Especificaciones Técnicas para integrar PagosWeb

Protocolo versión 3.4

siemprepago

Joaquín Requena 1580 Of. 102 • Montevideo • Uruguay
*Teléfonos: +598 2402 2330**
www.pagosweb.com.uy • e-mail: comercial@pagosweb.com.uy

Versiones

Versión	Fecha	Comentarios
1.0	15/04/2008	Versión inicial.
2.0	20/07/2009	Modificaciones para Plugin VISA.
3.1	07/10/2010	Actualización de documentación.
3.2	11/11/2011	Actualización de diagramas y formato general del documento.
3.3	02/05/2012	Actualización de nombres de procedimientos a invocar.
3.4	04/06/2012	Se agrega parámetro opcional *numeroOrden* en el Response y URL de pruebas.
3.5	20/07/2012	Ajustes de redacción y actualización de URL de testing.
3.6	23/10/2012	Cambios por la versión 3.3 del protocolo, configuración de OCA y uso de llave 3DES. Campo *numeroOrden* pasa de alfanumérico a numérico.
3.7	15/02/2013	Se agrega configuración de RedPagos y Componente PHP.
3.8	15/04/2013	Se agrega configuración de e-BROU y Banred.
3.9	22/05/2013	Se agrega configuración de Skrill, Diners, DinersDiscover y Lider.
3.10	08/11/2013	Se modifica algoritmo de encriptación, se modifica método de acceso OCA.
3.11	28/05/2014	Se agrega tabla de medios de pago disponibles. Se agrega campo *cantidadCuotas* en el response. Se aclara operativa con la cédula para RedPagos.
3.12	02/07/2014	Creación de protocolo v 3.4 que contiene los cambios necesarios para la ley 19210 (ley de inclusión financiera).
3.13	01/10/2014	Se modifica ejemplo de desencriptación en PHP.
3.14	18/11/2014	Modificamos largo de número de factura.
3.15	20/03/2015	Se agrega Abitab como medio de pago. Se agrega URL Offline de notificaciones.
3.16	01/04/2015	Se agrega documentación de Abitab.
3.17	01/05/2015	Se agrega documentación de Url Offline y Configuración Abitab.
3.18	24/09/2015	Se agrega campo opcional *tipoDocumento.*
3.19	26/10/2015	Mejoras en explicación de notificaciones offline.
3.20	27/10/2015	Mejoras en explicación de notificaciones offline.
3.21	15/03/2016	Se agrega aclaración de formato de fecha.
3.22	28/03/2016	Se agrega ID de tarjeta para integrar CABAL.

3.23	31/05/2016	Se corrigen detalles de formato.
3.24	07/08/2017	Se hacen comentarios sobre el tratamiento de cuotas para VISA.
3.25	24/04/2018	Obligatoriedad de Teléfono, tipo documento y documento para OCA. Dirección obligatoria para VISA
3.26	16/07/2019	Se agregan los medios de pago del 17 al 27
3.27	14/08/2019	Se agregan los medios de pago 28 (HSBC) y 29 (Scotia)

Contenido del documento

Introducción

El objetivo de este documento es mostrar cómo integrar la solución de pagos ofrecida por PagosWeb con el sitio web de compras.

