

# **Integración inSite**

Versión: 3.0

18/04/2022

RS.ADQUI.CNOPRESENCECOMM.LIST.0000



Redsys · C/ Francisco Sancha, 12 · 28034 · Madrid · ESPAÑA

## Autorizaciones y control de versión

Versión	Fecha	Afecta	Breve descripción del cambio
1.0	03/04/2017	Todo el documento	Versión inicial del documento
1.1	28/05/2018	Todo el documento	Inclusión del epígrafe 2 de Conceptos y Ventajas, e inclusión de mejoras en la redacción para mayor claridad y entendimiento.
1.2	06.03.2018	Se modifica url de descarga de librerías	Se modifica la url de descarga de librerías (punto 5)
1.3	07.03.2019	Se incluye la url de acceso al entorno de test	En el punto 4 se incluye la url de acceso al entorno de test.
1.4	05.07.2019	Se incluye el nuevo flujo de autenticación del SIS vía WS/REST	En el punto 6 se incluye el nuevo flujo de autenticación vía WS /REST
2.0	15/10/2019	Todo el documento	Se incluye la versión 2 con nuevas funcionalidades.
2.1	18/11/2019	Todo el documento	Modificaciones en el interfaz e inclusión de nuevas funcionalidades
2.2	09/12/2021	Punto 4.1	Se incluye nuevo parámetro opcionar para ocultar el logo de la entidad.
2.3	18/04/2022	Punto 5	Apartado mejoras versión inSite V3
3.0	18/04/2022	Todo el documento	Se actualiza documento a inSite V3 con nuevas funcionalidades

## ÍNDICE DE CONTENIDO

<b>1. Objetivo de esta guía .....</b>	<b>4</b>
<b>2. Conceptos y ventajas de la conexión inSite.....</b>	<b>5</b>
<b>3. Descripción general del flujo .....</b>	<b>6</b>
<b>4. Página de Pago – Obtención de ID de operación .....</b>	<b>7</b>
4.1    Integración unificada (todo en uno) .....	7
4.2    Integración de elementos independientes .....	13
<b>5. Catálogo de errores.....</b>	<b>16</b>
<b>6. Catálogo de idiomas .....</b>	<b>17</b>
<b>7. Envío de operación tras generación de ID de operación .....</b>	<b>19</b>
6.1    Implementación sin uso de las librerías de ayuda .....	19
<b>8. Identificación de operaciones inSite en el Portal de Administración del TPV Virtual .....</b>	<b>21</b>
<b>9. Autenticación 3DSecure en InSite .....</b>	<b>21</b>
8.1    3DSecure v1.0.2 .....	21
8.1.1    Solicitar autorización .....	21
8.1.2    Ejecución de la autenticación .....	23
8.1.3    Confirmación de autorización 3DSecure 1.0 posterior al Challenge .....	24
8.2    EMV3DS .....	25
8.2.1    Iniciar Petición.....	25
8.2.2    Ejecución del 3DSMethod .....	26
8.2.3    Petición de autorización con datos EMV3DS .....	28
8.2.4    Ejecución del Challenge .....	30
8.2.5    Confirmación de autorización EMV3DS posterior al Challenge .....	31

Redsys · C/ Francisco Sancha, 12 · 28034 · Madrid · ESPAÑA

## 1. Objetivo de esta guía

En este documento se describe cómo implementar en una tienda web la conexión **inSite** del TPV Virtual, un modelo de conexión que permite recoger los datos de pago del cliente sin que éste tenga que abandonar la página web del comercio.

Las ventajas de este tipo de integración son varias y se describen con mayor detalle en el siguiente epígrafe. El objetivo principal es el de disponer de un proceso de pago rápido, sencillo e integrado al máximo en las páginas de la tienda web, adaptado completamente al diseño del comercio online, fácil de usar y de integrar, pero a la vez que mantiene la seguridad sobre los datos de pago introducidos por el cliente, evitando que el comercio tenga que soportar costosos procesos de seguridad derivados del cumplimiento obligatorio de la normativa PCI DSS<sup>1</sup>.

**Esta guía se centra en las particularidades de este tipo de integración. Para conceptos generales del funcionamiento del servicio de TPV Virtual SIS por favor consulte la documentación correspondiente.**

---

<sup>1</sup> PCI DSS (Payment Card Industry Data Security Standard) establece los requerimientos de seguridad que los intervenientes en el proceso de pago con tarjetas deben cumplir. La solución descrita en el documento facilita la consideración del proceso como un tipo SAQ-A, al basar la implementación en iframes cuyo contenido sólo es accesible por nuestros servidores.

## 2. Conceptos y ventajas de la conexión inSite

Con la solución de pago **inSite** el comercio o tienda online consigue una serie de ventajas que favorecen el aumento de la conversión de ventas:

- Una **experiencia de pago sencilla y satisfactoria** para sus clientes, al estar **totalmente integrada** en las páginas web del comercio y sin saltos de navegación.
- **Mayor control** del flujo de checkout y pago, ya que toda petición se realiza de forma síncrona por parte del servidor de la tienda web y sin necesidad de procesos asíncronos de “escucha”.
- **Facilidad de uso en su integración,**
- **Alto nivel de seguridad**, similar a la solución basada en redirección del cliente hacia una página de pago externa.

En definitiva, además de un proceso de pago totalmente integrado en el checkout al comprador, se permite al comercio una mayor **flexibilidad y control** en el proceso de pago, pudiendo además separar los pasos de captura de datos y ejecución de la operación.

