

Manual de Integración



Versión 2.18
Agosto de 2020

Contenido

1. Métodos de integración.	4
2. Integración mediante método POST.	4
2.1 Introducción	4
2.2 Ambientes	4
2.3 Seguridad	4
2.3.1 ASP NET FRAMEWORK	4
Compatibilidad	4
Uso	5
2.3.2 ASP NET CORE5	6
Uso	6
2.3.3 Java - Android	7
Uso	7
2.3.4 iOS	10
Uso	11
2.3.5 PHP COMPOSER	13
composer require sha256encrypt/aes-encrypter @dev	13
Ejemplo de Integración	15
3.3 Formulario POST	16
3.4 Proceso de pago en Macro Click de Pago	18
3.5 Cancelación de un pago en Macro Click de Pago	20
3.6 Expiración de transacciones en Macro Click de Pago	20
4. Notificaciones	21
5. Devoluciones	25
6. Tarjetero	26
Almacén de tarjetas	26
Pago con tarjeta almacenada	27
7. Entorno de Sandbox	29

Historial de Revisiones

Versión	Fecha	Persona	Comentarios
1.0	26-03-2018	LM	Versión Inicial
2.0	23-07-2018	FB	Versión con implementación de seguridad con hash
2.1	23-08-2018	AG	URL ambiente de sandbox
2.2	29-10-2018	LM	Versión con nueva implementación de seguridad
2.3	08-01-2019	LM	URL ambiente de producción y modificaciones POST al comercio.
2.4	14-01-2019	AC	Documentación de Java actualizada
2.5	24-01-2019	AC	Documentación Java/Android y iOS
2.6	04-07-2019	LM	Campos adicionales información y datos del cliente
2.7	22-07-2019	LM	Documentación PHP
2.9	22-01-2020	LM	Estados de la transacción
2.13	19-03-2020	LM	Modificaciones librería PHP en la obtención de la dirección IP pública.
2.15	19-05-2020	LM	Pago con medio de pago DEBIN

1. Métodos de integración.

Macro Click de Pago cuenta con integración mediante método POST.

2. Integración mediante método POST.

2.1 Introducción

En este método el comercio, al momento del pago (checkout), direcciona al comprador al entorno de **Macro Click de Pago**, en el cual se ingresará la información relacionada al medio de pago seleccionado por el comprador de aquellos habilitados para su comercio. Una vez finalizada la transacción, se redireccionará al comprador a la página especificada por el comercio en el formulario de pago o a la página por defecto especificada por el comercio al darse de alta en **Macro Click de Pago**.

2.2 Ambientes

Para realizar pruebas y validar la correcta integración con el Botón de Pago, el envío de los datos de la compra por parte del comercio mediante una solicitud **POST**, deberá realizarse sobre **HTTPS** a la siguiente URL correspondiente al ambiente de **Sandbox**:

- <https://sandboxpp.asjservicios.com.ar/>

Una vez finalizadas las pruebas y estando completa la integración con el Botón de Pago, el envío de los datos de la compra por parte del comercio deberá realizarse mediante una solicitud **POST**, sobre **HTTPS** a la siguiente URL correspondiente al ambiente **Producción**:

- <https://botonpp.asjservicios.com.ar/>

Tener en cuenta que los datos suministrados mediante email por **Macro Click de Pago** al Comercio, es decir, el identificador de comercio (GUID) y la SecretKey van a variar según el ambiente en el que se realice la integración con el Botón de Pago.

2.3 Seguridad

Para una comunicación más segura con los servicios ofrecidos por **Macro Click de Pago**, el comercio que se integre deberá incluir en el REQUEST, un hash codificado en SHA256 y encriptar ciertos datos en AES-256-CBC. **Macro Click de Pago** provee el servicio de obtención de dicho hash y el método para la encriptación de los campos, que dependerá del lenguaje en que se encuentre el desarrollo a integrar.

2.3.1 ASP NET FRAMEWORK

Compatibilidad

La siguiente librería es compatible a partir de la versión 3.5 hasta la versión 4.6.1 de .NET Framework.

Agregar la siguiente librería a través de la consola de administración de paquetes de Nuget:

```
PM > Install-Package PlusPagosSecurityKitNetFramework -Version 2.3.1
```

Para utilizarla deberá agregar el siguiente using:

```
using SHA256EncryptNetFramework;
```

Uso

Generar hash

Obtener la dirección IP y pasarla como parámetro al método **Generate** de la librería. El siguiente es un ejemplo de obtención de la dirección IP.

```
string ip = System.Web.HttpContext.Current.Request.ServerVariables["HTTP_X_FORWARDED_FOR"];  
if (string.IsNullOrEmpty(ip))  
{  
    ip = System.Web.HttpContext.Current.Request.ServerVariables["REMOTE_ADDR"];  
}
```

Instanciar la clase **SHA256Hash** e invocar el método **Generate**. El mismo recibe como parámetros la dirección IP obtenida con anterioridad, el GUID del comercio, el monto de la transacción (sin comas ni puntos y considerando los dos últimos dígitos como decimales. Ejemplo: 2.00 → 200), y el identificador de la sucursal:

```
var hash = SHA256Hash.Generate("ip", "SecretKey", "GUIDComercio", "SucursalComercio", "Monto")
```

Aclaración: en caso de no querer especificar una sucursal, enviar vacío el campo **SucursalComercio**

Encriptar datos del POST

Se deberán encriptar los siguientes datos del POST:

- CallbackSuccess
- CallbackCancel
- Monto
- SucursalComercio (en caso de no pasar el dato de la sucursal, encriptar una cadena vacía)
- Informacion (opcional – sólo si se envía información, encriptar)
- CallbackPending (opcional para uso de medio de pago DEBIN – sólo si se envía la url, encriptar)

Instanciar la clase **AESDecrypter** e invocar el método **EncryptString ()**. Este método recibe como parámetro un string y una palabra secreta (en este caso la palabra secreta será la **SecretKey** del comercio).

```
var secretkey = "prueba_7f3a9253-c83a-4d68-b151-6e2fb1e10463"; //secretkey provista por PP  
  
var monto = AESDecrypter.EncryptString(montotransaccion, secretkey);  
var callbackSuccess = AESDecrypter.EncryptString(CallbackSuccess, secretkey);  
var callbackCancel = AESDecrypter.EncryptString(CallbackCancel, secretkey);  
var sucursalComercio = AESDecrypter.EncryptString(SucursalComercio, secretkey);
```

2.3.2 ASP NET CORE

Agregar la siguiente librería a través de la consola de administración de paquetes de Nuget:

```
PM > Install-Package PlusPagosSecurityKitCore -Version 2.1.0
```

Para utilizarla deberá agregar el siguiente using:

```
using SHA256EncryptCore;
```

Uso

Instanciar la clase **SHA256Hash** e invocar el método **Calculate**.