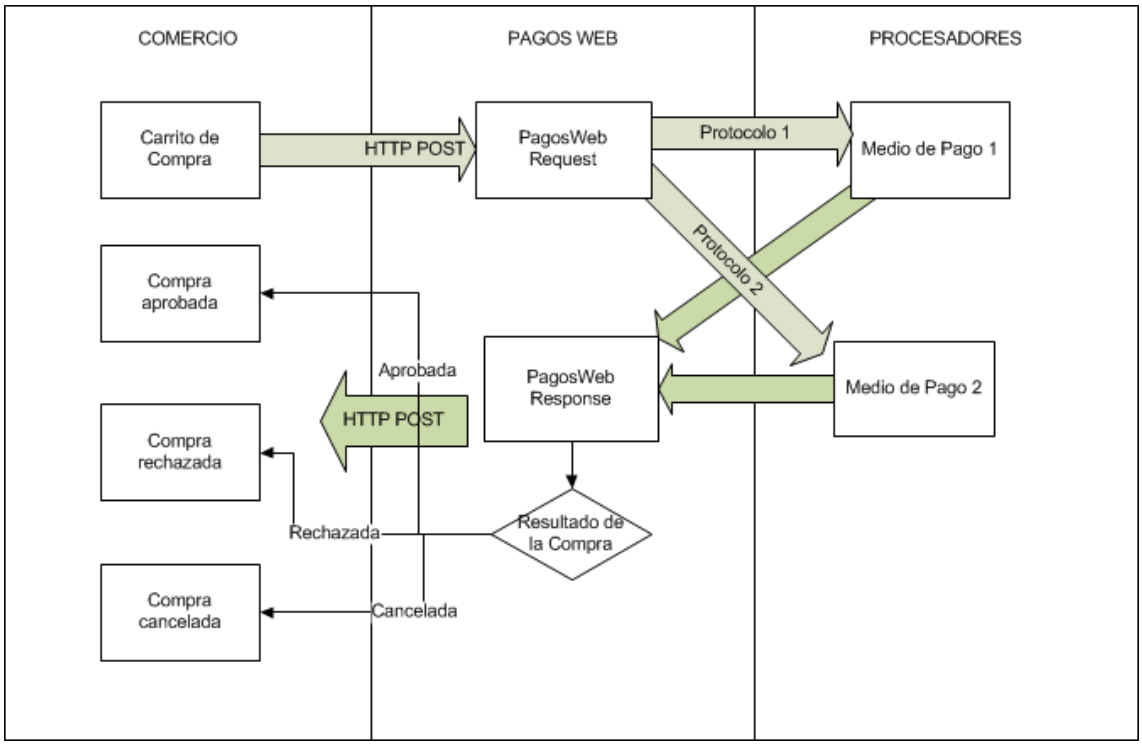
Solución de Pagos con PagosWeb

Esta solución consta de un componente que resuelve la comunicación entre el sitio web del comercio y las plataformas de pago seguro de los distintos medios de pago.

Dicho componente funciona como vínculo seguro entre el comercio y la página de pagos y se inicia cuando el socio efectúa una compra en el sitio web del comercio.

Descripción del Proceso de Pago

Diagrama de Alto Nivel



Proceso de Compra

- El cliente ingresa normalmente al sitio web del comercio, donde elige los artículos o servicios que desea comprar. Cuando llena su carrito de compra y va a la página en la cual podrá pagar por los mismo a través de los medios de pago disponibles.

- Cuando el cliente elige uno de estos medios de pago, la página del comercio debe redirigir el explorador del cliente hacia la página de Request de PagosWeb a través de HTTP/POST.

- El sistema PagosWeb redirigirá nuevamente al cliente a la página web del procesador correspondiente donde ingresará los datos de la tarjeta de crédito o medio de pago elegido. En este punto el número de tarjeta de crédito, fecha de vencimiento, CVC u otros datos confidenciales son encriptados y manejados por los servidores del medio de pago correspondiente.

- Cuando el cliente ingrese los datos requeridos por el medio de pago y confirme (o cancele) la transacción, el medio de pago enviará nuevamente el explorador hacia la página de procesamiento de respuesta de PagosWeb.

- Por último, luego de haber ejecutado el paso anterior, el sistema PagosWeb redirigirá a través de HTTP/POST hacia una de las páginas proporcionadas por el comercio al momento de la afiliación al sistema PagosWeb.

Se deben configurar 3 URLs que corresponden a los distintos resultados a los que puede llegar una compra, los cuales se detallan a continuación:

  - **Aprobada** El medio de pago ha aprobado la compra y la misma fue registrada.

  - **Rechazada** El medio de pago rechazó la compra, este estado se puede alcanzar por ejemplo si la tarjeta habiente no tiene suficiente saldo, o existe algún problema con la tarjeta de crédito o medio de pago elegido.

  - **Cancelada** El usuario ha cancelado la compra al momento de ingresar los datos de la tarjeta de crédito o medio de pago elegido.

- Si el cliente no finaliza el proceso de compra correctamente en el medio de pago, la transacción quedará incompleta y en la pasarela se registrará con el estado "*Request Enviado*". Esto se puede dar en los casos en que el cliente cierra el navegador inesperadamente o cuando el cliente no finaliza

correctamente la operación en el medio de pago y no aguarda el retorno al sitio del comercio.