A la hora de integrar la conexión **inSite** existen **dos posibilidades**:

- Integración unificada (todo en uno)
- Integración por elementos independientes

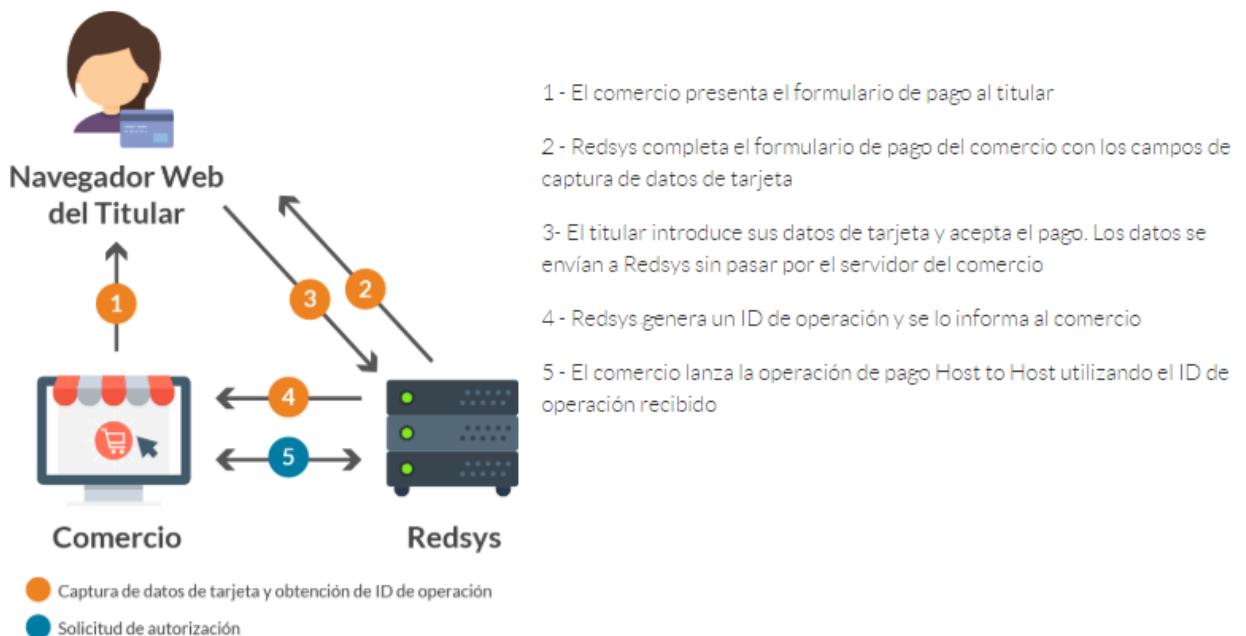
En ambos casos, la integración se puede realizar utilizando fragmentos de código que se exponen como ejemplos, donde sólo se requiere cambiar valores propios como el identificador del comercio o las claves utilizadas. Además, como ayuda adicional se proveen librerías proporcionadas por Redsys para los principales lenguajes de programación.

En la conexión **inSite**, se facilitan a la tienda online las piezas o “campos” necesarios del formulario de pago de forma que se integran uno a uno (o como un conjunto) perfectamente incrustados en la página checkout de la tienda web y además cada elemento permite personalización del diseño con estilos configurables, en perfecta sintonía del diseño del resto de la página web del comercio.

La seguridad se preserva de forma que el formulario resultante con la información de pago de los clientes queda inaccesible al mismo servidor del comercio o incluso de terceros que hayan podido comprometer el servidor web del comercio.

### 3. Descripción general del flujo

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



En resumen, los datos de pago introducidos por el cliente son enviados desde la página del comercio al TPV Virtual, donde se almacenan temporalmente y se asocian a un Id de Operación que se devuelve al comercio. Con este Id de Operación (que viene a ser un "alias" de los datos de pago del cliente) el comercio puede solicitar posteriormente y directamente al tpv virtual la realización de la operación de pago deseada.

## 4. Página de Pago – Obtención de ID de operación

Como primer paso para poder integrar los campos de introducción de datos de tarjeta directamente en su propia página web, se debe incluir el fichero Javascript alojado en el servidor de Redsys con la siguiente línea de código (el fichero varía según se vaya a usar el entorno de test o el entorno de producción real):

- Entorno de Test:

```
<script src="https://sis-t.redsys.es:25443/sis/NC/sandbox/redsysV3.js"></script>
```

- Entorno de Producción:

```
<script src="https://sis.redsys.es/sis/NC/redsysV3.js"></script>
```

El siguiente paso para incluir los elementos del formulario de pago depende de la alternativa que se desee implementar. A la hora de integrar la conexión **inSite** existen **dos posibilidades**:

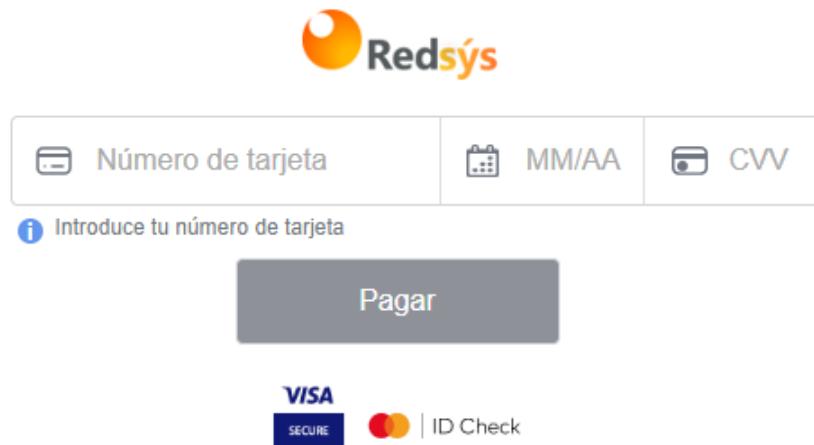
- a) **Integración unificada (todo en uno)**: Los elementos de pago, como las cajas de introducción de número de tarjeta, fecha de caducidad, cvv.. y botón de pago se incrustan como un solo elemento que se adapta a la página del comercio (responsive), con diseño ligero y estilos CSS personalizables. Incluye por defecto ayudas interactivas animadas y una buena usabilidad al usuario.
- b) **Integración por elementos independientes**: los campos se deben incrustar cada uno de forma independiente dentro de la página web de la tienda web, lo que permite el control total del diseño, posición, gestión de los errores, etc.

