3. El resultado obtenido se codifica en BASE 64 URL Encoded (manteniendo el padding de =) y ese será el valor del parámetro **Ds_Signature**, tal y como se puede observar en el ejemplo de formulario mostrado al inicio del apartado 3.

Para nuestro ejemplo, el resultado de HMAC SHA 512 es el siguiente:

sNshBIGLKfv04FBXKt_IMaueFt_yA7VZ1Mw4USg4HiLehAdiQ8xUt5pEM-oHvXCBNZJKZkk7ogzPjhxDW3hAEQ==

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

4. Estructura de respuesta REST

Una vez gestionada la petición, el TPV Virtual responderá al servidor del comercio con la información del resultado incluida en una cadena JSON.

En función de si la petición se ha procesado correctamente o no, se recibirán dos tipos de respuesta:

1) Respuesta de una operación procesada correctamente

Cuando una petición se ha procesado correctamente, una vez recibida la respuesta de la petición al TPV Virtual, el comercio debe capturar y validar los parámetros de retorno para conocer el resultado de la operación.

Cualquier respuesta del TPV-Virtual será un JSON que incluirá los siguientes parámetros:

- **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.
- **Ds_Signature:** Firma de los datos recibidos. Es el resultado del HMAC SHA 512 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 cálculo de la firma en la respuesta se realiza del mismo modo que en la petición.**

NOTA: Puede utilizar las librerías de ayuda para validar la firma, ver Anexos. Punto 2

2) Respuesta de una operación No procesada correctamente

Cuando una petición no se ha procesado correctamente, se informará en un JSON el código de error que identificará el motivo por el cual la petición no se ha podido procesar.

El error que se ha producido se informará en el parámetro *errorCode*, Ej.: {"errorCode":"SIS0042"}

NOTA: Ver listado de códigos de error en la guía "TPV-Virtual Parámetros Entrada-Salida.xlsx", apdo. códigos de error

3) Verificación del resultado de la operación

El resultado de la operación se informará mediante el parámetro Ds_Response o "Código de respuesta". Los valores de este campo para indicar si una operación ha sido autorizada se indican en la siguiente tabla.

Valor de Ds_Response	Descripción
Valor de 0 a 100	Operación autorizada en pagos y preautorizaciones.
Valor 900	Operación autorizada en devoluciones, confirmaciones y autenticación PUCE
Valor 400	Operación autorizada en anulaciones
Cualquier otro valor	Revisar listado detallado en "TPV-Virtual Parámetros Entrada-Salida.xlsx"

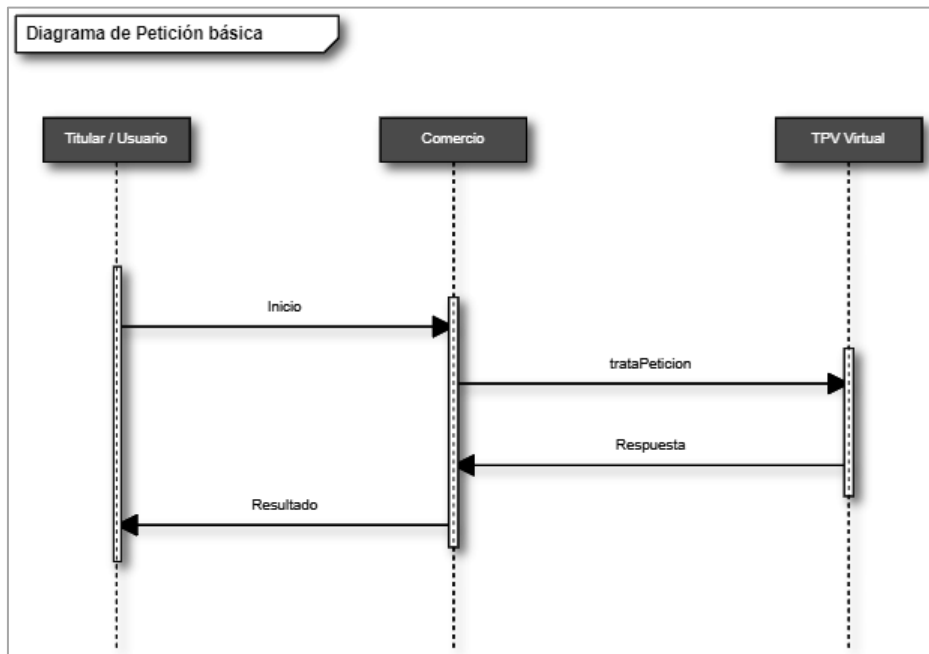
Un ejemplo de respuesta de una operación de pago autorizada sería la siguiente:

```
{
  "Ds_Amount":"1000",
  "Ds_Currency":"978",
  "Ds_Order":"1552568529",
  "Ds_MerchantCode":"999008881",
  "Ds_Terminal":"2",
  "Ds_Response":"0000",
  "Ds_AuthorisationCode":"842841",
  "Ds_TransactionType":"0",
  "Ds_SecurePayment":"0",
  "Ds_Language":"1",
  "Ds_Card_Type":"C",
  "Ds_Card_Country":"724",
  "Ds_Card_Brand":"1"
}
```

NOTA2: El listado completo de parámetros de entrada del Tpv Virtual (SIS) está disponible en el documento "TPV-Virtual Parámetros Entrada-Salida.xlsx".

5. Transacciones directas (sin autenticación)

El siguiente esquema muestra el flujo general de una operación realizada a través de la entrada REST del TPV Virtual sin autenticación del titular.



Para realizar una petición en la que no es necesario autenticar al titular (ej: pago sin autenticación, devolución, confirmación de preautorización, etc.), el comercio deberá preparar la petición con los parámetros necesarios según se ha indicado en el punto 3 y enviarla a los siguientes endpoints, dependiendo de si se quiere realizar una petición en el entorno de prueba u operaciones reales:

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

NOTA: El listado completo de parámetros de entrada del Tpv Virtual (SIS) está disponible en el documento "TPV-Virtual Parámetros Entrada-Salida.xlsx".

6. Transacciones con autenticación EMV 3DS

Pasos para realizar una transacción con autenticación

El interfaz de integración REST permite realizar pagos con autenticación del titular utilizando el protocolo EMV 3D Secure definido por las marcas..

Para realizar la autenticación es necesario incorporar en el flujo de pago varios pasos adicionales, que se explican a continuación.

El proceso de pago de una operación con autenticación en la conexión REST sigue los siguientes pasos:

- **Paso 1: Iniciar petición**

El comercio deberá hacer una petición al TPV Virtual para obtener información sobre la capacidad de la tarjeta en cuanto a autenticación (versión protocolo EMV 3DS), posibilidad de aplicación de exenciones, poder realizar operativas especiales (por ej. si permite DCC), información que indicará como deben gestionarse los siguientes pasos para realizar la transacción.

- **Paso 2: 3DSMethod (sólo en protocolo EMV3DS)**

Si la tarjeta tiene asociada una URL de 3DSMethod, el comercio debe adaptar la página en el navegador del titular para que se establezca una conexión directa con el emisor que le permita capturar la información del dispositivo utilizado por el titular: User-Agent, modelo de dispositivo, etc. Más información sobre este paso ver punto 6.5.2.

- **Paso 3: Solicitud de autorización**

El comercio, en la solicitud de autorización de la operación enviará el resultado del 3DSMethod, los datos adicionales del protocolo EMV3DS (versión de protocolo), así como una posible solicitud de exención SCA dentro del marco de la PSD2.

El TPV Virtual iniciará el proceso de autenticación. En el caso de una autenticación en versión EMV3DS se podrá obtener como resultado:

- a. Autenticación OK (Frictionless): la operación ha sido autenticada sin necesidad de solicitar ninguna acción al titular de la tarjeta y el TPV Virtual continuará el proceso de autorización.
- b. Challenge Required: La entidad emisora requiere verificar la autenticidad del cliente mediante una autenticación explícita o reto (challenge).
- c. Otro resultado: autenticación no disponible, autenticación rechazada, error en la autenticación, etc.

En los casos a (frictionless) y c (otro resultado), la operación finaliza en ese punto y el TPV Virtual responde con el resultado de la misma ya sea autorizada o denegada. Si la respuesta del emisor en este paso es la solicitud de challenge, el comercio debe continuar el flujo con los siguientes pasos para completar la operación.

- **Paso 4: Autenticación (en caso de que sea requerido challenge)**

En este paso, el navegador del titular es redirigido para conectarse con la Entidad emisora que verificará su autenticidad mediante una autenticación, con participación del titular de la tarjeta (challenge), esta autenticación se realizará mediante el sistema que solicite la Entidad Emisora: OTP por SMS (One Time Password), contraseña estática, biometría, combinación de los anteriores, etc.

- **Paso 5: Autorización (validación del resultado del challenge)**

El comercio enviará de nuevo la petición, con el resultado del challenge, al TPV Virtual para finalizar el proceso de autorización.