Generar hash

Obtener la dirección IP y pasarla como parámetro al método **Calculate** de la librería. El siguiente es un ejemplo de obtención de la dirección IP.

- Inyectar la dependencia **IHttpContextAccessor** en el Controller dónde se va a instanciar la librería.

```
public class HomeController : Controller
{
    public IHttpContextAccessor _accessor;
    public HomeController (IHttpContextAccessor httpAccessor)
    {
        _accessor = httpAccessor;
    }
}
```

- Obtener la dirección IP

```
var ip = _accessor.HttpContext.Connection.RemoteIpAddress.ToString();
```

- Pasar la dirección IP obtenida como parámetro del método **Calculate** de la librería, además de los parámetros: **SecretKey**, **GUID** del comercio, **identificador de la sucursal** y **monto de la transacción** (sin comas ni puntos y considerando los dos últimos dígitos como decimales. Ejemplo: 2.00 → 200).

```
var hash = SHA256Hash.Generate(ip,"SecretKey", "GUIDComercio", "SucursalComercio", "Monto")
```

Aclaración: en caso de no querer especificar una sucursal, enviar vacío el campo SucursalComercio

Encriptar datos del POST

Se deberán encriptar los siguientes datos del POST:

- CallbackSuccess
- CallbackCancel
- Monto
- SucursalComercio (en caso de no pasar el dato de la sucursal, encriptar una cadena vacía)
- Informacion (opcional – sólo si se envía información, encriptar)
- CallbackPending (opcional para uso de medio de pago DEBIN – sólo si se envía la url, encriptar)

Instanciar la clase `AESCrypter` e invocar el método `EncryptString ()`. Este método recibe como parámetro un string y una palabra secreta (en este caso la palabra secreta será la `SecretKey` del comercio).

```
var secretkey = "prueba_7f3a9253-c83a-4d68-b151-6e2fb1e10463"; //secretkey provista por PP
var monto = AESCrypter.EncryptString(montotransaccion, secretkey);
var callbackSuccess = AESCrypter.EncryptString(CallbackSuccess, secretkey);
var callbackCancel = AESCrypter.EncryptString(CallbackCancel, secretkey);
var sucursalComercio = AESCrypter.EncryptString(SucursalComercio, secretkey);
```

2.3.3 Java - Android

En un proyecto Maven, incluir lo siguiente en el archivo `build.gradle`:

```
dependencies {
    implementation 'com.pluspagos:firm:0.7.4'
}
```

En un proyecto Maven, incluir lo siguiente en el archivo `pom.xml`:

```
<project>
  ...
  <dependencies>
    <dependency>
      <groupId>com.pluspagos</groupId>
      <artifactId>firm</artifactId>
      <version>0.7.4</version>
    </dependency>
  </dependencies>
  ...
  <repositories>
    <repository>
      <id>central</id>
      <name>bintray</name>
      <url>https://jcenter.bintray.com</url>
    </repository>
  </repositories>
  ...
</project>
```

Uso

Construir un objeto `FormularioModel` especificando los datos requeridos por Macro Click de Pago para el pago:

```
String secretKey = "secret";
String guidComercio = "000000-0000-0000-0000000000";
String ip = ... // obtener la IP pública del cliente, ver ejemplo abajo

FormularioModel model = new FormularioModel.Builder(secretKey)
    .ip(ip)
    .comercio(guidComercio)
    .callbackSuccess("http://www.google.com")
    .callbackCancel("http://www.bing.com")
    .transaccionId(...) //generar un id unívoco para la transacción
    .sucursal("1") // no es obligatorio, por defecto envía ""
    .monto(120.47) // valor double del precio real
    .addProducto("Producto 1") // llamar para cada producto
    .addProducto("Producto 2")
    .build();
```

Luego obtener el body para enviar en el POST a Macro Click de Pago:

```
byte[] body = model.toUrlEncodedForm();
```

Ejemplo Android

En Android, utilizando el componente **`android.webkit.WebView`**, se envía el POST y se muestra el formulario de pago:

```
String url = "https://sandboxpp.asjservicios.com.ar/"; // ambiente
WebView webView = findViewById(R.id.webview);
webView.getSettings().setJavaScriptEnabled(true);
webView.postUrl(url, body); // body obtenido de FormularioModel
```

Obtener IP

Es necesario incluir en el post la IP pública desde la cual se hará la conexión con Macro Click de Pago. Para obtenerla es posible utilizar un servicio como `checkip.amazonaws.com`:

Con `java.net.URL`:

```
URL url = new URL("http://checkip.amazonaws.com");
BufferedReader reader = new BufferedReader(new InputStreamReader(
    url.openStream()));
String ip = reader.readLine().trim();
reader.close();
```


Con OkHttp:

```
OkHttpClient client = new OkHttpClient.Builder().build();
client.newCall(new Request.Builder()
    .url("http://checkip.amazonaws.com")
    .get().build())
    .enqueue(new Callback() {
        public void onFailure(Call call, IOException e) {e.printStackTrace();}

        public void onResponse(Call call, Response response) throws
IOException {
            String ip = response.body().string().trim();
        }
    });
```

La clase FormularioModel realiza todas las tareas necesarias para obtener el POST a enviar a Macro Click de Pago. Pero si desea generar el hash o encriptar los campos manualmente, puede utilizar los siguientes métodos.

Generar hash

```
new SHA256Firm().getFirm(ipClient, secretKey, guidComercio, sucursalId, monto)
```

Parámetros:

- **guidComercio:** proporcionado por Macro Click de Pago
- **sucursalId:** proporcionado por Macro Click de Pago
- **secretKey:** proporcionado por Macro Click de Pago
- **monto:** el monto total del pago que se realizará, con los decimales incluidos sin separador decimal. Por ejemplo: \$156,34 se envía como "15634".
- **ipClient:** es la IP que solicita el pago.

El método devuelve como resultado, un valor String que será solicitado por el Post Request de Macro Click de Pago bajo el nombre Hash.

Aclaración: en caso de no querer especificar una sucursal, enviar vacío el campo sucursalId

Encriptar datos del POST

Se deberán encriptar los siguientes datos del POST:

- CallbackSuccess
- CallbackCancel
- Monto
- SucursalComercio (en caso de no pasar el dato de la sucursal, encriptar una cadena vacía)
- Informacion (opcional – sólo si se envía información, encriptar)
- CallbackPending (opcional para uso de medio de pago DEBIN – sólo si se envía la url, encriptar)

Invocar el método estático `AESCrypter.encryptString()`. Este método recibe como parámetro un String y una palabra secreta (en este caso la palabra secreta será la `SecretKey` del comercio).

```
String secretKey = "prueba_7f3a9253-c83a-4d68-b151-6e2fb1e10463"; //secretkey provista por PP
String monto = AESCrypter.encryptString(montoTransaccion, secretKey);
String callbackSuccess = AESCrypter.encryptString(callbackSuccessParam, secretKey);
String callbackCancel = AESCrypter.encryptString(callbackCancelParam, secretKey);
String sucursalComercio = AESCrypter.encryptString(sucursalComercioParam, secretKey);
```

2.3.4 iOS

Utilizando CocoaPods, agregar al Podfile:

```
target 'MiApp' do
  ...
  pod 'PlusPagosFirm'
end
```

Y luego ejecutar `pod install` dentro del directorio del proyecto.