### 4.1 Integración unificada (todo en uno)

En esta modalidad de la integración inSite se proveerá un único iframe de tamaño muy ajustado en el que se incluirá el formulario de pago al completo. En cuanto a la personalización del mismo, se podrán aplicar los estilos CSS que el comercio requiera a los diferentes elementos.

Incluye elementos interactivos que facilitan la usabilidad, como el reconocimiento de la marca de tarjeta, mostrando el logo de la misma, verificación de los formatos y contenidos y resaltando visualmente alguno es incorrecto (check digit, fecha cad...).

Ejemplo:



Una vez importado el fichero JS, se deberá crear el formulario de pago. Para recoger de forma segura los datos de tarjeta, Redsys creará y alojará los campos de introducción de dichos datos.

Se deberán crear un único contenedor, con un id único, ya que se deberá indicar para que se genere iframe con los elementos en él.

```
<div id="card-form-example"></div>
```

Se debe incluir una función de escucha de mensajes (listener) para recibir el ID de operación cuando éste se genere. Se debe utilizar la función *storeIdOper* con la siguiente definición:

```
storeIdOper(event, idToken, idErrorMsg, validationFunction);
```

en la que se deberá indicar el evento recogido por el listener (*event*), el ID del elemento del DOM se debe almacenar el ID de operación una vez sea generado (*idToken*), el identificador del elemento en el que se almacenarán los códigos de error en caso de que existan errores de validación de los datos (*idErrorMsg*). En el ejemplo posterior ambos se almacenan en un input de tipo "hidden".

Opcionalmente, se establece la posibilidad de ejecutar una función propia para realizar validaciones previas por parte del comercio. Únicamente se continuará con la generación del ID de operación si la función de validación ejecutada por el comercio retorna un valor *true*.

```
<input type="hidden" id="token" ></input>
<input type="hidden" id="errorCode" ></input>
<script>
function merchantValidationEjemplo(){
//Insertar validaciones...
return true;
}
<!-- Listener -->
window.addEventListener("message", function receiveMessage(event) {
    storeIdOper(event, "token", "errorCode", merchantValidationEjemplo);
});
</script>
```

Una vez preparado el listener para la recepción de los datos, se llamará a la función proporcionada para generar los elementos de introducción de datos de tarjeta. Hay disponibles dos funciones, pudiendo usar la que prefiramos. En la primera pasaremos cada dato en un parámetro (`getInSiteForm()`), y en la segunda pasaremos los datos en formato JSON(`getInSiteFormJSON()`). La ventaja de esta última es que podremos pasar solo los datos que necesitemos, sin necesidad de enviar parámetros vacíos.:

```
<!-- Petición de carga de iframe clásica-->
getInSiteForm(idContenedor, estiloBoton, estiloBody, estiloCaja, estiloInputs, buttonValue, fuc,
terminal, merchantOrder, idiomalnsite, mostrarLogo, estiloReducido, estiloInsite);

<!-- Ejemplo -->
getInSiteForm('card-form', "", "", "", 'Texto botón pago', '123456789', '1', 'ped4227', 'ES', true,
false, 'twoRows');

<!-- Petición de carga de iframe JSON-->
var insiteJSON = {
    "id" : "card-form",
    "fuc" : "123456789",
    "terminal" : "1",
    "order" : "ped4227",
    "estiloInsite" : "inline"
}
```

getInSiteFormJSON(insiteJSON);

Como parámetros de las funciones se indicará el id del contenedor reservado para su generación, así como el estilo requerido para los diferentes elementos (formato CSS). En esta modalidad, se podrán incluir estilos para diferentes elementos:

- Botón de pago → Se permite la personalización completa del botón de pago.
- Cuerpo del formulario → Se recomienda utilizar para establecer un color de fondo o modificar el color o estilo de los textos.
- Caja de introducción de datos → Se podrá establecer un color de fondo diferenciado para la caja de introducción de datos. El color del texto aplicado en este elemento se aplicará al “placeholder” de los elementos.
- Inputs de introducción de datos → Se recomienda su uso si se quiere utilizar un tipo de letra diferente o modificar el color del texto de los campos de introducción de datos.

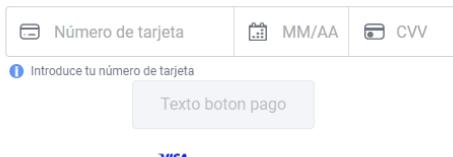
Adicionalmente, se podrá personalizar el texto a incluir en el botón y, por último, se deberá informar el valor del FUC, terminal y número de pedido (cadena de texto alfanumérica de entre 4 y 12 posiciones) en la petición de carga del iframe con el formulario de pago.

Se incluyen cuatro parámetros opcionales en la carga del iframe.

Los parámetros opcionales son los siguientes (en orden):

- **Idioma:** Indica el idioma de los textos. La relación de códigos se puede encontrar en el apartado [Catálogo de idiomas](#).  
Se puede usar tanto el código SIS como el código ISO 639-1 del idioma.  
En caso de no establecerse ningún idioma o informar un código incorrecto, el idioma por defecto será el Castellano.
- **Logo entidad.** Establece si se quiere mostrar el logo de la entidad, indicando **true** si se desea mostrarlo o **false** si en el caso contrario.  
En caso de no establecer ningún valor, por defecto se mostrará el logo de la entidad.
- **Estilo reducido.** Indica si se quiere mostrar Insite con un ancho reducido.
- **Estilo de Insite.** Se puede elegir entre dos estilos predefinidos de Insite, '**inline**' o '**twoRows**'. Por defecto se pintará con el estilo 'inline'. A continuación, se puede ver un ejemplo de cada estilo de Insite con sus respectivas versiones reducidas:

Inline



Este formulario muestra los campos para introducir el número de tarjeta, la fecha de vencimiento y el CVV en una sola fila horizontal. Abajo de los campos hay un botón "Texto botón pago" y las logos de VISA, Mastercard y American Express.