Confirmación de Compra

La operativa con las redes físicas y los bancos nos proporciona la alternativa técnica de recibir una confirmación Offline de la realización del pago. Dado que en las redes físicas (RedPagos, Abitab) el pago real de la compra se realiza cuando el cliente se presenta en caja, esta operación permite recibir una confirmación de que el cliente efectivizó el pago en caja.

Para ello el comercio puede proporcionar una URL de notificación a la cual el sistema PagosWeb invocará a través de HTTP/POST comunicando el pago de la transacción en el medio de pago (RedPagos, Abitab , Santander, BROU, BBVA).

En el caso de los bancos siempre se recibe la notificación Offline, por lo tanto si el cliente realizó correctamente la operación de pago en el sitio del banco podemos recibir 2 confirmaciones del mismo. Una en el momento de efectivizar el pago y la segunda por la notificación Offline. En caso que el proceso finalice correctamente, pero el usuario no sea redireccionado nuevamente al sitio del comercio, se recibirá una sola notificación.

Las campos recibidos en el POST se encuentran detallados en el Item*Comunicación HTTP/Response de PagosWeb.*

Secuencia de una Transacción

Cliente	Carrito de compras	PagosWeb
• Selecciona un producto y hace clic en agregar al carrito		
	• Registra el producto	
	• Hace clic en el botón comprar.	
	• Muestra el formulario de compra y solicita los datos del cliente.	
• Ingresa los datos y hace clic en "Pagar"		
	• Genera el formulario para enviarlo por POST a PagosWeb.	
		• Recibe las variables e inicia la transacción, empaquetando lo datos de acuerdo a las especificaciones del medio de pago utilizado.
		• Redirige el navegador del cliente a la página del medio de pago donde deberá ingresar los datos solicitados por estas firmas.
• Ingresa datos de la tarjeta de crédito o medio de pago elegido y acepta las condiciones.		

		• *Procesa la transacción e invoca la página de respuesta de PagosWeb*
		• *Procesa la información recibida del medio de pago correspondiente y verifica si la transacción fue aprobada o negada.*
		• *Redirige al cliente a la página del comercio.*

Implementación

Configuración del sistema

Configuración PagosWeb

♦ **URL Compra aprobada**. URL del sitio del comercio a donde invocar desde PagosWeb si la transacción de compra fue aprobada por el procesador.

♦ **URL Compra rechazada**. URL del sitio del comercio a donde invocar desde PagosWeb si la transacción de compra fue rechazada por el procesador.

♦ **URL Compra cancelada**. URL del sitio del comercio a donde invocar desde PagosWeb si el procesador de la compra indica que la transacción de compra fue cancelada por el usuario.

♦ **URL Notificación Offline**. URL del sitio del comercio a donde invocar desde PagosWeb cuando el procesador de la compra indica que la transacción fue paga por el usuario.
Disponible: RedPagos, Abitab, Santander, BROU, BBVA.

Configuración VISA

La siguiente información es necesaria para el uso de Visa a través PagosWeb.

La información solicitada, es provista por VISA al comercio en el momento de su afiliación

1. Código de Adquirente
2. Código de Comercio
3. Código de Terminal
4. Vector de Inicialización
5. Llaves Públicas de Cifrado y Firma de Certificado Digital
6. Se solicita al comercio Dirección como campo obligatorio

Configuración MasterCard

La siguiente información es necesaria para el uso de MasterCard a través PagosWeb.

La información solicitada, es provista por MasterCard al comercio en el momento de su afiliación

1. Número de Comercio
2. SERVLET de comunicación con MasterCard

Configuración OCA

La siguiente información es necesaria para el uso de OcaCard a través PagosWeb.

La información solicitada, es provista por OCA SA al comercio en el momento de su afiliación

1. Número de Comercio
2. Número Terminal
3. Tipo de Compra (0 = común, 3 = Metros)
4. También se solicita al comercio como obligatorio el Nro de documento, Tipo Documento y el teléfono.

Configuración Cabal

La siguiente información es necesaria para el uso de Cabal a través PagosWeb.

La información solicitada, es provista por Cabal al comercio en el momento de su afiliación

1. Número de Comercio en Pesos Uruguayos
2. Número de Comercio en Dólares Americanos

Configuración RedPagos

La siguiente información es necesaria para el uso de RedPagos a través PagosWeb.