Uso

Instanciar e inicializar un objeto `FormularioModel` especificando los datos requeridos por Macro Click de Pago para el pago:

```
import PlusPagosFirm
...
let secretKey = "secret"
let guidComercio = "000000-0000-0000-00000000"
let ip = ... // obtener la IP pública del cliente, ver ejemplo abajo

let model = FormularioModel(secretKey: secretKey)
    .ip(ip)
    .comercio(guidComercio)
    .callbackSuccess("http://www.google.com")
    .callbackCancel("http://www.bing.com")
    .transaccionComercioId(...) //generar un id unívoco para la transacción
    .sucursalComercio("1") // no es obligatorio, por defecto envía ""
    .monto(120.47) // valor Double del precio real
    .addProducto("Producto 1") // llamar para cada producto
    .addProducto("Producto 2")
```

Luego obtener el objeto Data para enviar en el POST a Macro Click de Pago:

```
let data: Data? = model.toUrlEncodedForm()
```

Ejemplo Swift

En iOS, utilizando el componente WKWebView, se envía el POST y se muestra el formulario de pago:

```
@IBOutlet weak var webView: WKWebView!

if let data: Data? = model.toUrlEncodedForm() {
    var req = URLRequest(url:
        URL(string: "https://sandboxpp.asjservicios.com.ar/")!)
    req.httpBody = data
    req.httpMethod = "POST"
    req.setValue("application/x-www-form-urlencoded",
        forHTTPHeaderField: "Content-Type")

    webView.load(req)
}
```

Obtener IP

Es necesario incluir en el post la IP pública desde la cual se hará la conexión con Macro Click de Pago. Para obtenerla es posible utilizar un servicio como checkip.amazonaws.com:

Con Alamofire:

```
Alamofire.request("https://checkip.amazonaws.com")
    .responseString { (response) in
        switch (response.result) {
            case .success(_):
                if let ip = response.value {
                    // utilizar ip
                }
            case .failure(let error):
                print("Error buscando IP", error.localizedDescription)
            }
        }
    }
```

La clase FormularioModel realiza todas las tareas necesarias para obtener el POST a enviar a Macro Click de Pago. Pero si desea generar el hash o encriptar los campos manualmente, puede utilizar los siguientes métodos.

Generar hash

```
SHA256Firm.getFirm(ipClient: ip, secretKey: secretKey, guidComercio: comercio,  
sucursalId: sucursalComercio, monto: monto)
```

Parámetros:

- **guidComercio:** proporcionado por Macro Click de Pago
- **sucursalId:** proporcionado por Macro Click de Pago
- **secretKey:** proporcionado por Macro Click de Pago
- **monto:** el monto total del pago que se realizará, con los decimales incluidos sin separador decimal. Por ejemplo: \$156,34 se envía como “15634”.
- **ipClient:** es la ip que solicita el pago.

El método devuelve como resultado, un valor String que será solicitado por el Post Request de Macro Click de Pago bajo el nombre Hash.

Aclaración: en caso de no querer especificar una sucursal, enviar vacío el campo sucursalId

Encriptar datos del POST

Se deberán encriptar los siguientes datos del POST:

- CallbackSuccess
- CallbackCancel
- Monto
- SucursalComercio (en caso de no pasar el dato de la sucursal, encriptar una cadena vacía)
- Informacion (opcional – sólo si se envía información, encriptar)
- CallbackPending (opcional para uso de medio de pago DEBIN – sólo si se envía la url, encriptar)

Invocar el método estático **AESCrypter.encryptString()**. Este método recibe como parámetro un String y una palabra secreta (en este caso la palabra secreta será la SecretKey del comercio).

```
let secretKey = "prueba_7f3a9253-c83a-4d68-b151-6e2fb1e10463"; //secretkey provista por PP  
let monto = AESCrypter.encryptString(montoTransaccion, secretKey);  
let callbackSuccess = AESCrypter.encryptString(callbackSuccessParam, secretKey);  
let callbackCancel = AESCrypter.encryptString(callbackCancelParam, secretKey);  
let sucursalComercio = AESCrypter.encryptString(sucursalComercioParam, secretKey);
```

2.3.5 PHP COMPOSER

Para poder utilizar la librería, deberá agregarla como dependencia en su proyecto PHP. Será necesario tener **composer** instalado en su sistema operativo (<https://getcomposer.org/>).

Para agregar la dependencia, desde una consola o terminal deberá posicionarse en la carpeta raíz de su proyecto y ejecutar lo siguiente:

```
composer require sha256encrypt/aes-encrypter @dev
```

Uso

Instanciar el paquete tal como se muestra a continuación:

```
require_once __DIR__ . '/vendor/autoload.php';  
use PlusPagos\SHA256Encrypt;  
use PlusPagos\AESEncrypter;
```

Generar hash

Obtener la **IP pública** desde la cual se hará la conexión con Macro Click de Pago. A modo de ejemplo:

```
private function getRealIpAddr()  
{  
    if (!empty($_SERVER['HTTP_CLIENT_IP']))  
        return $_SERVER['HTTP_CLIENT_IP'];  
  
    if (!empty($_SERVER['HTTP_X_FORWARDED_FOR']))  
        return $_SERVER['HTTP_X_FORWARDED_FOR'];  
  
    return $_SERVER['REMOTE_ADDR'];  
}
```

Importante: Cuando se está realizando pruebas de manera local, asegurarse que la IP enviada es la IP pública y no la de localhost.