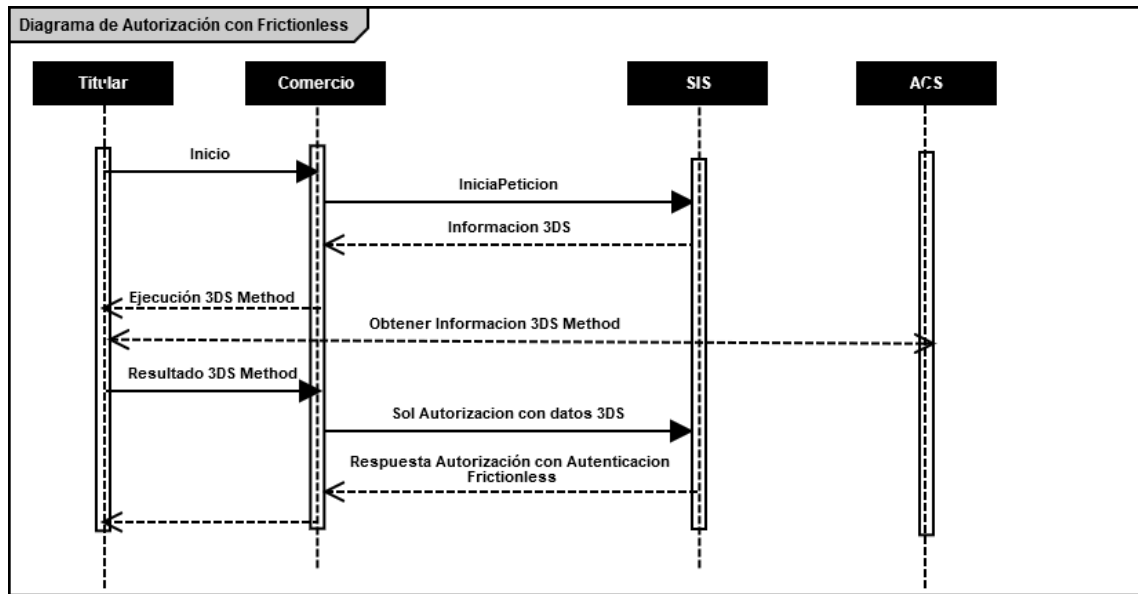
NOTA IMPORTANTE: *el comercio deberá estar preparado para soportar ambos procesos de Autenticación: Challenge (con intervención del titular) o Frictionless (sin intervención del titular). El emisor de la tarjeta será el encargado de determinar el proceso de Autenticación que se deberá llevar a cabo.*

NOTA: Recomendamos que en el paso 3 (solicitud de autorización), el comercio proporcione toda la información adicional posible para ayudar al emisor a identificar que la operación se está realizando por el auténtico titular de la tarjeta. Esta información adicional aumentará la probabilidad de un flujo frictionless (autenticación sin intervención del titular), ayudando así a reducir la tasa de abandono.

El tiempo máximo desde el inicio de la petición y la solicitud de autorización es de 1 hora. Pasado este tiempo la petición se da como perdida y se deberá volver a realizar el flujo desde el principio.

Flujo autorización con autenticación EMV3DS frictionless

El siguiente esquema presenta el flujo general de una operación con autenticación frictionless realizada a través del TPV Virtual.



1. El titular selecciona los productos que desea comprar e introduce los datos de tarjeta en un formulario mostrado por el comercio.
2. El comercio realiza un inicia petición enviando los datos al TPV Virtual.
3. El TPV Virtual comprueba la configuración de la tarjeta, y en la respuesta informará si la tarjeta soporta autenticación EMV3DS y la versión de protocolo EMV3DS que se aplica.

3.1 Si la tarjeta lo requiere, ejecutar el 3DSMethod: se inicia conexión desde el browser con el ACS y este devuelve el resultado de la ejecución al comercio.

4. El comercio envía la solicitud de autorización con tarjeta que soporta EMV3DS. Además de los datos de pago, es necesario enviar el resultado del 3DSMethod y los datos adicionales para la autenticación.

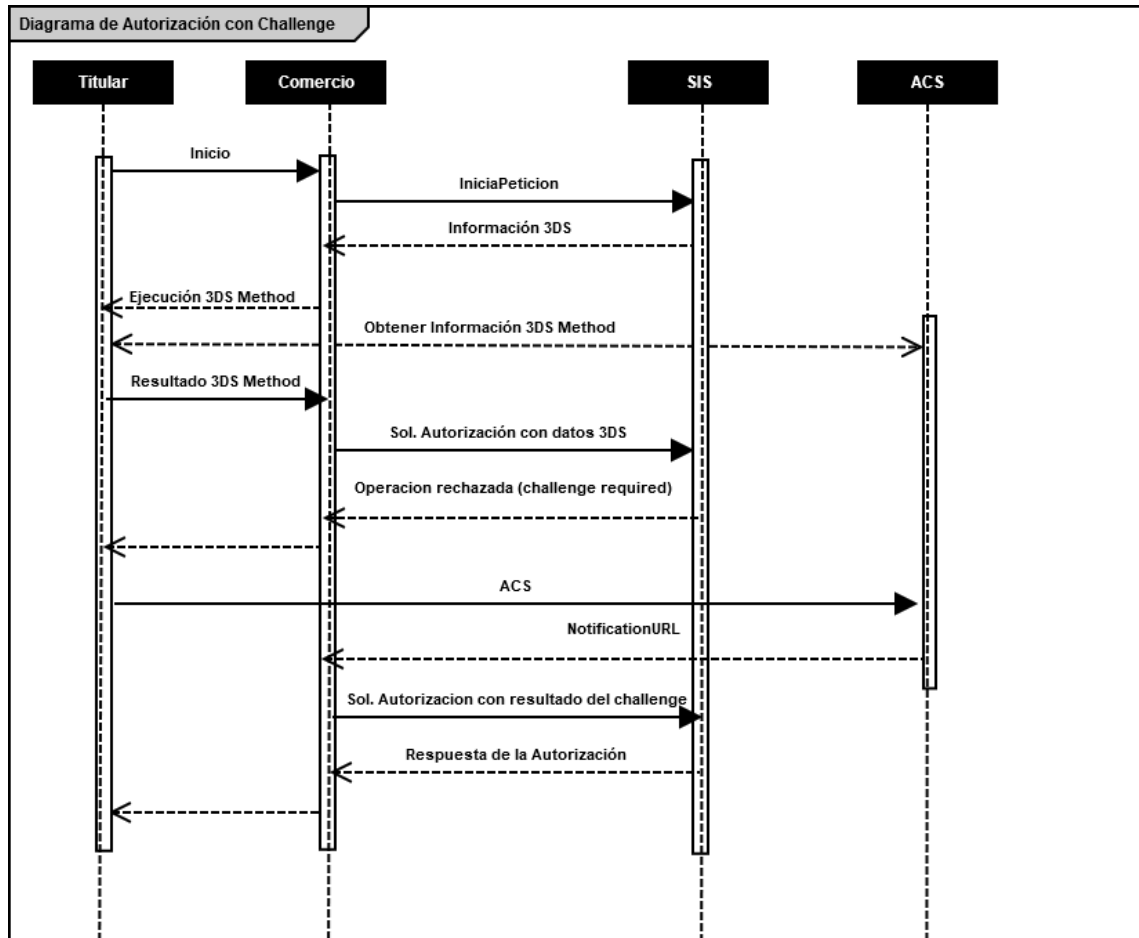
El TPV Virtual inicia la autenticación, y el emisor, en base a los datos recibidos, autentica la operación sin necesidad de intervención del titular. A continuación, el TPV Virtual procesará la autorización

5. Una vez realizado el pago, el TPV virtual informa del resultado de la operación.
6. El comercio muestra el resultado del pago al titular.

NOTA: en los diagramas solo se tienen en cuenta los flujos del comercio y los actores con lo que interviene en contacto directo.

Flujo autorización con autenticación EMV3DS challenge

El siguiente esquema presenta el flujo general de una operación con autenticación por challenge realizada a través del TPV Virtual.



1. El titular selecciona los productos que desea comprar e introduce los datos de tarjeta en un formulario mostrado por el comercio.
2. El comercio realiza un inicio petición enviando los datos al TPV Virtual.
3. El TPV Virtual comprueba la configuración de la tarjeta, y en la respuesta informará si la tarjeta soporta autenticación EMV3DS y la versión de protocolo EMV3DS que se aplica.

3.1 Si la tarjeta lo requiere, ejecutar el 3DSMethod: se inicia conexión desde el browser con el ACS y este devuelve el resultado de la ejecución al comercio

4. El comercio envía la solicitud de autorización con tarjeta que soporta EMV3DS. Además de los datos de pago es necesario enviar el resultado del 3DSMethod y los datos adicionales para la autenticación.

El TPV Virtual inicia la autenticación, y el emisor en base a los datos recibidos decide que el titular debe verificar su autenticidad (challenge)

5. El TPV Virtual devuelve la información para que el titular pueda realizar el challenge (autenticación) con su banco emisor. En este momento, y hasta que se reciba la respuesta

de la autenticación en el paso 9, las peticiones quedarán marcadas en el Portal de administración como "Sin Finalizar" con el código = 8210.

6. El comercio redirige al titular vía browser para que conecte con su emisor
7. El titular completa el challenge (autenticación)
8. La entidad emisora devuelve el resultado del challenge (autenticación) a la url indicada por el comercio
9. El comercio envía el resultado del challenge (autenticación) al TPV Virtual para finalizar el proceso de autorización
10. Una vez realizado el pago, el TPV virtual responde con el resultado de la operación.
11. El comercio responde con la información del resultado del pago al titular

NOTA: en los diagramas solo se tienen en cuenta los flujos del comercio y los actores con lo que interviene en contacto directo.

Ejemplo de peticiones para realizar una transacción con autenticación EMV3DS

6.5.1 Iniciar Petición

Esta petición permite obtener información sobre el tipo de autenticación 3D Secure que se puede realizar con la tarjeta, además de la URL del 3DSMethod, en caso de que exista.