La información solicitada, es provista por RedPagos al comercio en el momento de su afiliación

1. Número de Comercio
2. Código de Comercio

También se le solicita al comercio

3. Email de Notificación.
4. Tiempo de Vencimiento de Compra
5. Cédula, es un campo obligatorio.

Al usar RedPagos a través de PagosWeb este genera una pantalla de confirmación que se muestra a continuación.

La cual solicita al comprador que acepte o cancele la compra y además le muestra la lista de transacciones pendientes de pago.

Si el comprador acepta la transacción se le enviara un cupón de compra, por lo que se le solicita al comercio que valide el email del comprador.

Configuración Abitab

La siguiente información es necesaria para el uso de Abitab a través PagosWeb.

La información solicitada, es provista por Abitab al comercio en el momento de su afiliación

1. Número de Empresa

También se le solicita al comercio

2. Email de Notificación.
3. Tiempo de Vencimiento de Compra
4. Cédula,es un campo obligatorio

Al usar Abitab a través de PagosWeb este genera una pantalla de confirmación que se muestra a continuación.

La cual solicita al comprador que acepte o cancele la compra y además le muestra la lista de transacciones pendientes de pago.

Si el comprador acepta la transacción se le enviara un cupón de compra, por lo que se le solicita al comercio que valide el email del comprador.

Configuración e-BROU

La siguiente información es necesaria para el uso de e-BROU a través PagosWeb.

La información solicitada, es provista por e-BROU al comercio en el momento de su afiliación

1. Número de Organismo
2. Tipo de Servicio

Configuración Banred

La siguiente información es necesaria para el uso de Banred a través PagosWeb.

La información solicitada, es provista por Banred al comercio en el momento de su afiliación

1. Código de Comercio

También se le solicita al comercio

2. Tiempo de Vencimiento de Factura

Configuración Skrill

La siguiente información es necesaria para el uso de Skrill a través PagosWeb.

La información solicitada, es provista por Skrill al comercio en el momento de su afiliación

3. Número de Organismo
4. Tipo de Servicio

Comunicación HTTP

Request a PagosWeb

El Request a la página de PagosWeb debe ser efectuado mediante el método HTTP POST.

El formulario HTTP debe contener los siguientes campos:

Campo	Tipo	Presencia	Descripción
idCliente	Numérico	M	Número de cliente dentro de la aplicación PagosWeb, asignado en la instalación. Es fijo a partir de la primera instalación.
idTarjetaCredito	Numérico	M	Ver Detalle de Medios de Pago.
primerNombre	Alfanumérico	M	Primer nombre del cliente.
primerApellido	Alfanumérico	M	Primer apellido del cliente.
email	Alfanumérico	M	Dirección de correo del cliente.
valorTransaccion	Numérico	M	Monto de la compra (sin separadores de miles y como separador de decimales se debe usar el punto). Ejemplo: $ 1.234,50 debe enviarse: 1234.5
cantidadCuotas	Numérico	M	Cantidad de cuotas en que se realiza la compra (ver apartado “Cuotas en VISA”).
moneda	Numérico	M	Códigos de monedas. Valores posibles: 858 – Pesos 840 – Dólares
numeroOrden	Alfanumérico	M	Número de orden del comercio. Sirve como referencia de la transacción efectuada en el comercio.
version	Alfanumérico	M	Versión actual del paquete de información. Valor actual: “3.4”
fecha	Alfanumérico	M	Fecha hora de la transacción. En formato “yyyy-MM-ddhh:mm:ss” (formato 12 horas).