Inline reducido



Este formulario es similar al anterior pero con un diseño más compacto, agrupando los campos de fecha y CVV en un solo cuadro.

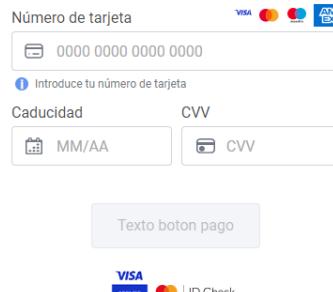
  

TwoRows



Este formulario organiza los campos en dos filas: el número de tarjeta y las demás opciones (fecha y CVV) en la primera fila, y el botón de pago y los logos en la segunda fila.

TwoRows reducido



Este formulario es similar al TwoRows pero con un diseño más compacto.

Puede consultar la lista de parámetros para enviar los datos en formato JSON a continuación:

Parámetro	Obligatorio / Opcional	Por defecto	Por defecto
id	Obligatorio	-	-
fuc	Obligatorio	-	-
terminal	Obligatorio	-	-
order	Obligatorio	-	-
styleButton	Opcional	Estilo CSS	""
styleBody	Opcional	Estilo CSS	""
styleBox	Opcional	Estilo CSS	""
styleBoxText	Opcional	Estilo CSS	""
buttonValue	Opcional	Texto botón pago	Pagar
idiomaInsite	Opcional	<a href="#">Catálogo de idiomas</a>	ES
mostrarLogoInsite	Opcional	true, false	true
estiloReducidoInsite	Opcional	true, false	false
estiloInsite	Opcional	inline, twoRows	inline

**\*El merchantOrder utilizado en la carga del iframe y generación del idOper deberá reutilizarse en la posterior petición de autorización.**

De esta forma, cuando el cliente introduzca sus datos de tarjeta en los elementos generados por Redsys y pulse el botón de pago, se generará y almacenará en el formulario del comercio un ID asociado a la operación para que éste formalice la compra sin necesidad de tratar datos de tarjeta.

## 4.2 Integración de elementos independientes

En esta modalidad de la integración inSite se permitirá a los comercios una total personalización de la página de pago, por lo que podrá colocar los campos de introducción de datos de tarjeta y el botón de pago con total libertad, al generar iframes diferenciados y personalizables con estilos para cada uno de ellos.

También se incluyen elementos para mejorar la usabilidad como, por ejemplo, el reconocimiento de la marca de tarjeta mostrando el icono de ésta, longitud del CVV teniendo en cuenta la marca de la tarjeta o el formato de fecha de caducidad.

Una vez importado el fichero, se deberá crear el formulario de pago. Para recoger de forma segura los datos de tarjeta, Redsys creará y alojará los campos de introducción de dichos datos.

Se deberán crear contenedores vacíos, con un id único, ya que se deberá indicar para que se genere el campo de introducción de datos en él.

```
<div class="cardinfo-card-number">
    <label class="cardinfo-label" for="card-number">Número de tarjeta</label>
    <div class='input-wrapper' id="card-number"></div>
</div>
<div class="cardinfo-exp-date">
    <label class="cardinfo-label" for="expiration-month">Mes Caducidad (MM)</label>
    <div class='input-wrapper' id="expiration-month"></div>
</div>
<div class="cardinfo-exp-date2">
    <label class="cardinfo-label" for="expiration-year">Año Caducidad (AA)</label>
    <div class='input-wrapper' id="expiration-year"></div>
</div>
<div class="cardinfo-cvv">
    <label class="cardinfo-label" for="cvv">CVV</label>
    <div class='input-wrapper' id="cvv"></div>
</div>
<div id="boton"></div>
```

En este ejemplo se utilizan elementos de fecha independientes, uno para mes ("expiration-month") y otro para año ("expiration-year").

Si se desea mostrar el mes y el año correspondientes a la fecha de caducidad en el mismo campo está disponible un elemento que incluye ambos valores en formato mm/aa.

Para ello reemplazaremos los elementos "*expiration-month*" y "*expiration-year*" por el elemento "*card-expiration*".

```
<div class="cardinfo-card-number">
    <label class="cardinfo-label" for="card-number">Número de tarjeta</label>
    <div class='input-wrapper' id="card-number"></div>
</div>

<div class="cardinfo-exp-date">
    <label class="cardinfo-label" for="card-expiration">Caducidad</label>
    <div class='input-wrapper' id="card-expiration"></div>
</div>

<div class="cardinfo-cvv">
    <label class="cardinfo-label" for="cvv">CVV</label>
    <div class='input-wrapper' id="cvv"></div>
</div>

<div id="boton"></div>
```

Se debe incluir una función de escucha de mensajes (*listener*) para recibir el ID de operación cuando éste se genere. Se debe utilizar la función *storeIdOper* con la siguiente definición:

```
storeIdOper(event, idToken, idErrorMsg, validationFunction);
```

en la que se deberá indicar el evento recogido por el listener (*event*), el ID del elemento del DOM se debe almacenar el ID de operación una vez sea generado (*idToken*), el identificador del elemento en el que se almacenarán los códigos de error en caso de que existan errores de validación de los datos (*idErrorMsg*). En el ejemplo posterior ambos se almacenan en un input de tipo "hidden".