Instanciar la clase **SHA256Encrypt ()** e invocar al método **Generate ()**. El mismo recibe como parámetros la dirección IP obtenida con anterioridad, el monto de la transacción (sin comas ni puntos y considerando los dos últimos dígitos como decimales. Ej: 2.00 → 200), la **secretKey** (provista por Macro Click de Pago) y el identificador de la sucursal. El resultado de la invocación deberá ser enviado en el POST en el campo Hash.

```
$hash = new SHA256Encrypt();  
$ipAddress = "179.65.262.7";  
$secretKey = "Test_1338e92c-ae90-4573-95b9-51f56865d794";  
$comercio = "6103dc82-6a9e-4b41-a206-a83ab3887e32";  
$sucursal = ""; //enviar vacío si no se tiene una sucursal configurada en PlusPagos  
$amount = "1200";  
  
echo $hash->Generate($ipAddress, $secretKey , $comercio , $sucursal , $amount);
```

Encriptar datos del POST

Se deberán encriptar los siguientes datos del POST:

- CallbackSuccess
- CallbackCancel
- Monto
- SucursalComercio (en caso de no pasar el dato de la sucursal, encriptar una cadena vacía)
- Informacion (opcional – sólo si se envía información, encriptar)
- CallbackPending (opcional para uso de medio de pago DEBIN – sólo si se envía la url, encriptar)

Instanciar la clase **AESDecrypter ()** e invocar al método **EncryptString ()**. Este método recibe como parámetro el string a encriptar y una palabra secreta (en este caso la palabra secreta será la **SecretKey** del comercio provista por Macro Click de Pago).

```
$aes = new AESDecrypter();  
  
$callbackSuccess= "http://www.test.com.ar/success";  
$callbackCancel= "http://www.test.com.ar/cancel";  
$monto = "1200";  
$sucursalComercio = "";  
$secretKey = "Test_1338e92c-ae90-4573-95b9-51f56865d794";  
  
echo $callbackEncriptada = $aes -> EncryptString($callbackSuccess, $secretKey);  
echo $cancelEncriptada = $aes -> EncryptString($callbackCancel, $secretKey);  
echo $montoEncriptado = $aes -> EncryptString($monto, $secretKey);  
echo $sucursalEncriptada = $aes -> EncryptString($sucursalComercio, $secretKey);
```

Ejemplo de integración

```
<?php

require('vendor/autoload.php');
use PlusPagos\SHA256Encrypt;
use PlusPagos\AESEncrypter;

$monto = 100; //monto total del pago
$transaccion_item = "TEST";
$transaccion_priceitem = "100"; //precio del item
$transaccion_numero = 1225; //Valor variable por transacción

function arreIp()
{
    if (!empty($_SERVER['HTTP_CLIENT_IP']))
        return $_SERVER['HTTP_CLIENT_IP'];
    if (!empty($_SERVER['HTTP_X_FORWARDED_FOR']))
        return $_SERVER['HTTP_X_FORWARDED_FOR'];
    return $_SERVER['REMOTE_ADDR'];
}

$hash = new SHA256Encrypt();
$ipAddress = arreIp(); //PARA PRUEBAS LOCALHOST ESPECIFICAR IP PÚBLICA.
$secretKey = "SECRETKEY_7ce56a47-6718-41a4-8677-52fe438d4f7b"; //secretkey
$comercio = "44431f82-0785-4414-9890-f95f566e5bbb"; //guid
$sucursal = ""; //enviar vacío si no se tiene una sucursal configurada en PlusPagos
$amount = trim($monto);

$hashfinal = $hash->Generate($ipAddress, $secretKey , $comercio , $sucursal , $amount);

$aes = new AESEncrypter();

$callbackSuccess= "http://urlsuccess";
$callbackCancel= "http://urlCancel"
$callbackEncriptada = $aes -> EncryptString($callbackSuccess, $secretKey);
$cancelEncriptada = $aes -> EncryptString($callbackCancel, $secretKey);
$montoEncriptado = $aes -> EncryptString($amount, $secretKey);
$sucursalEncriptada = $aes -> EncryptString($sucursal, $secretKey);
$url = 'https://sandboxpp.asjservicios.com.ar/';

$data = array("p1", "p2", "p3");

?>
```

```
<html>
<head>
  <meta name="referrer" content="no-referrer">
</head>
<body>
  <form method="post" action="?=$url?" id="form-firma">
    <input type="hidden" name="CallbackSuccess" id="CallbackSuccess" value="?=$callbackEncriptada; ?" /><br />
    <input type="hidden" name="CallbackCancel" id="CallbackCancel" value="?=$cancelEncriptada; ?" /><br />
    <input type="hidden" name="COMERCIO" id="Comercio" value="?=$comercio?" /><br />
    <input type="hidden" name="SucursalComercio" id="Sucursal" value="?=$sucursalEncriptada?" /><br />
    <input type="hidden" name="TransaccionComercioId" value="?=$transaccion_numero?"
id="TransaccionComercioId" /><br />
    <input type="hidden" name="Monto" id="Monto" value="?=$montoEncriptado?" /><br />
    <input type="hidden" name="Producto[0]" id="producto1" value="?=$transaccion_item?" /><br />
    <input type="hidden" name="MontoProducto[0]" id="montoproducto1" value="?=$transaccion_priceitem?" /><br />
    <input type="hidden" name="Hash" id="firma" class="col-md-6" value="?=$hashfinal?" readonly /><br />
    <button type="submit" id="botonForm" style="display: none" /><br></button></form>
    <script language="javascript">document.getElementById('botonForm').click();</script>
    <!--<input type="submit" value="COMPRAR" />-->
  </form>
</body>
</html>
```

3.3 Formulario POST

Incluir el hash obtenido anteriormente, en la solicitud **POST** sobre **HTTPS** que envía los datos de la compra por parte del comercio, a la **URL** correspondiente según el ambiente con el que se esté integrando. Los parámetros a enviar mediante **POST** son los siguientes:

Nombre	Formato	Descripción	Observaciones	Encriptado
CALLBACKSUCCESS	Alfanumérico	Url a la que se debe redireccionar el usuario una vez finalizada la transacción.	Ej: http://comercio.com?articuloid=123	SI
CALLBACKCANCEL	Alfanumérico	Url a la que se debe redireccionar al usuario cuando se cancele el pago.	Ej: http://comercio.com	SI
COMERCIO	Alfanumérico	Identificador único del comercio en Macro Click de Pago.	-	NO
SUCURSALCOMERCIO (opcional)	Alfanumérico	Identificador único de la sucursal del comercio en Macro Click de Pago.	-	SI
TRANSACCIONCOMERCIOID	Alfanumérico	Identificador mediante el cual el comercio identifica unívocamente la transacción. Debe ser único para cada operación.	-	NO
MONTO	Númérico con dos lugares decimales sin puntos ni comas ni símbolos de moneda	Total, a pagar	Ej: \$ 943,00 → 94300 \$ 4,00 → 400	SI
PRODUCTO[n]	Alfanumérico	Descripción de los productos que componen el pago.	El primer producto debe enviarse con nombre PRODUCTO[0]. El segundo con PRODUCTO[1] El producto N con PRODUCTO[n-1]	NO