plan	Alfanumérico	M	Campo de largo 2 solo usado por Oca. Especifica el tipo de compra (0 = común, 3 = con metros).
segundoNombre	Alfanumérico	O	Segundo nombre del cliente.
segundoApellido	Alfanumérico	O	Segundo apellido del cliente.
direccionEnvio	Alfanumérico	C	Dirección de envío de la mercadería. Este dato es obligatorio para VISA.
plazoEntrega	Alfanumérico	O	Plazo de entrega declarado al cliente.
telefono	Alfanumérico	C	Teléfono de contacto del cliente. Este campo es obligatorio para OCA.
cedula	Alfanumérico	C	Cédula del cliente. Ver “Condición de cédula”.
consumidorFinal	Numérico	M	(Ley 19210) Se debe indicar si la venta es realizada a un consumidor final. El valor 1 indica que es consumidor final, 0 que no lo es.
importeGravado	Numérico	C	(Ley 19210) Importe gravado por IVA (sin separadores de miles y como separador de decimales se debe usar el punto). Es mandatorio si consumidorFinal=1 y debe tener el valor gravado con IVA. Ver “Detalle de importe gravado”.
numeroFactura	Numérico	C	(Ley 19210) Número de factura que será utilizado por el medio de pago elegido, para informar la devolución de IVA a la DGI. Numérico largo máximo 7 Es aplicable si consumidorFinal=1.
transactionSecurityToken	Alfanumérico	M	Datos de la transacción encriptada con 3DES.
tipoDocumento	Numérico	C	Valores: 1=Cedula 2=Rut 3=Doc. Extranjero. Para uso en las redes físicas y OCA.

Notas

- Los nombres de los campos deben ser establecidos tal cual están escritos en la documentación, respetando minúsculas y mayúsculas.
- Presencia:
  - M = Mandatorio
  - C = Condicional
  - O = Opcional

Condición de Cédula.

La cédula de identidad es el documento de identificación nacional en Uruguay, y está compuesto por un número de 6 a 7 dígitos seguido de un dígito verificador.

Este campo es mandatorio para OCA (ID=3), RedPagos (ID=4) y Abitab (ID=14) y opcional para el resto de los medios de pago.

En caso de ser incluido, se debe validar en el sitio del cliente que el número sea correcto, porque de lo contrario la transacción será rechazada debido a que el cliente no podrá efectuar el pago en las agencias.

Cuotas en VISA

Se debe tener en cuenta que la plataforma de VISA (Alignet) solicita la cantidad de cuotas al cliente luego de que ingrese el número de tarjeta, no dando importancia a lo que el cliente digite en el sitio web.

La cantidad de cuotas que se habilitan en la página de VISA depende de lo que haya acordado con el comercio. PagosWEB no tiene forma de enviar la cantidad de cuotas que el cliente haya elegido y la página de VISA tome ese valor.

En la respuesta al comercio, se envía la cantidad de cuotas seleccionadas por el cliente.

Detalle de Importe gravado

El importe gravado es la parte del monto de la transacción al cual se le aplicó el IVA.

Dado que el monto total de una transacción puede contener valores gravados y no gravados por IVA, la ley de inclusión financiera exige que se reporte este valor para hacer la deducción de IVA correspondiente.

Ejemplo:

> Transacción compuesta por: $ 600 gravados con IVA y $ 300 exentos de IVA.
>
> El valor de la transacción se compone de:
>
> 600 + (600 * 22%) + 300 = 1.032
>
> En este caso, los valores a enviar son:
>
> valorTransaccion = 1032
>
> importeGravado = 600

Algoritmo de validación de cédula

Se adjunta código JavaScript con validación de la cédula de identidad

```
function ValidateDocument(cedula) {
	if (isNaN(cedula) || parseInt(cedula) != cedula) {
	//La cédula no es un número
		return false;
	}
	var multiplicador = [4, 3, 6, 7, 8, 9, 2 ];
	var cd = cedula % 10;
	var i = 0;
	var calc_cd = 0;
	while (cedula> 0 && i< 7) {
		cedula = Math.floor(cedula / 10);
		calc_cd += cedula % 10 * multiplicador[i++];
	}
	calc_cd = (calc_cd % 10 == 0) ? 0 : 10 - (calc_cd % 10);

	if (calc_cd == cd){
		return true;
	}
	else{
		return false;
	}
}
```

Detalle de Medios de Pago

Identificador	Medio de Pago
1	Visa
2	MasterCard
3	Oca
4	RedPagos
5	e-Brou
6	Banred
7	Skrill
8	Diners
9	DinersDiscover
10	Lider
11	Santander
12	BBVA
13	Banque Heritage
14	Abitab
15	Cabal
17	Créditos Directos
24	PassCard
25	Banco Itaú
26	Creditel
27	Banco Bandes
28	Banco HSBC
29	Banco Scotia