El inicia petición se hace a través de una petición REST al TPV Virtual siguiendo la estructura de petición indicada en el punto 3 de este documento. Los endpoints a utilizar dependiendo de si se quiere realizar una petición en el entorno de prueba u operaciones reales son los siguientes:

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

A continuación, se describen los datos que debe incluir el Ds_MerchantParameters para enviar un inicia petición al Servicio REST:

```
{
  "DS_MERCHANT_ORDER": "1552571678",
  "DS_MERCHANT_MERCHANTCODE": "999008881",
  "DS_MERCHANT_TERMINAL": "999",
  "DS_MERCHANT_CURRENCY": "978",
  "DS_MERCHANT_TRANSACTIONTYPE": "0",
  "DS_MERCHANT_AMOUNT": "1000",
  "DS_MERCHANT_PAN": "XXXXXXXXXXXXXXXXXX ",
  "DS_MERCHANT_EMV3DS": {"threeDSInfo": "CardData"}
```

```
}

```

Como respuesta se obtendrá lo siguiente:

```
{
  "Ds_Order": "1552571678",
  "Ds_MerchantCode": "999008881",
  "Ds_Terminal": "2",
  "Ds_TransactionType": "0",
  "Ds_EMV3DS": {
    "protocolVersion": "2.1.0",
    "threeDSServerTransID": "8de84430-3336-4ff4-b18d-f073b546ccea",
    "threeDSInfo": "CardConfiguration",
    "threeDSMethodURL": "https://sis.redsys.es/sis-simulador-web/threeDsMethod.jsp"
  },
  "Ds_Card_PSD2": "Y"
}
```

El parámetro **Ds EMV3DS** contiene información sobre las opciones de autenticación de la tarjeta. Estará compuesto por los siguientes campos:

- **protocolVersion**: siempre indicará el número de versión mayor permitido en la operación. El comercio será responsable de utilizar el número de versión para el cual esté preparado.
- **threeDSServerTransID**: identificador de la transacción EMV3DS.
- **threeDSInfo**: CardConfiguration.
- **threeDSMethodURL**: URL del 3DSMethod en caso de que el emisor de la tarjeta lo tenga definido.

El parámetro **Ds Card PSD2** informará al comercio si la tarjeta informada en la petición está afectada o no por PSD2. Los valores posibles serán "Y" para indicar que la tarjeta está afectada por PSD2, o "N" para indicar lo contrario.

6.5.2 Ejecución del 3DSMethod

El 3DSMethod es un proceso que permite a la entidad emisora capturar la información del dispositivo que está utilizando el titular. Esta información, junto con los datos EMV3DS, que son enviados en la autorización, será utilizada por la entidad para hacer una evaluación del riesgo de la transacción. En base a esto, el emisor puede determinar que la transacción es confiable y por lo tanto no requerir la intervención del titular para verificar su autenticidad (frictionless).

La captura de datos del dispositivo se realiza mediante un **iframe oculto** en el navegador del cliente, que establecerá conexión directamente con la entidad emisora de forma transparente para el usuario. El comercio recibirá una notificación cuando haya terminado la captura de información y en el siguiente paso, al realizar la petición de autorización al TPV Virtual, el comercio deberá enviar el parámetro **threeDSCompInd** indicando la ejecución del 3DSMethod.

Pasos para la ejecución del 3DSMethod:

1. En la respuesta recibida con la configuración de la tarjeta (iniciaPetición) se recibe los datos siguientes para ejecutar el 3DSMethod:

- a. `threeDSMethodURL`: url del 3DSMethod
- b. `threeDSSTransID`: Identificador de transacción EMV3DS

Si en la respuesta no se recibe `threeDSMethodURL`, este paso finaliza y el comercio deberá enviar `threeDSCompInd = N` en la petición de autorización.

2. Construir el JSON Object con los siguientes parámetros:

- a. `threeDSSTransID`: valor recibido en la respuesta de consulta de tarjeta
- b. `threeDSMethodNotificationURL`: url del comercio a la que será notificada la finalización del 3DSMethod desde la entidad emisora.

3. Codificar el JSON anterior en Base64url encode

4. Debe incluirse un iframe oculto en el navegador del cliente, y enviar un campo **`threeDSMethodData`** con el valor del objeto json anterior en un formulario http post a la url obtenida en la consulta inicial **`threeDSMethodURL`**

5. La entidad emisora interactúa con el browser para proceder a la captura de información. Al finalizar enviará el campo **`threeDSMethodData`** en el iframe html del navegador por http post a la url **`threeDSMethodNotificationURL`** (indicada en el paso 2), y el 3DSMethod termina.

6. Si el 3DSMethod se ha completado en menos de 10 segundos se enviará **`threeDSCompInd = Y`** en la autorización. Si no se ha completado en 10 segundos debe detener la espera y enviar la autorización con **`threeDSCompInd = N`**

6.5.3 Petición de autorización con datos EMV3DS

La petición de autorización se hace a través de otra petición REST al TPV Virtual con la estructura indicada en el punto 3 de este documento.

La petición de autorización se debe enviar a los siguientes endpoints dependiendo de si se quiere realizar una petición en el entorno de prueba u operaciones reales:

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

A continuación, se describen los datos de debe incluir el `Ds_MerchantParameters` para enviar una petición de autorización con autenticación EMV3DS al Servicio REST:

```
{
  "DS_MERCHANT_ORDER": "1552572812",
  "DS_MERCHANT_MERCHANTCODE": "999008881",
  "DS_MERCHANT_TERMINAL": "2",
  "DS_MERCHANT_CURRENCY": "978",
}
```

```

"DS_MERCHANT_TRANSACTIONTYPE":"0",
"DS_MERCHANT_AMOUNT":"1000",
"DS_MERCHANT_PAN":"XXXXXXXXXXXXXXXXXX ",
"DS_MERCHANT_EXPIRYDATE":"XXXX",
"DS_MERCHANT_CVV2":"XXX",
"DS_MERCHANT_EMV3DS":
{
    "threeDSInfo":"AuthenticationData",
    "protocolVersion":"2.1.0",
    "browserAcceptHeader":"text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0
.8,application/json",
    "browserUserAgent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36",
    "browserJavaEnabled":"false",
    "browserJavaScriptEnabled":"true",
    "browserLanguage":"ES-es",
    "browserColorDepth":"24",
    "browserScreenHeight":"1250",
    "browserScreenWidth":"1320",
    "browserTZ":"52",
    "threeDSServerTransID":"8de84430-3336-4ff4-b18d-f073b546ccea",
    "notificationURL":"https://comercio-inventado.es/recibe-respuesta-autenticacion",
    "threeDSCompInd":"Y"
}
}

```

NOTA: El valor de la versión indicada en el campo `protocolVersion` no puede ser superior al devuelto por el TPV Virtual en la llamada previa a `iniciaPetición`.

Como respuesta se obtendrá:

- Si se hace un **Frictionless**, se obtendrá directamente el resultado final de la operación:

```

{
  "Ds_Amount":"1000",
  "Ds_Currency":"978",
  "Ds_Order":"1552572812",
  "Ds_MerchantCode":"999008881",
  "Ds_Terminal":"2",
  "Ds_Response":"0000",
  "Ds_AuthorisationCode":"694432",
  "Ds_TransactionType":"0",
  "Ds_SecurePayment":"2",
  "Ds_Language":"1",
  "Ds_CardNumber":"454881*****0004",
  "Ds_Card_Type":"C",
  "Ds_MerchantData":"",
  "Ds_Card_Country":"724",
  "Ds_Card_Brand":"1"
}

```

- Si se requiere **Challenge**, se obtendrán los datos para realizar el Challenge:

```

{
  "Ds_Amount":"1000",
  "Ds_Currency":"978",
  "Ds_Order":"1552572812",
  "Ds_MerchantCode":"999008881",
  "Ds_Terminal":"2",
  "Ds_TransactionType":"0",
  "Ds_EMV3DS":{

```

```

    "threeDSInfo":"ChallengeRequest",
    "protocolVersion":"2.1.0",
    "acsURL":"https://sis.redsys.es/sis-simulador-web/authenticationRequest.jsp",
    "creq":"eyJ0aHJIZURTU2VydmVYVHJhbnNJRCI6IjZlLTZlOTJiZjE5NDZlMmFjc1RyYW5zSUQlOiJkYjVjOTIjNC1hMmZkLTQ3ZWU0OTI2Zi1mYTBiMDk0MzUyYTAiLCJtZXNzYWdlVHlwZSI6IkNSZXEiLCJtZXNzYWdlVmVyc2l2bWVudlV2luZG93U2l6ZSI6IjA1In0"
  }
}

```

6.5.4 Ejecución del Challenge

Describimos este proceso en 3 pasos:

Paso 1. - Conexión desde el comercio el ACS del banco emisor

El siguiente paso consiste en conectar desde el comercio con la entidad emisora para que el cliente se pueda autenticar. Esta conexión se hace enviando un formulario http POST a la url del ACS del banco. Para esta conexión utilizamos los datos recibidos en el parámetro Ds_EMV3DS del paso anterior (parámetros acsURL y creq):

```

<Ds_EMV3DS>{"threeDSInfo":"ChallengeRequest",
  "protocolVersion":"2.1.0",
  "acsURL":"https://sis.redsys.es/sis-simulador-web/authenticationRequest.jsp",
  "creq":"eyJ0aHJIZURTU2VydmVYVHJhbnNJRCI6ImU5OWMzYzI2LTZlOTJiZjE5NDZlMmFjc1RyYW5zSUQlOiJkYjVjOTIjNC1hMmZkLTQ3ZWU0OTI2Zi1mYTBiMDk0MzUyYTAiLCJtZXNzYWdlVHlwZSI6IkNSZXEiLCJtZXNzYWdlVmVyc2l2bWVudlV2luZG93U2l6ZSI6IjA1In0"}
</Ds_EMV3DS>

```

Ejemplo:

```

<form action="{acsURL}" method="POST" enctype = "application/x-www-form-urlencoded">
  <input type="hidden" name="creq" value="{creq}" ">
</form>

```

Con los datos recibidos en <Ds_EMV3DS> sería:

```

<form action="https://sis.redsys.es/sis-simulador-web/authenticationRequest.jsp" method="POST" enctype =
"application/x-www-form-urlencoded">

<input type="hidden" name="creq"
value="eyJ0aHJIZURTU2VydmVYVHJhbnNJRCI6ImU5OWMzYzI2LTZlOTJiZjE5NDZlMmFjc1Ry
YW5zSUQlOiJkYjVjOTIjNC1hMmZkLTQ3ZWU0OTI2Zi1mYTBiMDk0MzUyYTAiLCJtZXNzYWdlVHlwZSI6IkNSZXEiLCJtZXNzYWdlVmVyc2l2bWVudlV2luZG93U2l6ZSI6IjA1In0">
</form>

```

NOTA: es importante respetar las mayúsculas y minúsculas en el nombre de los parámetros.