Nombre	Formato	Descripción	Observaciones	Encriptado
MONTOPRODUCTO[n]	Numérico con dos lugares decimales sin puntos ni comas ni símbolos de moneda	Importe correspondiente a los productos indicados con anterioridad.	-Se deberán enviar tantos importes como productos se envíen. El importe correspondiente al primer producto se deberá enviar como MontoProducto[0]. El monto correspondiente al producto N se deberá enviar en el campo MontoProducto[n-1]. Opcional.	NO
HASH	Alfanumérico	Hash de verificación de la transacción.	-	NO
INFORMACION	Alfanumérico	Campo adicional que permite enviar información extra propia del comercio.	Opcional. En formato JSON. En caso de enviarlo, se debe encriptar.	SI
CLIENTDATA.CUIT	Alfanumérico	CUIT del cliente sin guiones.	Opcional	NO
CLIENTDATA.NOMBREAPELLIDO	Alfanumérico	Nombre y apellido del cliente.	Opcional	NO
CALLBACKPENDING	Alfanumérico	URL a que se debe redireccionar al usuario cuando se realiza la creación de un DEBIN correctamente.	Opcional. En caso de enviarlo se debe encriptar.	SI (solo si se envía URL)
USERID	Numérico	Identificador del usuario (Sólo para integración con Tarjetero)	Opcional. Sólo en caso de enviarlo se debe encriptar.	Si (sólo si se envía)

A continuación, se presenta el ejemplo para la compra de tres artículos por un monto total de \$3789,00.

```
<html>
<head>
  <meta name="referrer" content="no-referrer">
</head>
<body>
<form method="post" action="https://sandboxpp.asjservicios.com.ar">
  <input type="hidden" name="CallbackSuccess"
    value="vh0BWNrIATF5r3Td5++2iEPPyoTV012AZTF2hqC4KRY" ;/> //www.google.com
  <input type="hidden" name="CallbackCancel"
    value="vh0BWNrIATF5r3Td5++2iEPPyoTV012AZTF2hqC4KRY=" /> //www.google.com
  <input type="hidden" name="Comercio" value="aaaaaaaa-aaaa-aaaa-aaaa-aaaaaaaaaaaa" />
  <input type="hidden" name="SucursalComercio"
    value="WRc5LDcZyNDVykouWjysfV4+skt1lf55z9eBanWaq0yNrcznN1uBjEvj4" />
  <input type="hidden" name="Hash"
    value="4be6253c592a60c3ffb27acc9d130908b05f7e10fd3ae671d326b5224e69dff6" />
  <input type="hidden" name="TransaccionComercioId" value="abc001" />
  <label>Monto: </label>
    <input type="text" name="Monto"
      value="9G3kuKBjQEv3qXWHUFR1hX/dukOKa4j2uYGvciwaLeI=" /> //378900
    <label>Producto 1: </label>
    <input type="text" name="Producto[0]" value="Descripción artículo N°1" />
    <label>Producto 2: </label>
    <input type="text" name="Producto[1]" value="Descripción artículo N°2" />
    <label>Producto 3: </label>
    <input type="text" name="Producto[2]" value="Descripción artículo N°3" />
    <label>Monto Producto 1: </label>
    <input type="text" name="MontoProducto[0]" value="300000" />
    <label>Monto Producto 2: </label>
    <input type="text" name="MontoProducto[1]" value="70000" />
    <label>Monto Producto 3: </label>
    <input type="text" name="MontoProducto[2]" value="8900" />
    <input type="text" name="Informacion"
      value="vh0BWNrIATF5r3Td5++2iEPPyoTV012AZTF2hqC4KRY=" />
    <input type="hidden" name="ClientData.CUIT" value="23456456769" />
    <input type="hidden" name="ClientData.NombreApellido" value="Pedro Martinez" />
    <button type="submit">ENVIAR</button>
</form>
</body>
</html>
```

3.4 Proceso de pago en Macro Click de Pago

Una vez enviado el formulario mediante **POST**, se mostrará al comprador la siguiente pantalla, donde podrá visualizar un resumen de la compra realizada (detalle de los productos y monto total de la compra) y deberá ingresar información respecto al medio de pago a utilizar:

LOGO COMERCIO

09:26 Minutos para finalizar la transacción

Medio de pago

Seleccione

Titular de la tarjeta

Tal como está impreso en la tarjeta.

Número de tarjeta

Número de tarjeta, se recomienda un dígito a la vez.

Vencimiento

Mes

Año

2018

Código Seguridad

Email

Tipo

DNI

Número Documento

Calle

Número

Donde llega el resumen de tu tarjeta.

Fecha Nacimiento

Día

Mes

Año

RESUMEN DE COMPRA

Producto 1

Producto 2

Total a pagar:

\$ 25,25

☐ Acepto los términos y condiciones y política de privacidad.

☐ Acepto guardar mis datos.

Cancelar

Pagar

PlusPagos

© 2018 Todos los derechos reservados

- LOGO COMERCIO:** Se mostrará el logo de su comercio.
- MEDIO DE PAGO:** Se listarán los medios de pago habilitados para su comercio. Se incluirá a su vez, un campo selector de cuotas en caso de que el medio de pago admita más de una cuota.
- TITULAR DE LA TARJETA, NÚMERO DE TARJETA, VENCIMIENTO, CÓDIGO DE SEGURIDAD:** Información de la tarjeta a utilizar. Esta información NO SERÁ ALMACENADA.
- EMAIL, TIPO, NÚMERO DOCUMENTO, CALLE, NUMERO, FECHA NACIMIENTO:** Esta información sólo se almacenará si el comprador marca la casilla "Acepto guardar mis datos".
- RESUMEN DE COMPRA:** Si en la integración se envía, el monto por cada producto (campo MontoProducto[n]), se visualizará en el resumen de pago del formulario.

07:52 Minutos para finalizar la transacción

Medio de pago

-Seleccione-

Titular de la tarjeta

Tal como está impreso en la tarjeta.

Número de tarjeta

Número de tarjeta, se recomienda un dígito a la vez.

Vencimiento

Mes

Año

Cód. Seguridad

RESUMEN DE PAGO

Producto 1

100,00

Producto 2

70,00

Producto 3

5,00

Total a pagar:

\$ 175,00

☐ Acepto los términos y condiciones y política de privacidad.

☐ Acepto guardar mis datos (opcional).

Pagar

Es importante mencionar que el sistema no realiza la validación de que la suma de los montos de los productos discriminados coincida con el valor enviado en el campo **Monto**.

Al confirmar el pago y realizarse exitosamente, se redireccionará a la URL indicada en la solicitud **POST**, en el campo **CALLBACKSUCCESS**.

En caso de que el usuario cancele el pago se redireccionará a la URL indicada en la solicitud **POST**, en el campo **CALLBACKCANCEL**.

En caso de utilizar el medio de pago **DEBIN** y que la operación de creación de este sea correcta, se redireccionará a la URL indicada en la solicitud **POST**, en el campo **CALLBACKPENDING**.