URL para Pruebas

http://testing.pagosweb.com.uy/v3.4/requestprocessor.aspx

Response de PagosWeb

Campo	*Tipo*	*Presencia*	*Descripción*
ventaAprobada	Bool	M	true = indica que la venta fue aprobada. En caso de redes físicas puede quedar como: Venta Aprobada o Venta Pendiente (validar en campo Mensaje). false = indica que la venta fue rechazada.
codigoAutorizacion	Alfanumérico	O	Opcionalmente, si la venta fue aprobada puede reportarse un código de autorización provisto por el procesador.
numeroTransaccion	Alfanumérico	M	Número de transacción asignado por el procesador.
monto	Numérico	M	Monto original de la transacción.
mensaje	Alfanumérico	M	Mensaje asociado a la transacción. Se puede mostrar en la página para referencia del cliente.
numeroOrden	Alfanumérico	O	Se devuelve el número de orden que vino originalmente en el Request (se envía de acuerdo a la configuración por comercio).
idCliente	Numérico	M	Número de cliente dentro de la aplicación PagosWeb, asignado en la instalación.
Fecha	Alfanumérico	M	Fecha en que se efectuó la respuesta en formato “yyyy-MM-ddhh:mm:ss”.
cantidadCuotas	Numérico	O	Se devuelve la cantidad de cuotas de la operación. Agregado para notificar especialmente el valor de cuotas elegido en la plataforma de Visa.
responseSecurityToken	Alfanumérico	M	Información de la respuesta

			encriptados con 3DES.

Seguridad 3DES en Request y Response.

En la versión 3.4 de PagosWeb se incluye un campo de seguridad en las transacciones, de esta forma PagosWeb asegura el origen de la transacción. Este campo debe estar cifrando utilizando el algoritmo de cifrado 3DES en su modo CBC (CipherBlockChaining)

En la instalación de un nuevo comercio en PagosWeb se le asigna una llave de encriptado, la cual es enviada al contacto del comercio.

Una vez obtenida la llave hay dos formas de implementar la integración:

Implantación por parte del comercio

Con la llave provista por PagosWeb el comercio debe generar el campo "**transactionSecurityToken**" y puede descifrar el campo "**responseSecurityToken**" con el cual se asegura que ha sido PagosWeb quien envía la respuesta de la transacción.

Generar "transactionSecurityToken"

A continuación se listan los pasos a seguir para cifrar la información de la transacción:

- Concatenar información de transacción.
  La información que se debe cifrar es la misma de la transacción, concatenada en el siguiente orden sin espacios en blanco ni caracteres especiales:

  idCliente + idTarjetaCredito + primerNombre + primerApellido + segundoNombre + segundoApellido + cedula + email + telefono + direccionEnvio + valorTransaccion + cantidadCuotas + moneda + numeroOrden + version + plan + plazoEntrega + fecha + consumidorFinal + importeGravado + numeroFactura

  Es imprescindible que los valores concatenados sean idénticos a los enviados en la transacción así como el orden en que se concatenan, de lo contrario la validación de PagosWeb no dará por válido el campo de seguridad y se retornara a la página de transacción cancelada del comercio.

- Convertir información a cifrar en matriz de bytes de 7 bits ASCII. Para esto se utiliza la función GetBytes() de la clase ASCIIEncoding.

- Generar el vector de inicialización.
  El vector de inicialización es requerido por 3DES para evitar que el resultado del cifrado sea el mismo en el caso que se intente cifrar la misma información.
  PagosWeb utiliza la fecha de la transacción para generar el vector de inicialización. Por eso se requiere que la fecha enviada en el Request sea exactamente la misma que se utiliza en el cifrado.
  Para generar el vector se utilizan los datos de la fecha a partir del último dígito del año, sin espacios ni caracteres de separación y el carácter "=" al final.
  Ejemplo: dFechaTrn.ToString("yyMMddhhmmss").Substring(1) + "="

  Esto representa el vector en base 64, el cual se debe convertir en una matriz de 8 bits con la función "System.Convert.FromBase64String".

- Convertir la llave del comercio. La llave provista por PagosWeb está codificada en dígitos en base 64. Esta llave se debe convertir a una matriz de 8 bits utilizando la misma función.