Paso 2. - Ejecución del challenge

El titular se autentica por los métodos que le exija su entidad emisora: OTP, contraseña estática, biometría, etc.

Paso 3. - Recepción del resultado de la autenticación

Una vez finalizado el challenge, la entidad emisora enviará el resultado al comercio, haciendo un http POST a la url del parámetro *notificationURL* que el comercio envió previamente en la petición de autorización:

"notificationURL": "https://comercio-inventado.es/recibe-respuesta-autenticacion"

El comercio recibirá el parámetro "cres" que utilizará en la petición de autorización final que vemos en el siguiente apartado.

6.5.5 Confirmación de autorización EMV3DS posterior al Challenge

A continuación se describen los datos que debe incluir el Ds_MerchantParameters para enviar una petición de confirmación de autorización EMV3DS al Servicio REST:

```
{
  "DS_MERCHANT_ORDER": "1552577128",
  "DS_MERCHANT_MERCHANTCODE": "999008881",
  "DS_MERCHANT_TERMINAL": "2",
  "DS_MERCHANT_CURRENCY": "978",
  "DS_MERCHANT_TRANSACTIONTYPE": "0",
  "DS_MERCHANT_AMOUNT": "1000",
  "DS_MERCHANT_PAN": "XXXXXXXXXXXXXXXXXX",
  "DS_MERCHANT_EXPIRYDATE": "XXXX",
  "DS_MERCHANT_CVV2": "XXX",
  "DS_MERCHANT_EMV3DS": {
    "threeDSInfo": "ChallengeResponse",
    "protocolVersion": "2.1.0",
    "cres": "eyJ0aHJIZURTU2VydmVyVHJhbnNJRCI6IjkhkZTg0NDMwLTMzMzYtNGZmNC1iMThkLWYwNzNiNTQ2Y2NIYSIsImFjc1RyYW5zSUQiOiJkYjVjOTljNC1hMmZkLTQ3ZWUtOTI2Zi1mYTBiMDk0MzUyYTAiLCJtZXNzYWdlVHlwZSI6IkNSZXMiLCJtZXNzYWdlVmVyc2lvbil6IjluMS4wliwidHJhbnNTdGF0dXMiOiJZIn0="
  }
}
```

NOTA: el contenido del parámetro *cres* debe enviarse como una cadena continua sin retornos de carro ni saltos de línea.

Como respuesta se obtendrá el resultado final de la operación:

```
{
  "Ds_Amount": "1000",
  "Ds_Currency": "978",
  "Ds_Order": "1552572812",
  "Ds_MerchantCode": "999008881",
  "Ds_Terminal": "2",
  "Ds_Response": "0000",
  "Ds_AuthorisationCode": "694432",
  "Ds_TransactionType": "0",
  "Ds_SecurePayment": "2",
  "Ds_Language": "1",
  "Ds_CardNumber": "454881*****0004",
  "Ds_Card_Type": "C",
  "Ds_MerchantData": "",
  "Ds_Card_Country": "724",
  "Ds_Card_Brand": "1"
}
```

7. Transacciones con DCC

A continuación, se detallan las características adicionales de la operativa DCC en los comercios que utilicen la interfaz REST. El comercio tiene que estar configurado para realizar este tipo de operativa.

Un pago con DCC en la conexión Rest sigue los siguientes pasos:

- **Paso 1: Iniciar petición**

En el "inicia petición" el comercio hace una consulta al TPV Virtual para conocer si la tarjeta permite operativa DCC. Para realizar esta petición DCC, el Ds_MerchantParameters tiene que incluir el parámetro **DS_MERCHANT_DCC**.

Ejemplo de Ds_MerchantParameters en el inicia petición para una operación con DCC.

```
{
  "DS_MERCHANT_ORDER":1552580496,
  "DS_MERCHANT_MERCHANTCODE":"999008881",
  "DS_MERCHANT_TERMINAL":"2",
  "DS_MERCHANT_TRANSACTIONTYPE":"0",
  "DS_MERCHANT_PAN":"XXXXXXXXXXXXXXXXXXXX",
  "DS_MERCHANT_CURRENCY":"978",
  "DS_MERCHANT_AMOUNT":"5785",
  "DS_MERCHANT_DCC":"Y"
}
```

Como respuesta, si la tarjeta permite realizar DCC, se obtendrá lo siguiente:

```
{
  "Ds_Order":"1552580496",
  "Ds_MerchantCode":"999008881",
  "Ds_Terminal":"2",
  "Ds_TransactionType":"0",
  "Ds_DCC":{
    "InfoMonedaTarjeta":{
      "monedaDCC":"826",
      "litMonedaDCC":"POUND STERLING.",
      "litMonedaRDCC":"GBP",
      "importeDCC":"53.60",
      "cambioDCC":"1.079288",
      "fechaCambioDCC":"2019-01-16",
      "markUp":"0.03",
      "cambioBCE":"1,136092",
      "porcentajeSobreBCE":"0,05"
    },
    "InfoMonedaComercio":{
      "monedaCome":"978",
      "litMonedaCome":"EUR",
      "importeCome":"57.85"
    }
  },
  "Ds_Card_PSD2":"Y"
}
```

Si la tarjeta no permite realizar DCC la respuesta será la siguiente:

```
{
  "Ds_Order":"1552580496",
  "Ds_MerchantCode":"999008881",
  "Ds_Terminal":"2",
  "Ds_TransactionType":"0",
  "Ds_DCC":{
    "dataDCC":"No se aplica DCC"
  }
  "Ds_Card_PSD2":"Y"
}
```

- **Paso 2: Solicitud de autorización**

El comercio enviará la solicitud de autorización de la operación indicando la información de la moneda e importe DCC obtenidos en el paso anterior.

Ejemplo solicitud autorización con DCC

```
{
  "DS_MERCHANT_ORDER":1552581014,
  "DS_MERCHANT_MERCHANTCODE":"999008881",
  "DS_MERCHANT_TERMINAL":"2",
  "DS_MERCHANT_CURRENCY":"978",
  "DS_MERCHANT_TRANSACTIONTYPE":"0",
  "DS_MERCHANT_AMOUNT":"1000",
  "DS_MERCHANT_PAN":"XXXXXXXXXXXXXXXXXXXX",
  "DS_MERCHANT_EXPIRYDATE":"XXXX",
  "DS_MERCHANT_CVV2":"XXX",
  "DS_MERCHANT_DCC":{
    "monedaDCC":"840",
    "importeDCC":"11.50"
  }
}
```

Como respuesta se obtendrá el resultado final de la operación:

```
{
  "Ds_Amount":"1000",
  "Ds_Currency":"978",
  "Ds_Order":"1552572812",
  "Ds_MerchantCode":"999008881",
  "Ds_Terminal":"2",
  "Ds_Response":"0000",
  "Ds_AuthorisationCode":"694432",
  "Ds_TransactionType":"0",
  "Ds_SecurePayment":"1",
  "Ds_Language":"1",
  "Ds_CardNumber":"454881*****0004",
  "Ds_Card_Type":"C",
  "Ds_MerchantData":"",
  "Ds_Card_Country":"724",
  "Ds_Card_Brand":"1"
}
```

NOTA: El tiempo máximo desde el inicio de la petición y la solicitud de autorización es de 1 hora. Pasado este tiempo la petición se da como perdida y se deberá volver a realizar el flujo desde el principio.

NOTA: *En esta operativa DCC hay que tener en cuenta que, por normativa del Banco Central Europeo (BCE), en las operaciones en las que intervengan las siguientes monedas: Lev, Kuna croata, Corona danesa, Florín húngaro (forinto), Zloty, Corona Checa, Leu rumano, Corona sueca, Libra, es necesario informar al cliente del % de incremento entre el cambio aplicado y el cambio del BCE. Esta información se devolverá, en las operaciones en estas monedas, en los parámetros cambioBCE y porcentajeSobreBCE, y un posible texto de ejemplo sería el siguiente:*

"The exchange rate has a difference of x,xx % over published by ECB".

8. Transacciones DCC con autenticación

A continuación, se detallan las características adicionales para una transacción con autenticación en la que se desee utilizar la operativa DCC para comercios que utilicen la interfaz REST. El comercio tiene que estar configurado para hacer este tipo de operativa.

Partiendo de los pasos necesarios para la realización de una transacción con autenticación, incluiremos la parte específica de una operativa con DCC:

- **Paso 1: Iniciar petición**

El comercio deberá hacer una consulta al TPV Virtual para saber si la tarjeta está inscrita en EMV3DS y poder iniciar el proceso de autenticación. En esta petición el comercio puede consultar también si la tarjeta ofrece **DCC**.