Opcionalmente, se establece la posibilidad de ejecutar una función propia para realizar validaciones previas por parte del comercio. Únicamente se continuará con la generación del ID de operación si la función de validación ejecutada por el comercio retorna un valor *true*.

```
<input type="hidden" id="token" ></input>
<input type="hidden" id="errorCode" ></input>
<script>
function merchantValidationEjemplo(){
//Insertar validaciones...
return true;
}
```

```
<!-- Listener -->
window.addEventListener("message", function receiveMessage(event) {
    storeIdOper(event,"token", "errorCode", merchantValidationEjemplo);
});
</script>
```

Una vez preparado el envío de datos y la posterior recepción, se llamará a las funciones proporcionadas para generar los elementos de introducción de datos de tarjeta:

```
<!-- Petición de carga de iframes -->
getCardInput('card-number', estiloCaja, placeholder, estiloInput);
getExpirationMonthInput('expiration-month', estilosCSS, placeholder);
getExpirationYearInput('expiration-year', estilosCSS, placeholder);
getCVVInput('cvv', estilosCSS, placeholder);
getPayButton('boton', estilosCSS, 'Pagar con Redsys', fuc, terminal, merchantOrder);
```

Si se utiliza el elemento de fecha unificado (mm/aa) reemplazaremos las funciones getExpirationMonthInput() y getExpirationYearInput() por la función getExpirationInput().

```
// Petición de carga de iframes
getCardInput('card-number', estiloCaja, placeholder, estiloInput);
getExpirationInput('card-expiration', estilosCSS, placeholder);
getCVVInput('cvv', estilosCSS, placeholder);
getPayButton('boton', estilosCSS, 'Pagar con Redsys', fuc, terminal, merchantOrder);
```

Como parámetros de las funciones se indicará el id del contenedor reservado para su generación, el estilo requerido para el mismo (formato CSS) y el placeholder de dicho campo. En el caso de la función getCardInput() se pasarán dos campos de estilo CSS, uno con el estilo de la caja exterior y otro con el estilo del input que contiene. Adicionalmente, se podrá personalizar el texto del botón de pago y, por último, se deberá informar el valor del FUC, terminal y número de pedido (alfanumérico de entre 4 y 12 posiciones) en la petición de carga del iframe con el botón de pago.

**El merchantOrder utilizado en la generación del idOper deberá reutilizarse en la posterior petición de autorización.**

De esta forma, cuando el cliente introduzca sus datos de tarjeta en los elementos generados por Redsys y pulse el botón de pago, se generará y almacenará en el formulario del comercio un ID asociado a la operación para que éste formalice la compra sin necesidad de tratar datos de tarjeta.

## 5. Catálogo de errores

Cuando se pulse el botón generado por Redsys se lanzarán las validaciones a partir de los datos introducidos. Se podrá recibir el siguiente catálogo de errores.

Se incluye la descripción de los errores, pero es responsabilidad del comercio mostrarlos de la forma que considere adecuada.

msg1	Ha de llenar los datos de la tarjeta
msg2	La tarjeta es obligatoria
msg3	La tarjeta ha de ser numérica
msg4	La tarjeta no puede ser negativa
msg5	El mes de caducidad de la tarjeta es obligatorio
msg6	El mes de caducidad de la tarjeta ha de ser numérico
msg7	El mes de caducidad de la tarjeta es incorrecto
msg8	El año de caducidad de la tarjeta es obligatorio
msg9	El año de caducidad de la tarjeta ha de ser numérico
msg10	El año de caducidad de la tarjeta no puede ser negativo
msg11	El código de seguridad de la tarjeta no tiene la longitud correcta
msg12	El código de seguridad de la tarjeta ha de ser numérico
msg13	El código de seguridad de la tarjeta no puede ser negativo
msg14	El código de seguridad no es necesario para su tarjeta
msg15	La longitud de la tarjeta no es correcta
msg16	Debe Introducir un número de tarjeta válido (sin espacios ni guiones).
msg17	Validación incorrecta por parte del comercio
msg18	Error de inicialización de dominio

## 6. Catálogo de idiomas

IDIOMA	CÓDIGO	ISO 639-1
Español	1	ES
Inglés	2	EN
Catalán	3	CA
Francés	4	FR
Alemán	5	DE
Neerlandés	6	NL
Italiano	7	IT
Sueco	8	SV
Portugués	9	PT
Valenciano	10	VA
Polaco	11	PL
Gallego	12	GL
Euskera	13	EU
Búlgaro	100	BG
Chino	156	ZH
Croata	191	HR
Checo	203	CS
Danés	208	DA
Estonio	233	ET
Finlandés	246	FI
Griego	300	EL
Húngaro	348	HU
Indio	356	HI
Japonés	392	JA
Coreano	410	KO
Letón	428	LV
Lituano	440	LT
Maltés	470	MT
Rumano	642	RO
Ruso	643	RU

Árabe	682	AR
Eslovaco	703	SK
Esloveno	705	SL
Turco	792	TR

## 7. Envío de operación tras generación de ID de operación

Una vez recibido y almacenado el ID de operación por parte del comercio según se ha descrito en los apartados anteriores, podrá lanzar la operación de autorización utilizando cualquiera de los interfaces disponibles en el TPV Virtual.

**En la operación de autorización, se tendrá que enviar el parámetro DS\_MERCHANT\_IDOPER en lugar de los campos habituales de envío de datos de tarjeta. Además, se deberá utilizar el mismo número de pedido (DS\_MERCHANT\_ORDER) que el utilizado en la generación del idOper.**

Se pone a disposición de los comercios librerías que simplifican esta conexión en lenguajes Java y PHP. Su descarga está disponible en la sección de descargas de la web de desarrolladores de Redsys:

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

La descarga de las librerías incluye documentación de ayuda para su uso.

### 6.1 Implementación sin uso de las librerías de ayuda

Si no se desea utilizar las librerías de ayuda o se quiere implementar para otros lenguajes de programación, pueden implementar directamente la llamada REST al TPV Virtual. **Para obtener más información**, se recomienda consultar la documentación de Integración vía REST.