- Cifrar la información con 3Des en modo CBC y codificarla en dígitos de base 64. Utilizando por ejemplo la función "System.Convert .ToBase64String"

Ejemplo de cifrado en asp.net con C#

Datos de la transacción:

```
idCliente = 3
idTarjetaCredito = 1
primerNombre = Armando
segundoNombre = Esteban
primerApellido = Quito
segundoApellido = Blanco
cedula = 1234567-9
email = unmail@dominio.com
telefono = 123456789
direccionEnvio = Av. Sin Nombre 0000
valorTransaccion = 1234.50
cantidadCuotas = 6
moneda = 858
numeroOrden = 3535
version = 3.3
```

```
        plan = 0
        plazoEntrega = 48hs
        fecha = 2012-10-2511:18:00
        consumidorFinal = 1
        importeGravado = 300
        numeroFactura = 123456


public static string GetSecurityToken()
{
    string retorno = "";
    //Fecha de la transacción = "2012-10-2511:18:00".
    DateTime dFechaTransaccion = DateTime.Now;
    //Llave 3Des del comercio.
    string keyComercio64 = "Y8LpWzag8MliH0PSXwijck+rh2wwsCi8";

    //Informacion de la transaccion a cifrar.
    string textoPlano = "31ArmandoQuitoEstebanBlanco1234567-9" +
      "unmail@dominio.com123456789Av. Sin Nombre 00001234.50685835353.3048hs" +
       dFechaTransaccion.ToString("yyyy-MM-ddhh:mm:ss") + "1300A123456";

    //Vector de incializacion= "21025111800="
    string sIV64 = dFechaTransaccion.ToString("yyMMddhhmmss").Substring(1) + "=";

    retorno = EncryptTextToMemory(textoPlano, keyComercio64, sIV64);
    // retorno=
    "M7lMF5dt13yCY2yX0ATjwgo5GnkOXx4kKXYpsVMqxL8PRm3I+vORpfJo8XplfeQLfnn3q
           J/WKw8fplY7URCGJfxXjSa7T6LLBcpQ6fuT3RUjRaH/TPrRpG5nW/DNhSSuuvYsUnU
           +2APH7vbKRcRqytxzF4QSjoCcDiFfdwu14WM="

    Return retorno;
}


Public static string EncryptTextToMemory(string Data, stringsKey, stringsIV)
{
   try
   {
      System.IO.MemoryStreammStream = new System.IO.MemoryStream();
      byte[] bKey = System.Convert.FromBase64String(sKey);
      byte[] bIV = System.Convert.FromBase64String(sIV);
      UTF8Encoding utf8 = newUTF8Encoding();
      Byte[] toEncrypt = utf8.GetBytes(Data);
      System.Security.Cryptography.CryptoStream cStream = new CryptoStream(mStream,
       new TripleDESCryptoServiceProvider() { Key = bKey, IV = bIV, Mode =
      CipherMode.CBC }.CreateEncryptor(), CryptoStreamMode.Write);

      cStream.Write(toEncrypt, 0, toEncrypt.Length);
      cStream.FlushFinalBlock();
      byte[] ret = mStream.ToArray();
      cStream.Close();
      mStream.Close();
      return System.Convert.ToBase64String(ret);
   }
   catch (CryptographicException e)
   {
      Return "Error";
```

```
    }
}
```

Ejemplo de Cifrado en PHP

```php
<?php
//Fecha de la transacción = "2012-10-2511:18:00"
$fechaTransaccion = date;
//Vector de inicializacion = "21025111800="
$iv = base64_decode(substr($fechaTransaccion ('ymdhis'),1).'=');
//datos de transaccion
$string ='31ArmandoQuitoEstebanBlanco1234567-9unmail@dominio.com123456789Av.
Sin Nombre 00001234.50685835353.3048hs'.$fechaTransaccion("Y-m-dh:i:s");

//Llave del comercio
$key = base64_decode('Y8LpWzaq8MliH0PSXwijck+rh2wwsCi8');
$encryptedString = encryptNET3DES($key,$iv,$string);
//Resultado =
"M7lMF5dt13yCY2yX0ATjwgo5GnkOXx4kKXYpsVMqxL8PRm3I+vORpfJo8XplfeQLfnn3q
J/WKw8fplY7URCGJfxXjSa7T6LLBcpQ6fuT3RUjRaH/TPrRpG5nW/DNhSSuuvYsUnU+2AP
            H7vbKRcRqytxzF4QSjoCcDiFfdwu14WM="

function encryptNET3DES($key, $vector, $text){
      $td = mcrypt_module_open(MCRYPT_3DES, '', MCRYPT_MODE_CBC, '');
      $key_add = 24 - strlen($key);
      $key .= substr($key, 0, $key_add);

      $text_add = strlen($text) % 8;
      for($i=$text_add; $i<8; $i++){
            $text .= chr(8 - $text_add);
      }

      mcrypt_generic_init($td, $key, $vector);
      $encrypt64 = mcrypt_generic($td, $text);
      mcrypt_generic_deinit($td);
      mcrypt_module_close($td);

      return base64_encode($encrypt64);
}
?>
```