A continuación, se describen los datos que debe incluir el `Ds_MerchantParameters` para enviar una petición de inicia petición al Servicio REST con posibilidad de autenticación y DCC:

```
{
  "DS_MERCHANT_ORDER":1552580496,
  "DS_MERCHANT_MERCHANTCODE":"999008881",
  "DS_MERCHANT_TERMINAL":"2",
  "DS_MERCHANT_TRANSACTIONTYPE":"0",
  "DS_MERCHANT_PAN":"XXXXXXXXXXXXXXXXXXXX",
  "DS_MERCHANT_CURRENCY":"978",
  "DS_MERCHANT_AMOUNT":"1000",
  "DS_MERCHANT_DCC":"Y",
  "DS_MERCHANT_EMV3DS":{"threeDSInfo":{"CardData"}}
}
```

Si la tarjeta requiere autenticación y permite DCC, la respuesta obtenida será como la siguiente:

```
{
  "Ds_Order":"1552580496",
  "Ds_MerchantCode":"999008881",
  "Ds_Terminal":"2",
  "Ds_TransactionType":"0",
  "Ds_DCC":{
    "InfoMonedaTarjeta":{
      "monedaDCC":"840",
      "litMonedaDCC":"DOLAR U.S.A.",
      "litMonedaRDCC":"USD",
      "importeDCC":"11.50",
      "cambioDCC":"0.869841",
      "fechaCambioDCC":"2019-01-16",
      "markUp":"0.03",
      "cambioBCE":"0.869835",
      "porcentajeSobreBCE":"0.05"
    },
    "InfoMonedaComercio":{
      "monedaCome":"978",
      "litMonedaCome":"EUR",
      "importeCome":"10.00"
    }
  },
  "Ds_EMV3DS":{
    "protocolVersion":"2.1.0",

```



```

    "threeDSServerTransID":"8de84430-3336-4ff4-b18d-f073b546ccea",
    "threeDSInfo":"CardConfiguration",
    "threeDSMethodURL":"https://sis.redsys.es/sis-simulador-web/threeDsMethod.jsp"
  },
  "Ds_Card_PSD2":"N"
}

```

- **Paso 2: 3DSMethod (Si procede)**

El comercio ejecuta el 3DSMethod para que el emisor capture la información del dispositivo.

- **Paso 3: Solicitud de autorización**

El comercio enviará la solicitud de autorización de la operación incluyendo el resultado del 3DSMethod y otros datos adicionales del protocolo EMV3DS. Además, incluyendo la información de **DCC** obtenida en el paso 1.

A continuación, se describen los datos de debe incluir el Ds_MerchantParameters para enviar una petición de autorización con DCC al Servicio REST:

```

{
  "DS_MERCHANT_ORDER":1552581014,
  "DS_MERCHANT_MERCHANTCODE":"999008881",
  "DS_MERCHANT_TERMINAL":"2",
  "DS_MERCHANT_CURRENCY":"978",
  "DS_MERCHANT_TRANSACTIONTYPE":"0",
  "DS_MERCHANT_AMOUNT":"1000",
  "DS_MERCHANT_PAN":"XXXXXXXXXXXXXXXXXXXX",
  "DS_MERCHANT_EXPIRYDATE":"XX",
  "DS_MERCHANT_CVV2":"XX",
  "DS_MERCHANT_DCC":{
    "monedaDCC":"840",
    "importeDCC":"11.50"
  },
  "DS_MERCHANT_EMV3DS":
  {
    "threeDSInfo":"AuthenticationData",
    "protocolVersion":"2.1.0",
    "browserAcceptHeader":"text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8,application/json",
    "browserUserAgent":"Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36",
    "browserJavaEnabled":"false",
    "browserJavascriptEnabled":"true",
    "browserLanguage":"ES-es",
    "browserColorDepth":"24",
    "browserScreenHeight":"1250",
    "browserScreenWidth":"1320",
    "browserTZ":"52",
    "threeDSServerTransID":"8de84430-3336-4ff4-b18d-f073b546ccea",
    "notificationURL":"https://comercio-inventado.es/recibe-respuesta-autenticacion",
    "threeDSCompInd":"Y"
  }
}

```

Como resultado a esta petición se podrán tener una de los siguientes opciones:

- Autenticación OK (Frictionless): la operación ha sido autenticada y el TPV Virtual continuará el proceso de autorización.

- b. Challenge Required: La entidad emisora requiere verificar la autenticidad del cliente
- c. Otro resultado: autenticación no disponible, autenticación rechazada, error en la autenticación, etc.

El TPV Virtual decidirá según el caso autorizar o rechazar la operación.

- **Paso 4: Autenticación (En el caso de que la entidad emisora haya solicitado challenge)**

La entidad emisora requiere que el titular verifique su autenticidad (mediante OTP, contraseña estática, biometría, etc).

- **Paso 5: Confirmación de autorización**

El comercio enviará la autorización con el resultado del challenge al TPV Virtual para finalizar el proceso de autorización. Debe incluir también toda la información para el pago, así como los campos de DCC del mismo modo que en el paso 3.

NOTA: Recomendamos que en el paso 3 el comercio proporcione toda la información adicional para aumentar la probabilidad de flujo frictionless y una mayor tasa de autorización.

9. Adaptaciones PSD2

De acuerdo a la norma de 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, 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 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. Estas operaciones están fuera del alcance de la PSD2. Este es el caso de las operativas de pagos de suscripciones, recurrentes, etc. y, en general, casi todas las que requieren el almacenamiento de las credenciales de pago del cliente (COF) o su equivalente "pago por referencia". 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.
- COR (Secure Corporate Payment): exención restringida al caso de operaciones entre empresas, no a consumidores.
- 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)

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

Ejemplos de peticiones con exenciones.

Como se indica en el punto anterior, la normativa contempla diferentes exenciones que se pueden marcar en la petición de pago para proponer no realizar el proceso de autenticación del titular. Hay que tener en cuenta que al proponer una exención la responsabilidad ante un posible fraude de la operación recae en el comercio.

Se contemplan dos tipos de mensajes donde podemos marcar una exención:

- **Petición con datos EMV3DS.** Las exenciones solicitadas en peticiones con envío de datos EMV3DS, se marcarán en la autenticación. Si esta exención no es aceptada por el emisor, se devolverá una petición de CHALLENGE para que el titular se autentique con SCA. De esta forma la petición no se pierde y continuará el flujo habitual, sin que el titular se vea afectado. SE RECOMIENDA ESTA OPCIÓN.
- **Petición sin datos EMV3DS.** Las exenciones solicitadas en las peticiones en las que no se han informado los datos EMV3DS, se marcarán en la autorización. Si esta exención no es aceptada por el emisor, se procederá a una denegación con **Ds Response = 0195** ("soft-decline" requiere SCA). En este caso el comercio puede decidir iniciar de nuevo la operación con datos EMV3DS pero deberá enviar otra petición completamente nueva.

Mensaje Inicia Petición (para conocer las exenciones permitidas al comercio)

Las exenciones permitidas dependen de la configuración del comercio. La activación de estas exenciones se realiza por parte de la entidad adquirente.

Para conocer qué exenciones puede aplicar el comercio en cada una de las transacciones, tiene que enviar el parámetro DS_MERCHANT_EXCEP_SCA con el valor "Y" en la llamada a *inicia petición*.

NOTA: Para la exención TRA, se establece un máximo de importe que vendrá también informado en la respuesta.

Ejemplo de inicia petición:

```
{
  "DS_MERCHANT_ORDER":"1552571678",
  "DS_MERCHANT_MERCHANTCODE":"999008881",
  "DS_MERCHANT_TERMINAL":"999",
  "DS_MERCHANT_CURRENCY":"978",
  "DS_MERCHANT_TRANSACTIONTYPE":"0",
  "DS_MERCHANT_AMOUNT":"1000",
  "DS_MERCHANT_EXCEP_SCA":"Y",
  "DS_MERCHANT_PAN":"xxxxxxxxxxxxxxxxxx ",
  "DS_MERCHANT_EMV3DS":{"threeDSInfo":"CardData"}
}
```

Un ejemplo de respuesta sería la siguiente:

```
{
  "Ds_Order":"1552571678",
  "Ds_MerchantCode":"999008881",
```

```

"Ds_Terminal": "2",
"Ds_TransactionType": "0",
"Ds_EMV3DS": {
  "protocolVersion": "2.1.0",
  "threeDSSTransID": "8de84430-3336-4ff4-b18d-f073b546ccea",
  "threeDSInfo": "CardConfiguration",
  "threeDSMethodURL": "https://sis.redsys.es/sis-simulador-web/threeDsMethod.jsp",
},
"Ds_Except_SCA": "LWV;TRA[30.0];COR;MIT",
"Ds_Card_PSD2": "Y"
}

```

Mensaje Trata Petición (Con EMV3DS)

Se incluye el parámetro DS_MERCHANT_EXCEPT_SCA con el valor de la excepción propuesta. Al estar informados también los campos de EMV3DS la exención se solicitará en el momento de solicitar la autenticación.

```

{
  "DS_MERCHANT_ORDER": "1552572812",
  "DS_MERCHANT_MERCHANTCODE": "999008881",
  "DS_MERCHANT_TERMINAL": "2",
  "DS_MERCHANT_CURRENCY": "978",
  "DS_MERCHANT_TRANSACTIONTYPE": "0",
  "DS_MERCHANT_AMOUNT": "100",
  "DS_MERCHANT_PAN": "XXXXXXXXXXXXXXXXXX",
  "DS_MERCHANT_EXPIRYDATE": "XXXX",
  "DS_MERCHANT_CVV2": "XXX",
  "DS_MERCHANT_EXCEPT_SCA": "LWV",
  "DS_MERCHANT_EMV3DS": {
    "threeDSInfo": "AuthenticationData",
    "protocolVersion": "2.1.0",
    "browserAcceptHeader": "text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8",
    "browserUserAgent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/71.0.3578.98 Safari/537.36",
    "browserJavaEnabled": "false",
    "browserJavascriptEnabled": "true",
    "browserLanguage": "ES-es",
    "browserColorDepth": "24",
    "browserScreenHeight": "1250",
    "browserScreenWidth": "1320",
    "browserTZ": "52",
    "threeDSSTransID": "8de84430-3336-4ff4-b18d-f073b546ccea",
    "notificationURL": "https://comercio-inventado.es/recibe-respuesta-autenticacion",
    "threeDSComplnd": "Y"
  }
}