La solicitud de autorización se hace a través de una petición al TPV Virtual. En dicha petición deberá incluir 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 (Consultar Parámetros de entrada y salida).
- Ds\_Signature: Firma de los datos enviados. Es el resultado del HMAC SHA256 de la cadena JSON codificada en Base 64 enviada en el parámetro anterior

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	Entorno
<a href="https://sis-t.redsys.es:25443/sis/rest/trataPeticionREST">https://sis-t.redsys.es:25443/sis/rest/trataPeticionREST</a>	Pruebas
<a href="https://sis.redsys.es/sis/rest/trataPeticionREST">https://sis.redsys.es/sis/rest/trataPeticionREST</a>	Real

Una vez gestionada la consulta, el TPV Virtual informará al servidor del comercio el resultado de esta con la información del resultado incluida en un fichero JSON. En él se incluirán los siguientes campos:

- Ds\_SignatureVersion: Constante que indica la versión de firma que se está utilizando.
- Ds\_MerchantParameters: Cadena en formato JSON con todos los parámetros de la respuesta codificada en Base 64 y sin retornos de carro.
- Ds\_Signature: Firma de los datos enviados. Resultado del HMAC SHA256 de la cadena JSON codificada en Base 64 enviada en el parámetro anterior. **El comercio es responsable de validar el HMAC enviado por el TPV Virtual para asegurarse de la validez de la respuesta. Esta validación es necesaria para garantizar que los datos no han sido manipulados y que el origen es realmente el TPV Virtual.**

A continuación, se describen los datos que debe incluir el Ds\_MerchantParameters para enviar una petición de autenticación 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_IDOPER":" 455097a74c21b761be86acb26c32609dce222e66",  
}
```

Como respuesta se obtendrá:

```
{  
    "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"  
}
```

## **8. Identificación de operaciones inSite en el Portal de Administración del TPV Virtual**

En el portal de administración del TPV Virtual se puede identificar las operaciones realizadas con inSite consultando el campo “entrada” de la consulta de operaciones.

Las operaciones se registrarán con la entrada “inSite REST”

## **9. Autenticación 3DSecure en InSite**

Los comercios que utilicen la conexión inSite tienen la posibilidad de incluir el protocolo 3DSecure (3DS) para autenticar a los titulares y obtener un nivel adicional de protección ante fraude.

Incluir la autenticación 3DS implica redirigir la navegación del cliente hacia el servidor de autenticación del banco/entidad emisora de la tarjeta para que éste pueda solicitar las credenciales necesarias. Este paso debe realizarse en un paso posterior al de recoger los datos de tarjeta descrito en los apartados anteriores.

Para utilizar la autenticación 3DS, el terminal del TPV Virtual debe estar configurado por parte de la entidad financiera para soportar autenticación 3D Secure. Igualmente podría ser necesario que por configuración del TPV Virtual este paso no solo sea opcional, sino que sea requerido para la correcta autorización de las operaciones (si tiene dudas consulte la configuración 3DS a su entidad financiera proveedora del TPV Virtual).

### **8.1 3DSecure v1.0.2**

#### **8.1.1 Solicitar autorización**

La solicitud de autorización se hace a través de una petición al TPV Virtual. En dicha petición deberá incluir 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 (Consultar Parámetros de entrada y salida).
- Ds\_Signature: Firma de los datos enviados. Es el resultado del HMAC SHA256 de la cadena JSON codificada en Base 64 enviada en el parámetro anterior

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	Entorno
<a href="https://sis-t.redsys.es:25443/sis/rest/trataPeticionREST">https://sis-t.redsys.es:25443/sis/rest/trataPeticionREST</a>	Pruebas
<a href="https://sis.redsys.es/sis/rest/trataPeticionREST">https://sis.redsys.es/sis/rest/trataPeticionREST</a>	Real

Una vez gestionada la consulta, el TPV Virtual informará al servidor del comercio el resultado de esta, con la información del resultado incluida en un fichero JSON. En él se incluirán los siguientes campos:

- Ds\_SignatureVersion: Constante que indica la versión de firma que se está utilizando.
- Ds\_MerchantParameters: Cadena en formato JSON con todos los parámetros de la respuesta codificada en Base 64 y sin retornos de carro.
- Ds\_Signature: Firma de los datos enviados. Resultado del HMAC SHA256 de la cadena JSON codificada en Base 64 enviada en el parámetro anterior. **El comercio es responsable de validar el HMAC enviado por el TPV Virtual para asegurarse de la validez de la respuesta. Esta validación es necesaria para garantizar que los datos no han sido manipulados y que el origen es realmente el TPV Virtual.**

A continuación, se describen los datos que debe incluir el Ds\_MerchantParameters para enviar una petición de autenticación al Servicio REST:

```
{
    "DS_MERCHANT_ORDER":1552642885,
    "DS_MERCHANT_MERCHANTCODE": "999008881",
    "DS_MERCHANT_TERMINAL": "2",
    "DS_MERCHANT_CURRENCY": "978",
    "DS_MERCHANT_TRANSACTIONTYPE": "0",
    "DS_MERCHANT_AMOUNT": "1000",
    "DS_MERCHANT_IDOPER": "455097a74c21b761be86acb26c32609dce222e66",
    "DS_MERCHANT_EMV3DS": {
        "threeDSInfo": "AuthenticationData",
        "protocolVersion": "1.0.2",
        "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"
    }
}
```

Como respuesta se obtendrá lo siguiente:

```
{
  "Ds_Order":"1552642885",
  "Ds_MerchantCode":"999008881",
  "Ds_Terminal":"2",
  "Ds_Currency":"978",
  "Ds_Amount":"1000",
  "Ds_TransactionType":"0",
  "Ds_EMV3DS": {
    "threeDSInfo":"ChallengeRequest",
    "protocolVersion":"1.0.2",
    "acsURL": "https://sis.redsys.es/sis-simulador-web/authenticationRequest.jsp",
    "PAREq":"eJxVUttygjAQ/RWG95KEooKzpkPVjj7QOpZ+QBp2KIYuDVxDx77tRqS0zmdmzJ+zlnMBDXxycbzRNXpUzV3jcdbDUVZaXHzP3LX26C90HCenOIC5eUXcGJSTYNOoDnTybuU1Rq7zPsFGIFmlU+mOfi4j7vrAfn3DOW9HYlbCjt/gI4dpKUifPBzZAqmnn0TpWtBKW/HtfPMhgFYSiAXSEuNYL+brcJstNnCy381X8nAK7pKFUBcp5RUjnlZOhU5sO35XzzFSXlbAzD7rqytac5MQPgA0AOOnOQu7atp4wdj0fPYNacGk9XBTBLAbvNtuls1FcPp9ks0/7lZQ+JftIn6R09p88WcRHOjNg9gZkqkU5KOIIPhVi6kfAznlQhZ1BintPcNr0gqC2TeKBsszfDJAHHiw6yWg50hYDAuzrjkS6QbL+xkDOaEY73Cafr6zGuiXZ/loloIEYWLnPjK2WkzhBJC7LABm/2VXJ9n1GVD073n8AOa7wW0=",
    "MD":"cd164a6d0b77c96f7ef476121acfa987a0edf602"
  }
}
```

### **8.1.2 Ejecución de la autenticación**

El comercio deberá montar un formulario que envíe un POST a la URL del parámetro acsURL obtenido en la respuesta de la petición de autorización anterior. Dicho formulario envía 3 parámetros necesarios para la autenticación:

- *PaReq*, cuyo valor se obtiene del parámetro PAREq obtenido en la respuesta de la petición de autorización anterior.
- *MD*, cuyo valor se obtiene del parámetro MD obtenido en la respuesta de la petición de autorización anterior.
- *TermUrl*, que identifica la URL a la que entidad Emisora hará un POST con el resultado de autenticación. Dicho formulario enviará un único parámetro *PAREs*, que contiene el resultado de la autenticación y que deberá ser recogido por el comercio para su posterior envío en la petición de confirmación de autorización.

### 8.1.3 Confirmación de autorización 3DSecure 1.0 posterior al Challenge

A continuación, se describen los datos que debe incluir el Ds\_MerchantParameters para enviar una petición de confirmación de autorización 3DSecure 1.0 al Servicio REST:

```
{
  "DS_MERCHANT_ORDER":1552642885,
  "DS_MERCHANT_MERCHANTCODE":"999008881",
  "DS_MERCHANT_TERMINAL":"2",
  "DS_MERCHANT_CURRENCY":"978",
  "DS_MERCHANT_TRANSACTIONTYPE":"0",
  "DS_MERCHANT_AMOUNT":"1000",
  "DS_MERCHANT_EMV3DS":{
    "threeDSInfo":"ChallengeResponse",
    "protocolVersion":"1.0.2",
    "PARes":"eJzFWNmSo0iyfecrymoeNVVsWqBNmWPBKlaJVcAbmwBJLALE9vWDIJVZWT3VNn3vw70yyRR4uDvu
ESeOu8X2XON+/dLFdZOvxctX9Dvy9UtcpGWUFcnLV8vkvhFf//W6NdM6jhkjDu91/LpV4qbxx/hL ....",
    "MD":"035535127d549298f11d7d2fc1b0d4e9300f93f1"
  }
}
```

Como respuesta se obtendrá el resultado final de la operación:

```
{
  "Ds_Amount":"1000",
  "Ds_Currency":"978",
  "Ds_Order":"1552642885",
  "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
}
```

## 8.2 EMV3DS

### 8.2.1 Iniciar Petición

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

El inicio petición se hace a través de una petición REST al TPV Virtual En dicha petición deberá incluir 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 (Consultar Parámetros de entrada y salida).
- Ds\_Signature: Firma de los datos enviados. Es el resultado del HMAC SHA256 de la cadena JSON codificada en Base 64 enviada en el parámetro anterior

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	Entorno
<a href="https://sis-t.redsys.es:25443/sis/rest/iniciaPeticionREST">https://sis-t.redsys.es:25443/sis/rest/iniciaPeticionREST</a>	Pruebas
<a href="https://sis.redsys.es/sis/rest/iniciaPeticionREST">https://sis.redsys.es/sis/rest/iniciaPeticionREST</a>	Real

Una vez gestionada la petición, el TPV Virtual informará al servidor del comercio el resultado de esta, con la información del resultado incluida en un fichero JSON. En él se incluirán los siguientes campos:

- Ds\_SignatureVersion: Constante que indica la versión de firma que se está utilizando.
- Ds\_MerchantParameters: Cadena en formato JSON con todos los parámetros de la respuesta codificada en Base 64 y sin retornos de carro.
- Ds\_Signature: Firma de los datos enviados. Resultado del HMAC SHA256 de la cadena JSON codificada en Base 64 enviada en el parámetro anterior. **El comercio es responsable de validar el HMAC enviado por el TPV Virtual para asegurarse de la validez de la respuesta. Esta validación es necesaria para garantizar que los datos no han sido manipulados y que el origen es realmente el TPV Virtual.**

A continuación, se describen los datos que debe incluir el Ds\_MerchantParameters para enviar un inicio petición al Servicio REST:

```
{  
    "DS_MERCHANT_ORDER":"1552571678",  
    "DS_MERCHANT_MERCHANTCODE":"999008881",  
    "DS_MERCHANT_TERMINAL":"2",  
    "DS_MERCHANT_CURRENCY":"978",  
    "DS_MERCHANT_TRANSACTIONTYPE":"0",  
    "DS_MERCHANT_AMOUNT":"1000",  
    "DS_MERCHANT_IDOPER":"455097a74c21b761be86acb26c32609dce222e66",  
    "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"  
    }  
}
```

El parámetro **Ds\_EMV3DS** 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.

### 8.2.2 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 (iniciaPeticion) se recibe los datos siguientes para ejecutar el 3DSMethod:
  - a. threeDSMethodURL: url del 3DSMethod
  - b. threeDSServerTransID: Identificador de transacción EMV3DS

Si en la respuesta no se recibe threeDSMethodURL el proceso finaliza. En la autorización enviar threeDSCompInd = N
2. Construir el JSON Object con los parámetros:
  - a. threeDSServerTransID: 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
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**

### 8.2.3 Petición de autorización con datos EMV3DS

La petición de autorización se hace a través de una petición REST al TPV Virtual En dicha petición deberá incluir 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 (Consultar Parámetros de entrada y salida).
- Ds\_Signature: Firma de los datos enviados. Es el resultado del HMAC SHA256 de la cadena JSON codificada en Base 64 enviada en el parámetro anterior

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	Entorno
<a href="https://sis-t.redsys.es:25443/sis/rest/trataPeticionREST">https://sis-t.redsys.es:25443/sis/rest/trataPeticionREST</a>	Pruebas
<a href="https://sis.redsys.es/sis/rest/trataPeticionREST">https://sis.redsys.es/sis/rest/trataPeticionREST</a>	Real

Una vez gestionada la consulta, el TPV Virtual informará al servidor del comercio el resultado de la misma con la información del resultado incluida en un fichero JSON. En él se incluirán los siguientes campos:

- Ds\_SignatureVersion: Constante que indica la versión de firma que se está utilizando.
- Ds\_MerchantParameters: Cadena en formato JSON con todos los parámetros de la respuesta codificada en Base 64 y sin retornos de carro.
- Ds\_Signature: Firma de los datos enviados. Resultado del HMAC SHA256 de la cadena JSON codificada en Base 64 enviada en el parámetro anterior. **El comercio es responsable de validar el HMAC enviado por el TPV Virtual para asegurarse de la validez de la respuesta. Esta validación es necesaria para garantizar que los datos no han sido manipulados y que el origen es realmente el TPV Virtual.**

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_IDOPER":" 455097a74c21b761be86acb26c32609dce222e66",  
    "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":"false",  
        "browserLanguage":"ES-es",  
        "browserColorDepth":"24",  
        "browserScreenHeight":"1250",  
        "browserScreenWidth":"1320",  
        "browserTZ":"52",  
        "threeDSServerTransID":"8de84430-3336-4ff4-b18d-f073b546cce",  
        "notificationURL":"https://comercio-inventado.es/recibe-respuesta-autenticacion",  
        "threeDSComplInd":"Y"  
    }  
}
```

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":"1",  
    "Ds_Language":"1",  
    "Ds_CardNumber":"454881*****0004",  
    "Ds_Card_Type":"C",  
    "Ds_MerchantData": "",  
    "Ds_Card_Country":"724",  
    "Ds_Card_Brand":1  
}
```

- Si no es así, se solicitará la ejecución de un 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":"eyJ0aHJIZURTU2VYdmVyVHJhbnNJRCI6ljhkZTg0NDMwLTMzMzYtNGZmNC1iMThkLWYwNzNiNTQ2Y2NIYSlsImFjc1RyYW5zSUQiOjKjYjOTljNC1hMmZkLTQ3ZWUtOTI2Zi1mYTBiMDk0MzUyYTAiLCJtZXNzYWdlVHlwZSI6IkNSZXEiLCJtZXNzYWdlVmVyc2lvbil6ljluMS4wliwiY2hhbGxlbmdlV2luZG93U2l6ZSI6ljA1In0"
    }
}
```

## 8.2.4 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": "eyJ0aHJIZURTU2VYdmVyVHJhbnNJRCI6lmU50WMzYzI2LTFiZWltNGY4NS05ZmE3LTi3OTJiZjE5NDZIMlsImFjc1RyYW5zSUQiOilyMTQzNDFhYi0wMjhLTrmMGEtOTEyNi1iMDFkYmE5OTc2MTkiLCJtZXNzYWdlVHlwZSI6IkNSZXEiLCJtZXNzYWdlVmVyc2lvbil6ljluMS4wliwiY2hhbGxlbmdlV2luZG93U2l6ZSI6ljA1In0"}
```

### 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="eyJ0aHJIZURTU2VYdmVyVHJhbnNJRCI6lmU50WMzYzI2LTFiZWltNGY4NS05ZmE3LTi3OTJiZjE5NDZIMlsImFjc1RyYW5zSUQiOilyMTQzNDFhYi0wMjhLTrmMGEtOTEyNi1iMDFkYmE5OTc2MTkiLCJtZXNzYWdlVHlwZSI6IkNSZXEiLCJtZXNzYWdlVmVyc2lvbil6ljluMS4wliwiY2hhbGxlbmdlV2luZG93U2l6ZSI6ljA1In0">
</form>
```

### 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.

### 8.2.5 Confirmación de autorización EMV3DS posterior al Challenge

A continuación, se describen los datos de 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_CVV2I":"XXX",  
    "DS_MERCHANT_EMV3DS":{  
        "threeDSInfo":"ChallengeResponse",  
        "protocolVersion":"2.1.0",  
        "cres":"eyJ0aHJIZURTU2VydmVyVHJhbNJRCl6jhkZTg0NDMwLTMzMzYtNGZmNC1iMThkLW  
YwNzNiNTQ2Y2NIYSIsImFjc1RyYW5zSUQiOjKjVjOTljNC1hMmZkLTQ3ZWUtOTI2Zi1mYTBiM  
Dk0MzUyYTAiLCJtZXNzYWdlVHlwZSI6IkNSZXMiLCJtZXNzYWdlVmVyc2lvbil6ljluMS4wliwidHJ  
hbnNTdGF0dXMiOjZIn0="  
    }  
}
```

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: Para mayor detalle sobre el protocolo EMV3DS 2.0, se deben consultar la guía de Integración vía REST del TPV-Virtual.**