En caso de ocurrir un error en el pago, se lo notificará al usuario en la misma pantalla del formulario de pago.

La transacción exitosa será confirmada mediante el envío de un correo electrónico al comercio y otro al usuario (Dicha configuración será acordada con **Macro Click de Pago**).

3.5 Cancelación de un pago en Macro Click de Pago

El comprador tiene la posibilidad de cancelar un pago, a través del botón Cancelar del formulario de pago. Este evento dispara un **POST** desde **Macro Click de Pago** al comercio que contiene en el body del request un objeto en formato **JSON** con la siguiente información:

- Tipo: TRANSACCIÓN
- TransaccionPlataformald: El ID de su transacción en Macro Click de Pago.
- TransaccionComerciold: El ID de su transacción en su comercio que proporcionó con anterioridad.
- Monto: El monto de la transacción de pago que se realizó.
- Estado: CANCELADA.
- Detalle: "El cliente decidió cancelar la transacción de pago."
- FechaTransaccion: La fecha del momento en que la transacción se creó/inició.
- FechaPago: La fecha del último pago en estado **REALIZADO**, sólo aplicable a transacciones **REALIZADAS**.

3.6 Expiración de transacciones en Macro Click de Pago

Cada un cierto tiempo, Macro Click de Pago buscará las transacciones que no hayan sido completadas por el comprador y actualizará su estado a **EXPIRADA**, enviándose un **POST** al comercio que contiene el body del request un objeto en formato **JSON** con la siguiente información:

- Tipo: TRANSACCIÓN
- TransaccionPlataformald: El ID de su transacción en Macro Click de Pago.
- TransaccionComerciold: El ID de su transacción en su comercio que proporcionó con anterioridad.
- Monto: El monto de la transacción de pago que se realizó.
- Estado: EXPIRADA.
- Detalle: "Se actualizó el estado de la transacción a **EXPIRADA**."
- FechaTransaccion: La fecha del momento en que la transacción se creó/inició.
- FechaPago: La fecha del último pago en estado **REALIZADO**, sólo aplicable a transacciones **REALIZADAS**.

4. Notificaciones

Al dar de alta el comercio en **Macro Click de Pago**, se le solicitará una URL de notificaciones (esto es opcional, en el caso de que se deseen recibir notificaciones). **Macro Click de Pago** enviará notificaciones a la URL de notificaciones informada anteriormente, mediante una llamada **HTTP POST**.

Estas notificaciones se envían cuando:

- Se realiza correctamente un pago
- Se ejecuta una anulación o devolución de un pago
- El usuario cancela un pago estando en el formulario de pago

La notificación enviada tendrá el formato descrito a continuación:

Notificaciones de pago exitoso/no exitoso:

- Tipo: PAGO
- TransaccionPlataformald: El ID de su transacción en **Macro Click de Pago**.
- TransaccionComerciold: El ID de su transacción en su comercio que proporcionó con anterioridad.
- Monto: El monto de la transacción (pago) que realizó.
- Estado: Descripción del estado.
- Detalle: -
- MedioPago: Código que identifica al tipo de medio de pago utilizado.

Código	Valor
8	CRÉDITO
9	DÉBITO
0	EFFECTIVO
10	DEBIN
11	CRÉDITO EN CUOTAS

EstadoId: Código que identifica al estado de la transacción.

Código	Valor
1	CREADA
2	EN_PAGO
3	REALIZADA
4	RECHAZADA
7	EXPIRADA
8	CANCELADA
9	DEVUELTA
10	PENDIENTE
11	VENCIDA

MarcaTarjeta: Marca de la tarjeta que procesó el pago.

• Visa Crédito	• Nexo	• Patagonia 365
• Maestro	• Credimás	• Tarjeta Club Día
• Cupón de pago	• Tarjeta Nevada	• Tuya
• Visa Débito	• Nativa	• Distribution
• Diners Club	• MasterCard Débito	• Tarjeta La Anónima
• MasterCard	• Cabal Débito	• CrediGuia
• Tarjeta Shopping	• Tarjeta Cencosud	• Cabal Prisma
• Tarjeta Naranja	• Tarjeta Carrefour / Cetelem	• Tarjeta SOL
• Cabal	• Tarjeta PymeNación	• American Express
• Italcared	• BBPS	• Favacard
• ArgenCard	• Qida	
• CoopePlus	• Grupar	

InformacionPagador: información del pagador, enviada sólo cuando está disponible.

InformacionPagador	
Campo	Tipo
Email	String
Nombre	String
NumeroDocumento	String
Telefono	String
TipoDocumento	String

- **InformacionAdicional:** información enviada durante la integración en el campo Informacion del POST.
- **InformacionAdicionalLink:** información de los campos personalizados de los links de pago.
- **Resultado:** en caso de estar disponible se envía el motivo de rechazo del pago junto a un código interno. También se envía en caso de pago correcto.
- **FechaTransaccion:** La fecha del momento en que la transacción se creó/inició.
- **FechaPago:** La fecha del último pago en estado REALIZADO, sólo aplicable a transacciones REALIZADAS.
- **ProductoTransaccion:** Listado de productos del pago. Se indica nombre del producto y monto del producto.

Notificaciones de devoluciones/anulaciones:

Al realizarse una devolución, el comercio recibirá un correo electrónico informativo al respecto. El cliente también recibirá un correo informándole respecto a la correcta realización de la devolución.

Macro Click de Pago envía mediante POST, la siguiente información, en caso de que el comercio decida almacenarla para tener un control de las devoluciones o realizar alguna acción:

- **Tipo:** DEVOLUCIÓN
- **TransaccionPlataformald:** El ID de su transacción en **Macro Click de Pago**.
- **TransaccionComerciold:** El ID de su transacción en su comercio que proporcionó con anterioridad.
- **Monto:** El monto de la devolución que se realizó.
- **Cuotas:** N.º de cuotas de la transacción.
- **MedioPago:** Código que identifica al tipo de medio de pago utilizado.

Código	Valor
8	CRÉDITO
9	DÉBITO
0	EFFECTIVO
10	DEBIN

- **Estado:** Descripción del estado de devolución.
- **Estadold:** Código que identifica al estado de la devolución.

Código	Valor
1	DEVOLUCION_PENDIENTE_APROBACION
2	DEVOLUCION_APROBADA
3	DEVOLUCION_NO_APROBADA
4	DEVOLUCION_RECHAZADA

- **FechaTransaccion:** La fecha del momento en que la transacción se creó/inició.
- **FechaPago:** La fecha del último pago en estado REALIZADO, sólo aplicable a transacciones REALIZADAS.