```

Mensaje Trata Petición (Sin EMV3DS)

Se incluye el parámetro DS_MERCHANT_EXCEPT_SCA con el valor de la excepción propuesta. Como no se informan los campos de EMV3DS, la exención se solicitará directamente en la petición de autorización por lo que si el emisor no la acepta podrá denegar con un 0195 "soft-decline" para indicar que requiere autenticación. Si el terminal tiene configurados métodos de pago seguros, se deberá añadir también el parámetro DS_MERCHANT_DIRECTPAYMENT con el valor true.

```
{
  "DS_MERCHANT_ORDER":1552572812,
  "DS_MERCHANT_MERCHANTCODE":"999008881",
  "DS_MERCHANT_TERMINAL":"2",
  "DS_MERCHANT_CURRENCY":"978",
  "DS_MERCHANT_TRANSACTIONTYPE":"0",
  "DS_MERCHANT_AMOUNT":"100",
  "DS_MERCHANT_PAN":"XXXXXXXXXXXXXXXXXX ",
  "DS_MERCHANT_EXPIRYDATE":"XXXX",
  "DS_MERCHANT_CVV2":"XXX",
  "DS_MERCHANT_DIRECTPAYMENT":"true",
  "DS_MERCHANT_EXCEP_SCA":"LWV"
}
```

Ejemplo transacción MIT (Merchant 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 suscripción. Este tipo de 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 inicial**) 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: Para más información sobre especificaciones de Credentials on File (COF) ver Guía Especificaciones COF Ecom.

NOTA2: El listado completo de todos los parámetros de entrada del SIS está disponible en el doc. "TPV-Virtual Parámetros Entrada-Salida.xlsx".

10. Funcionalidades avanzadas 3RI y 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.

11. Otras integraciones REST: PSPs/ MPI externo/ PUCE

Integración para PSP

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

Los parámetros a enviar en estas peticiones de pago son los mismos que en la petición desde un comercio, pero utilizando una versión de firma específica para PSP.

No se definen flujos específicos para el procesamiento de operaciones por parte de PSP, puesto que son los mismos que ya se han indicado en este documento.

NOTA: para realizar esta integración con el Tpv virtual de Redsys, se requiere la activación por parte de la Entidad Adquirente.

11.1.1 Configuración

Es necesario que un comercio-terminal para procesar peticiones a través de un PSP esté configurado asociado a este PSP. Esta configuración, el comercio tiene que solicitarla a su Entidad.

11.1.2 Solicitud y recepción de claves

Se recibirán dos claves privadas con el protocolo establecido por Redsys.

- Clave para realizar cifrado AES-128 CBC del campo DS_MERCHANT_PAN, DS_XPAYDECODEDDATA o DS_MERCHANT_TOKENDATA.
- Clave HMAC SHA512 para firmar la petición (DS_MERCHANTPARAMETERS).

11.1.3 Envío de petición al TPV Virtual

Al igual que en la petición de pago enviada por un comercio, el PSP deberá enviar una petición REST con un JSON formado por los siguientes tres campos:

- **Ds_SignatureVersion:** Constante que indica la versión de firma que se está utilizando. Para esta integración de PSP el valor a utilizar será **T25V2**.
- **Ds_MerchantParameters:** Cadena en formato JSON con todos los parámetros de la petición codificada en Base 64 URL Encoded 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).

Si se envía el campo DS_MERCHANT_PAN, DS_XPAYDECODEDDATA o DS_MERCHANT_TOKENDATA este debe estar cifrado en AES-128 CBC con la clave de cifrado proporcionada por Redsys.

- **Ds_Signature:** Firma de los datos enviados. Es el resultado del HMAC SHA512 del valor del parámetro Ds_MerchantParameters y la clave de firma proporcionada por

Redsys. El valor obtenido se debe codificar en BASE64 URL Encoded (manteniendo el padding de =).

Dichos parámetros deben enviarse a los siguientes endpoints dependiendo de si se quiere realizar una petición en el entorno de prueba u operaciones reales:

URL Conexión inicia petición	Entorno
https://sis-t.redsys.es:25443/sis/rest/iniciaPeticonPSPREST	Pruebas
https://sis.redsys.es/sis/rest/iniciaPeticonPSPREST	Real
URL Conexión trata petición	
https://sis-t.redsys.es:25443/sis/rest/trataPeticonPSPREST	Pruebas
https://sis.redsys.es/sis/rest/trataPeticonPSPREST	Real

11.1.4 Recepción del resultado

La recepción del resultado será firmada de la misma forma que la petición de envío con la clave de firma HMAC SHA512 proporcionada.

11.1.5 Ejemplo de peticiones

Cadena en JSON

```
{
  "DS_MERCHANT_AMOUNT": "145",
  "DS_MERCHANT_ORDER": "1446083946",
  "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": "4548*****7229",
  "DS_MERCHANT_EXPIRYDATE": "4912",
  "DS_MERCHANT_CVV2": "123"
}
```

Ciframos en AES-128 CBC el parámetro DS_MERCHANT_PAN

La clave proporcionada por Redsys no debe utilizarse directamente para el proceso de cifrado. Deberemos tratarla siguiendo los siguientes pasos:

- Convertir la clave proporcionada por Redsys de hexadecimal a binario (por ejemplo, con hex2bin()).
- Extraer los primeros 16 bytes del resultado binario.
- Utilizar esos 16 bytes como clave simétrica para el algoritmo de cifrado AES-128 en modo CBC.

Una vez tengamos tratada la clave deberemos cifrar el parámetro utilizando el algoritmo AES-128 en modo CBC, con las siguientes condiciones:

- La clave es el resultado del paso anterior.
- El vector de inicialización (IV) debe ser una secuencia de 16 bytes con valor a ceros.

El resultado del cifrado debe ser codificado en Base64 URL Safe manteniendo el padding de iguales.

Número de tarjeta	4548*****7229
Número de tarjeta cifrado en AES-128 CBC con la clave facilitada en el alta del PSP (tratada como se explicó antes) usando un vector de 16 bytes a ceros y convertido a Base64 URL Safe manteniendo el padding de =.	W4ye36P-VrrEjmE79cBHJfZevp2_YSmC2Rz8c- upsus=

Cadena en JSON con PAN cifrado:

```
{
  "DS_MERCHANT_AMOUNT": "145",
  "DS_MERCHANT_ORDER": "1446083946",
  "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": "W4ye36P-VrrEjmE79cBHjFZevp2 YSmC2Rz8c-upsus-",
  "DS_MERCHANT_EXPIRYDATE": "4912",
  "DS_MERCHANT_CVV2": "123"
}
```

A continuación, se muestra el objeto JSON 'DS_MERCHANTPARAMETERS' codificado en BASE64.

eyJEU19NRVJDSEFOVF09BTU9VTlQlOiilcDUiLCJEU19NRVJDSEFOVF09PUkrFUi6lEjE0NDYwODM5NDYiLCJEU19NRVJDSEFOVF09NRVJDSEFOVENPREUI0iil5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF09DVJSRU5DWSI6lj3CIsicRTXO1FUkNlQU5UX1RSQU5TQUNUSU9OVFIQRSl6ljaIlcJEU19NRVJDSEFOVF09URVINSU5BTCi6lEjEiLCJEU19NRVJDSEFOVF09NRVJDSEFOVFVSTCI6Imh0dHA6Ly93d3ducHJ1ZWJhLmNvbS59cmxOb3RpZmljYWNPpb24ucGhwiiwiRfNFJTUVSQ0hBTIRfUEF0ljoivzRSZTM2UC1WcnJFam1FNzljQkhqRlpIdnAyX1TbUmYUno4Yy1lcHN1cz0iLCJEU19NRVJDSEFOVF09FBUllEQVRfJioiNDksMiIsiksrTXO1FUkNlQU5UX0NWVjiOiilcWJfMjF0

Firmar los datos de la petición

Sobre toda la cadena obtenida en el paso anterior (DS_MERCHANTPARAMETERS) se calcula la firma completa con la clave HMAC SHA512.

Obtenemos la siguiente signature en BASE 64 URL Safe (manteniendo padding de =) con la clave de firma facilitada en el alta del PSP y el campo DS_MERCHANTPARAMETERS:

72_72cEGArqk1yyGlljbd_xf1mOFQVh2H5wskowPBj2YdeY6pEQnhmvMoWrLnKqoCKooWrV
VwtGZNIjYH-ckDg==

Formar el mensaje de la petición

- `Ds_SignatureVersion: T25V2`
 - `Ds_MerchantParameters:`
 - `eyJEU19NRVJDSEFOVF9BTU9VTiQiOiNDUilCJEU19NRVJDSEFOVF9Pb24ucGhwlwifRfUjE0NDYwODM5NDYiLCJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUjOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9DZVVJSRU5DWSi6lj3OCiSkRTX01FUkNlQU5UX1RSQU5TQUUNUSU9OVFIQRSl6ljlailCJEU19NRVJDSEFOVF9URVJINSU5BTCi6ljiLCJEU19NRVJDSEFOVF9NRVJDSEFOVFSTCiglmdh0dHA6lY93d3cucHJ1ZWJhLmNvbS91cmxOb3RpZmZlYWNpbi24ucGhwlwifRfUjE0NDYwODM5NDYiLCJEU19NRVJDSEFOVF9F9FWBJUllEQVRFljoindkxMlslkRTX01FUKNIQU5UX0NWVWjiiOilxMjMifQ==`
 - `Ds_Signature:`
- `72_72cEGArqk1yyGlljbD_xf1mOFQVh2H5wskowPBj2YdeY6pEQnhmvMoWrLnKqoCKooWrVVwtGZNIjYH-ckDg==`

MPI/3DSServer externo

En el caso de que el PSP o integrador disponga de su propia certificación para realizar la gestión de la autenticación de forma externa a Redsys, podrá realizar transacciones indicando que la operación ya ha sido autenticada. El PSP tiene que solicitar a la Entidad Adquirente la configuración para realizar esta operativa.

Es requisito imprescindible que el PSP esté certificado con EMVCO y con las marcas.

Las peticiones se realizarán como se ha indicado en este mismo documento, añadiendo en el campo DS_MERCHANTPARAMETERS el parámetro DS_MERCHANT_MPIEXTERNAL, del tipo JSON Object.

Campos a incluir en una operación autenticada con EMV 3DS:

- **threeDSServerTransID:** identificador de la transacción utilizado en los mensajes de autenticación.
- **authenticationValue:** valor devuelto por el ACS del emisor de la tarjeta.
- **dsTransID:** identificador del Directorio utilizado en los mensajes de autenticación.
- **protocolVersion:** versión de protocolo EMV 3DS con la que se ha realizado la autenticación.
- **Eci:** valor del campo Eci devuelto por el ACS en la autenticación.
- **authenticationFlow:** campo opcional para indicar si la autenticación se ha realizado con Challenge (C) o con Frictionless (F). Si no se indica nada, se considerará que se ha realizado con Challenge.

A continuación, se muestra un ejemplo con los datos a incluir en el Ds_MerchantParameters para enviar una petición de autenticación con 3DSServer externo en versión 2:

```
{
  "DS_MERCHANT_ORDER":1552572812,
  "DS_MERCHANT_MERCHANTCODE":"999008881",
  "DS_MERCHANT_TERMINAL":"2",
  "DS_MERCHANT_CURRENCY":"978",
  "DS_MERCHANT_TRANSACTIONTYPE":"0",
  "DS_MERCHANT_AMOUNT":"1000",
  "DS_MERCHANT_PAN":"XXXXXXXXXXXXXXXXXX ",
  "DS_MERCHANT_EXPIRYDATE":"XXXX",
  "DS_MERCHANT_CVV2":"XXX",
  "DS_MERCHANT_MPIEXTERNAL ":{
    "threeDSServerTransID":"7bb7e07d-d2e5-467e-9e96-69da4542b759",
    "dsTransID":"7bb7e07d-d2e5-467e-9999-69da4542b770",
    "authenticationValue":"jBXyoylcA+KICREAAA/aAAkAAAA=",
    "protocolVersion":"2.1.0",
    "Eci":"05",
    "authenticationFlow":"C"
  }
}
```

PUCE (autenticación)

Este tipo de integración permite a los comercios/integradores que realizan las peticiones de pago de comercio electrónico por protocolo PUC (PRICE comercios), realizar la autenticación del titular mediante protocolo EMV 3DS. Para realizar este tipo de integración ver *Manual integración PUCE-REST*.

Estos comercios/integradores pueden utilizar el protocolo REST para realizar la autenticación del titular mediante protocolo EMB 3DS y, una vez autenticado el titular, enviar la autorización por conexión PUC/PRICE. Este envío de autorización financiera a través de PUC no es estrictamente necesario, ya que con esta misma integración REST es posible realizar también la petición de autorización, es decir, con esta integración REST el comercio/integrador, puede hacer todo el flujo de la operación, la autenticación del titular y la petición de autorización. En este caso deben seguir el proceso de integración descrito en esta guía.

12. 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 bancaria para que le facilite los datos de acceso.

Las URLs de acceso al entorno de pruebas son:

URL Conexión inicia petición	Tipo integración
https://sis-t.redsys.es:25443/sis/rest/iniciaPeticonREST	Comercio
https://sis-t.redsys.es:25443/sis/rest/iniciaPeticonPSPREST	PSP
URL Conexión trata petición	
https://sis-t.redsys.es:25443/sis/rest/trataPeticonREST	Comercio
https://sis-t.redsys.es:25443/sis/rest/trataPeticonPSPREST	PSP
URL acceso al Portal de Administración	
https://sis-t.redsys.es:25443/canales/portal	Comercio/PSP

NOTA: El flujo de las operaciones en el entorno de pruebas es el mismo que en el entorno de producción, 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 secreta integración comercios: sq7HjrUOBfKmC576ILgskD5srU870gJ7

Tarjeta de pruebas	Descripción
--------------------	-------------

4918019160034602 Caducidad: no se valida CVV2: distinto de 999	Autenticación Frictionless · EMV3DS 2.1 con threeDSMethodURL.
4548814479727229 Caducidad: no se valida CVV2: distinto de 999	Autenticación Frictionless · EMV3DS 2.1 sin threeDSMethodURL.
4918019199883839 Caducidad: no se valida CVV2: distinto de 999	Autenticación Challenge · EMV3DS 2.1 con threeDSMethodURL.
4548817212493017 Caducidad: no se valida CVV2: distinto de 999	Autenticación Challenge · EMV3DS 2.1 sin threeDSMethodURL.
Operaciones denegadas	
Cualquier operación con valor de CVV2 = 999 o importe terminado en 96 Para realizar una prueba de denegación de operación por código de respuesta 172, 173 o 174 basta con usar el valor del código como CVV o poner un importe terminado en 72, 73 o 74. Esto sólo funcionará en los entornos de prueba.	
Pruebas PSD2	
4548816134581156 Caducidad: no se valida CVV2: distinto de 999	Autenticación Frictionless · EMV3DS 2.2 sin threeDSMethodURL
4548816131164386 Caducidad: no se valida CVV2: distinto de 999	Autenticación Challenge · EMV3DS 2.2 sin threeDSMethodURL
4548815324058868 Caducidad: no se valida CVV2: distinto de 999	EMV3DS 2.2 sin threeDSMethodURL. 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 sin threeDSMethodURL. Si se envía MIT devolverá Frictionless, en cualquier otro caso pedirá Challenge
5576441563045037 Caducidad: no se valida CVV2: distinto de 999	EMV3DS 2.2 sin threeDSMethodUR. Acepta solo pagos 3RI-OTaE

4548817212493017 Caducidad: no se valida CVV2: other than 999	Card with soft decline. Denial 195, if the authorization is not authenticated.
Pruebas DCC	
5424180805648190 Caducidad: no se valida CVV2: distinto de 999	Autenticación Frictionless · Tarjeta Master con DCC y EMV3DS
4117731234567891 Caducidad: no se valida CVV2: distinto de 999	Autenticación Challenge · Tarjeta Visa con DCC y EMV3DS
5409960031405146 Caducidad: no se valida CVV2: distinto de 999	Autenticación Challenge · Tarjeta Master con moneda Noruega y EMV3DS
5171471234567894 Caducidad: no se valida CVV2: distinto de 999	Autenticación Challenge · Tarjeta Master con moneda Libra y EMV3DS
4001151234567891 Caducidad: no se valida CVV2: distinto de 999	Autenticación Challenge · Tarjeta Visa con moneda Libra y EMV3DS
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

13. Ejemplos de tipos de operación más habituales

En los siguientes puntos se indican los parámetros a incluir en algunos de los tipos de operación más comunes. Se incluyen ejemplos de los campos incluidos en parámetro Ds_MerchantParameters. Los detalles de cómo se monta una petición REST al TPV Virtual se indican en el punto 3 de este documento.

Petición de pago/preautorización (con envío de datos de tarjeta sin autenticación)

En el campo DS_MERCHANT_TRANSACTIONTYPE se indica el tipo de operación que se desea realizar:

- * DS_MERCHANT_TRANSACTIONTYPE:"0" para PAGO
- * DS_MERCHANT_TRANSACTIONTYPE:"1" para PREAUTORIZACIÓN

El valor de Ds_MerchantParameters sería:

```
{ "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_PAN":"XXXXXXXXXXXX",
  "DS_MERCHANT_EXPIRYDATE":"XXXX",
  "DS_MERCHANT_CVV2":"XXX"}
```

Petición de Confirmación/Devolución/Anulación

En este caso no es necesario informar los datos de tarjeta y en el campo DS_MERCHANT_ORDER se debe indicar la operación. En el campo DS_MERCHANT_TRANSACTIONTYPE se indica el tipo de operación que se desea realizar:

- * DS_MERCHANT_TRANSACTIONTYPE:"2" para CONFIRMACIÓN
- * DS_MERCHANT_TRANSACTIONTYPE:"3" para DEVOLUCIÓN
- * DS_MERCHANT_TRANSACTIONTYPE:"9" para ANULACIÓN

El valor de Ds_MerchantParameters sería:

```
{ "DS_MERCHANT_ORDER":"1552565870",
  "DS_MERCHANT_MERCHANTCODE":"999008881",
  "DS_MERCHANT_TERMINAL":"999",
  "DS_MERCHANT_CURRENCY":"978",
  "DS_MERCHANT_TRANSACTIONTYPE":"3",
  "DS_MERCHANT_AMOUNT":"1000"}
```

Petición de Tokenización (Pago por Referencia - Pago 1-Clic)

Se debe incluir el campo DS_MERCHANT_IDENTIFIER=REQUIRED para solicitar la generación de la referencia. Se debe incluir también el campo DS_MERCHANT_COF_TYPE para indicar el uso que se va a realizar de dicha referencia.