Notificaciones de cancelación de un pago por parte del usuario:

- Tipo: TRANSACCION
- TransaccionPlataformald: El ID de su transacción en Macro Click de Pago.
- TransaccionComerciold: El ID de su transacción en su comercio que proporcionó con anterioridad.
- Monto: El monto de la transacción (pago) que se realizó.
- Estado: CANCELADA.
- Detalle: “El cliente decidió cancelar la Transacción de Pago.”
- FechaTransaccion: La fecha del momento en que la transacción se creó/inició.
- FechaPago: La fecha del último pago en estado REALIZADO, sólo aplicable a transacciones REALIZADAS.

Ejemplo:

Response

```
POST /api/post/ HTTP/1.1 200 OK
Host: testpost.org:8087
Content-Type: application/json
cache-control: no-cache

{
  "TransaccionComercioId": "603847",
  "TransaccionPlataformaId": "28397",
  "Tipo": "PAGO",
  "Monto": "1500,50",
  "Estado": "REALIZADA",
  "Detalle": "Pago realizado exitosamente.",
  "MetodoPago": null,
  "MedioPago": "8",
  "EstadoId": "3",
  "Cuotas": "1",
  "InformacionPagador": {
    "Email": "agustin.grau@gmail.com",
    "Nombre": "dasdsa",
    "NumeroDocumento": "213321",
    "Telefono": null,
    "TipoDocumento": "DNI"
  },
  "InformacionAdicional": "",
  "MarcaTarjeta": "Visa Crédito",
  "InformacionAdicionalLink": null,
  "FechaTransaccion": "dd/MM/yyyy hh:mm:ss",
  "FechaPago": "dd/MM/yyyy hh:mm:ss",
  "ProductoTransaccion": [
    {
      "NombreProducto": "Factura Julio 2020",
      "MontoProducto": 1000.00
    },
    {
      "NombreProducto": "Factura Agosto 2020",
      "MontoProducto": 500.00
    }
  ]
}
```


Si se quiere restringir desde donde ingresan las llamadas HTTP POST, a continuación, se listan las IP publicas desde donde se originan las notificaciones HTTP REQUEST:

- 190.104.203.202
- 190.111.243.247
- 190.210.103.178
- 190.210.103.179

5. Devoluciones

Para solicitar una devolución, deberá comunicarse con **Macro Click de Pago** informando el identificador interno y el identificador de la plataforma (**TRANSACCIONID** y **TRANSACCIONPLATAFORMAID**) de la transacción a devolver.

Una vez informada, la transacción quedara en estado **PENDIENTE** para su posterior revisión por parte de **Macro Click de Pago**.

Al confirmarse la devolución, el comercio recibirá un correo electrónico informativo al respecto. En caso de que el comprador haya aceptado el guardado de sus datos, también recibirá un correo informándole respecto a la correcta realización de la devolución.

Macro Click de Pago envía mediante POST, en el body del request un objeto en formato **JSON** con la siguiente información, en caso de que el comercio decida almacenarla para tener un control de las devoluciones o realizar alguna acción:

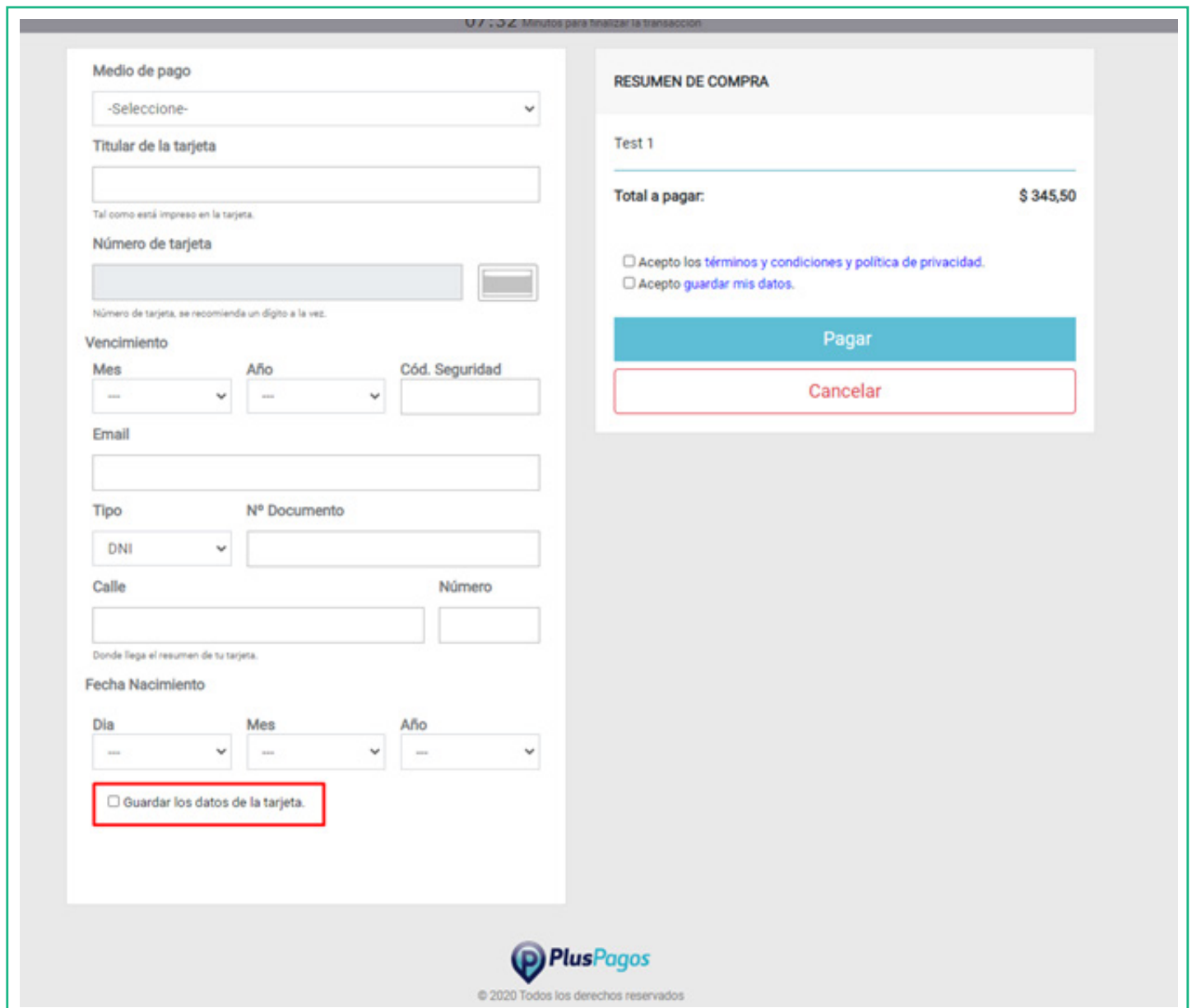
- Tipo: DEVOLUCIÓN
- TransaccionPlataformald: El ID de su transacción en Macro Click de Pago.
- TransaccionComerciold: El ID de su transacción en su comercio que proporcionó con anterioridad.
- Monto: El monto de la devolución que se realizó.
- Estado: APROBADA.
- Detalle: "Se aprobó exitosamente la solicitud de Devolución para la Transacción."
- FechaTransaccion: La fecha del momento en que la transacción se creó/inició.
- FechaPago: La fecha del último pago en estado REALIZADO, sólo aplicable a transacciones REALIZADAS.