El valor de Ds_MerchantParameters sería:

```
{ "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_PAN":"XXXXXXXXXXXX",
  "DS_MERCHANT_EXPIRYDATE":"XX",
  "DS_MERCHANT_CVV2":"XX",
  "DS_MERCHANT_IDENTIFIER":"REQUIRED",
  "DS_MERCHANT_COF_TYPE":"R"
}
```

NOTA: por requisito de PSD2 se deberá solicitar autenticación SCA por lo que también se deberán incluir los campos relacionados con la autenticación.

Petición de pago con Tokenización (Pago por Referencia - Pago 1-Clic)

Se incluye el DS_MERCHANT_IDENTIFIER con el valor de la referencia en lugar de los datos de tarjeta

El valor de Ds_MerchantParameters sería:

```
{ "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_IDENTIFIER":"XXXXXXXXXXXXXXXXXXXXXXXXXXXX"}
}
```

14. Timeout

¿Qué hacer en el caso de que el TPV Virtual no responda a una petición solicitada?

Este problema puede tener dos posibles causas:

- No se ha recibido la petición, con lo que TPV Virtual no responderá al mensaje de petición.
- El TPV Virtual ha recibido el mensaje de petición, pero no puede contactar con el Centro Autorizador. Esta conexión tiene definido un timeout de 30 segundos, por lo que si transcurrido ese tiempo, no se recibe respuesta del Centro Autorizador, se devolverá un mensaje de respuesta con código 9912/912 "Emisor no disponible". La aplicación cliente deberá por tanto establecer un timeout mayor (unos 40 o 50 segundos), para asegurar que TPV Virtual siempre le va a responder.

¿Qué hacer en caso de timeout?

Para las peticiones de pago, preautorización o confirmación se deberá enviar su operación de anulación correspondiente.

En el caso de operaciones de devoluciones u operaciones de anulaciones se podrá volver a realizar la petición.

15. Errores frecuentes

Error de firma (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 módulo de administración (apartado comercios).
- Se debe revisar que los comercios no están enviando espacios en blanco en la firma. Si la petición se hace mediante cURL o mediante el navegador Safari, puede que se conviertan los símbolos "+" en espacio en blanco. Para que esto no ocurra se deben sustituir los símbolos "+" de la firma por "%2B" (Valor URL encoded).
- Si el comercio no consigue localizar qué parámetro es el erróneo, debe contactar con el Centro de Atención al Cliente de Redsys, o con el departamento de Soporte a la Integración de Redsys, si su entidad le ha facilitado el contacto.

Tengo en mi comercio denegaciones por número de repetido (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. Si necesita este permiso debe ponerse en contacto con su entidad.

16. Preguntas frecuentes

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

En el punto 3 de este mismo documento se indica cómo acceder al valor de clave.

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

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

ANEXO 1

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 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*.

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 carpeta *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*. Además también estableceremos la variable con la que comunicaremos a Redsys el algoritmo que hemos usado para la firma (HMAC SHA-512) tal y como se muestra a continuación:

```
$miObj = new RedsysAPI512; // Para API HMAC SHA 512
$signatureVersion = "HMAC_SHA512_V1";
```

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->createMerchantParameters();
```

4. Calcular el parámetro Ds_Signature. Para llevar a cabo el cálculo de este parámetro, se debe llamar a la función de la librería "createMerchantSignature()" con la clave obtenida del módulo de administración, tal y como se muestra a continuación:

```

$claveModuloAdmin = "sq7HjrUOBfKmC576ILgskD5srU870gJ7";
$signature = $miObj->createMerchantSignature($claveModuloAdmin);

```

5. 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.

1.2 Librería JAVA

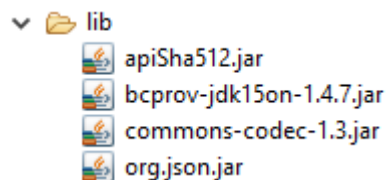
A continuación, se presentan los pasos que debe seguir un comercio para la utilización de la librería JAVA proporcionada por Redsys. Compatible con los algoritmos HMAC SHA-512:

HMAC SHA-512

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

```
<%@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
String signatureVersion = "HMAC_SHA512_V1";

```

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 se debe llamar a la función de la librería `createMerchantParameters()`, tal y como se muestra a continuación:

```
String params = miObj.createMerchantParameters();
```

4. Calcular el parámetro `Ds_Signature`. Para llevar a cabo el cálculo de este parámetro, se debe llamar a la función de la librería `createMerchantSignature()` con la clave obtenida del módulo de administración, tal y como se muestra a continuación:

```
String claveModuloAdmin = "sq7HjrUOBfKmc576ILgskD5srU870gJ7";
String signature = miObj.createMerchantSignature(claveModuloAdmin);
```

5. 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`.

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 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 carpeta

`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.

- Definir un objeto de la clase principal de la librería, en este caso llamada *RedsysAPI512*:

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

- Capturar los parámetros de la notificación on-line:

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

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

```
$decode = $miObj->decodeMerchantParameters($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 llevar a cabo la obtención del valor de un parámetro se debe llamar a la función “getParameter()” de la librería con el nombre de parámetro, tal y como se muestra a continuación para obtener el código de respuesta:

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

NOTA IMPORTANTE: Para garantizar la seguridad y el origen de las notificaciones el comercio debe llevar a cabo la validación de la firma recibida y de todos los parámetros que se envían en la notificación.

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

```
$claveModuloAdmin = "sq7HjrUOBfKmC576ILgskD5srU870gJ7";
$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 === $firmaRecibida) {
    echo "FIRMA OK. Realizar tareas en el servidor";
} else {
    echo "FIRMA KO. Error, firma inválida";
}
```

2.2 Librería JAVA

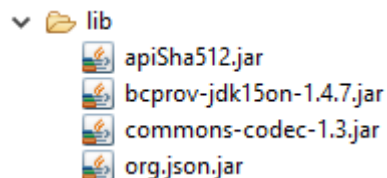
A continuación, se presentan los pasos que debe seguir un comercio para la utilización de la librería JAVA proporcionada por Redsys. Compatible con los algoritmos *HMAC SHA-512*:

HMAC SHA-512

- 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.



- 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
```

- 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");
```

- 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 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 = 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");
}
```

2.3 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 = "sq7HjrU0BfKmc576ILgskD5srU870gJ7";
string signatureCalculada = r.createMerchantSignatureNotif(kc, parms);
```

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

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

ANEXO 2

Petición con firma en HMAC SHA-256

Para generar petición con la firma en HMAC se deben realizar lo siguiente:

- 1) Se genera una clave de firma específica por operación. Esta clave se obtiene al cifrar el número de pedido de la operación (Ds_Merchant_Order) con la clave del comercio utilizando un cifrado 3DES (modo CBC).

Si partimos de la clave de comercio en base 64 (obtenida desde el portal de administración):

sq7HjrUOBfKmC576iLgskD5srU870gJ7 y el número de pedido **06080232580** tenemos los siguientes resultados:

Clave del comercio en representación hexadecimal:

b2aec78eb50e05f2a60b9efa20b82c903e6cad4f3bd2027b

Clave de la operación en representación hexadecimal:

a5334014a4f010c8779cef789886c123

- 2) Montamos el Ds_MerchantParameters y lo codificamos en base 64.

Ejemplo:

```
{ "DS_MERCHANT_MERCHANTCODE": "999008881", "DS_MERCHANT_TERMINAL": "1", "DS_MERCHANT_ORDER": "06080232580", "DS_MERCHANT_AMOUNT": "100", "DS_MERCHANT_CURRENCY": "978", "DS_MERCHANT_TRANSACTIONTYPE": "3" }
```

```
eyJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9URVJNSU5BT  
CI6IjEiLCJEU19NRVJDSEFOVF9PUkRFUiI6IjA2MDgwMjMyNTgwIiwiaWIRFNFTUVSQ0hBTlRfQU1PVU5UIj  
oiMTAwIiwiaWIRFNFTUVSQ0hBTlRfQ1VSUkVQ1kiOiI5NzgiLCJEU19NRVJDSEFOVF9UUkFOU0FDVElPTlR  
ZUEUiOiIzIn0=
```

- 3) Se calcula el HMAC SHA256 del valor del parámetro Ds_MerchantParameters en Base64 con la clave de operación obtenida en el paso anterior.

Ejemplo: Valor de firma en formato hexadecimal:

1a61f04e8bed872af3b4bb3b0fbec67e5725073b0035d7ab1dedf3145e898994

- 4) El resultado obtenido se codifica en BASE 64, y éste será el valor del parámetro Ds_Signature.

Ejemplo: Valor de la firma en Base 64:

GmHwTovthyrztLs7D77GfclBzsANderHe3zFF6JiZQ=

- 5) Tipo de firma, Ds_SignatureVersion. Para este tipo de firma el valor del campo será: **HMAC_SHA256_V1**

- 6) Montamos la cadena resultante con los tres campos obtenido anteriormente:
- ```
{ "Ds_MerchantParameters": "eyJEU19NRVJDSEFOVF9NRVJDSEFOVENPREUiOiI5OTkwMDg4ODEiLCJEU19NRVJDSEFOVF9URVJNSU5BTCI6IjEiLCJEU19NRVJDSEFOVF9PUkRFUiI6IjA2MDgwMjMyNTgwIiwiaWwifnFtUVVSQ0hBTlRfQU1PVU5UIjoimTAwIiwiaWwifnFtUVVSQ0hBTlRfQ1VSUkVOQ1kiOiI5Nzg1LCJEU19NRVJDSEFOVF9UUkFOU0FDVlPTlRZUEUiOiIzIn0=", "Ds_Signature": "GmHwTovthyrztLs7D77GflclBzsANderHe3zFF6JiZQ=", "Ds_SignatureVersion": "HMAC_SHA256_V1" }
```

**NOTA:** Puede utilizar las librerías de ayuda para generar la firma.