6. Tarjetero

Esta funcionalidad debe ser dada de alta por Macro Click de Pago.

Almacén de tarjetas

Si el comercio cuenta con un tarjetero configurado, deberá enviar el UserId en el POST, de manera encriptada. De esta manera, cuando el usuario realice un pago podrá elegir la opción de guardar la tarjeta utilizada para usos posteriores. Si el pago es correcto, se almacenará una tarjeta para ese usuario.



07:32 Minutos para finalizar la transacción

Medio de pago

-Selecione-

Titular de la tarjeta

Tal como está impreso en la tarjeta.

Número de tarjeta

Número de tarjeta, se recomienda un dígito a la vez.

Vencimiento

Mes Año Cód. Seguridad

Email

Tipo N° Documento

DNI

Calle Número

Donde llega el resumen de tu tarjeta.

Fecha Nacimiento

Día Mes Año

☐ Guardar los datos de la tarjeta.

RESUMEN DE COMPRA

Test 1

Total a pagar: \$ 345,50

☐ Acepto los términos y condiciones y política de privacidad.

☐ Acepto guardar mis datos.

Pagar

Cancelar

PlusPagos

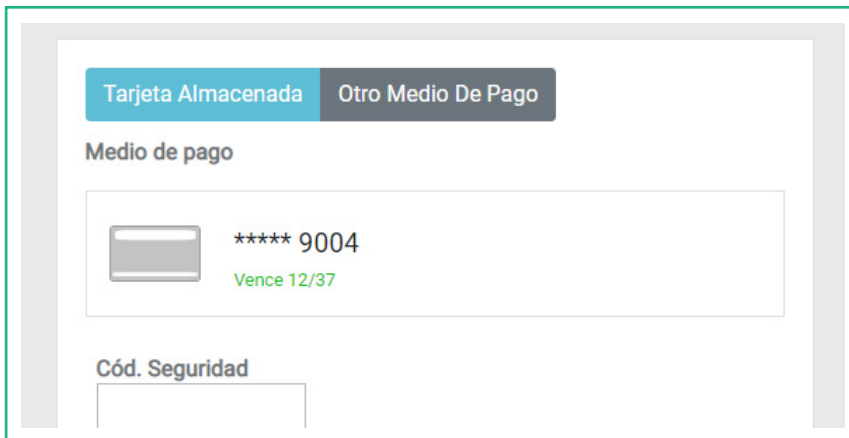
© 2020 Todos los derechos reservados

Pago con tarjeta almacenada

Para poder realizar un pago con una tarjeta almacenada, se debe enviar en el POST, el usuario del cual se van a listar las tarjetas en el campo UserId de manera encriptada.

Si el usuario posee tarjetas almacenadas podrá visualizarlas en el formulario de pago y utilizarlas para realizar un pago.

Para realizar un pago, seleccionar la tarjeta almacenada a utilizar (la misma se encuentra seleccionada cuando el recuadro se vuelve azul), ingresar el código de seguridad y el email y clickear el botón Pagar.



Formulario de pago con tarjeta almacenada. Incluye pestañas "Tarjeta Almacenada" (seleccionada) y "Otro Medio De Pago". Se muestra el medio de pago "***** 9004" con vencimiento "Vence 12/37". Hay un campo para el "Cód. Seguridad".



Formulario de pago completo. Encabezado: LOGO COMERCIO, 07:49 Minutos para finalizar la transacción. Sección de pago: Tarjeta Almacenada (seleccionada), Otro Medio De Pago. Medio de pago: ***** 9004, Vence 12/37. Campos: Cód. Seguridad, Email. Resumen de compra: Producto, Total a pagar: \$ 345,50. Opciones: ☐ Acepto los términos y condiciones y política de privacidad., ☐ Acepto guardar mis datos. Botones: Pagar, Cancelar. Logo PlusPagos. Copyright: © 2020 Todos los derechos reservados.

En el formulario, el usuario tendrá la posibilidad de guardar una nueva tarjeta si selecciona la opción “Otro medio de pago”.

02:16 Minutos para finalizar la transacción

Tarjeta Almacenada

Otro Medio De Pago

Medio de pago

-Seleccione-

Titular de la tarjeta

Tal como está impreso en la tarjeta.

Número de tarjeta

Número de tarjeta, se recomienda un dígito a la vez.

Vencimiento

Mes

Año

Cód. Seguridad

Email

Tipo

DNI

Nº Documento

Calle

Número

Donde llega el resumen de tu tarjeta.

Fecha Nacimiento

Día

Mes

Año

☐ Guardar los datos de la tarjeta.

RESUMEN DE COMPRA

Test 1

Total a pagar:

\$ 345,50

☐ Acepto los términos y condiciones y política de privacidad.
 ☐ Acepto guardar mis datos.

Pagar

Cancelar

7. Entorno de Sandbox

Macro Click de Pago cuenta con un entorno de Sandbox para poder efectuar la implementación contra un ambiente de pruebas, sin necesidad de realizar transacciones con información verdadera.

Para utilizar el entorno de Sandbox deberá efectuar el **POST** especificado en la **sección 3** a la siguiente URL: <https://sandboxpp.asjservicios.com.ar/>

Luego, en el formulario de **Macro Click de Pago**, se podrá utilizar la siguiente información para efectuar transacciones de prueba:

Número tarjeta	Fecha expiración	CVV	Tipo Tarjeta	Marca	Montos aceptados	Montos rechazados
4507990000004905	08/23	123	Crédito	VISA	Cualquier monto	Ninguno
4000000000000002	10/22	123	Crédito	VISA	\$1 a \$1000 ó mayores a \$2000	\$1000.01 a \$1999.99
5590000000000005	10/22	123	Crédito	MASTERCARD	\$1 a \$1000 ó mayores a \$2000	\$1000.01 a \$1999.99
376411234531007	12/23	1234	Crédito	AMEX	\$1 a \$1000 ó mayores a \$2000	\$1000.01 a \$1999.99
5010900000000008	10/22	123	Débito	MAESTRO	\$1 a \$1000 ó mayores a \$2000	\$1000.01 a \$1999.99
5895627823453005	08/23	123	Crédito	NARANJA	Cualquier monto	Ninguno