Descifrar "ResponseSecurityToken"

A continuación se listan los pasos a seguir para descifrar el campo de seguridad de respuesta:

- Convertir información a descifrar en matriz de bytes en base 64.
- Generar vector de inicialización, en la respuesta se agrega el campo "fecha", representa la fecha hora en que se realizo la respuesta. Con esta fecha se debe generar el vector de inicialización, siguiendo el mismo procedimiento que se describe en el punto anterior.

- Convertir la llave del comercio y el vector de inicialización en matriz de bytes en base 64.
- Descifrar información con 3Des en modo CBC y convertir matriz resultado (bytes de 7 bits ASCII) en texto plano. Para esto se utiliza la función GetString() de la clase UTF8Encoding.
- Comprobar información de respuesta.
  La información cifrada es la misma información que contiene la respuesta concatenada de la siguiente manera:
  ventaAprobada + numeroTransaccion + monto + codigoAutorizacion +mensaje +numeroOrden + idCliente + fecha

  Nota: No se incluye en la validación del token, el campo cantidadCuotas, agregado recientemente al protocolo.

Ejemplo de descifrado en ASP.NET con C#

Datos de la respuesta:

```
Venta Aprobada: True
Id Cliente: 3
Número Transacción: 496
Monto: 1234,5
Código Autorización: 1234
Mensaje: Venta Aprobada
Número Orden: 3535
Fecha: 2012-10-2511:18:05
Security Token:
X3XwnzaIFu/xANl/oZ7bHPjuJO0YeWot4KO9bev7GAOX3cOgkEhO4YpnBEjAbXd3
Evp11aKj8Y4=
Security Token (descifrado): True4961234,51234Venta
Aprobada353532012-10-2511:18:05
```

```
public string DesCifrarSecurityToken(string securityToken, string sfecha)
{
    //sfecha = “2012-10-2511:18:05”
    //securityToken =
    “X3XwnzaIFu/xANl/oZ7bHPjuJO0YeWot4KO9bev7GAOX3cOgkEhO4YpnBEjAbXd3Evp11aKj8Y4=”

    string retorno = "";
    DateTime dFechaRespuesta;
    if (DateTime.TryParse(sfecha, out dFechaRespuesta))
    {
       string keyComercio64 = "Y8LpWzag8MliH0PSXwijck+rh2wwsCi8";
       string sIV64 = dFechaRespuesta.ToString("yyMMddhhmmss").Substring(1) + "=";
       retorno = DecryptTextFromMemory(securityToken, keyComercio64, sIV64);
       //retorno = “True4961234,51234Venta Aprobada353532012-10-2511:18:05”
    }
    return retorno;
}


public static string DecryptTextFromMemory(string sData, string sKey, string sIV)
{
   try
   {
      byte[] Data = System.Convert.FromBase64String(sData);
      byte[] bKey = System.Convert.FromBase64String(sKey);
      byte[] bIV = System.Convert.FromBase64String(sIV);
      MemoryStream msDecrypt = new MemoryStream(Data);
      CryptoStream csDecrypt = new CryptoStream(msDecrypt,
        new TripleDESCryptoServiceProvider() { Mode = CipherMode.CBC, Key = bKey, IV
        = bIV }.CreateDecryptor(),CryptoStreamMode.Read);

      byte[] fromEncrypt = new byte[Data.Length];
      csDecrypt.Read(fromEncrypt, 0, fromEncrypt.Length);
      UTF8Encoding utf8 = new UTF8Encoding();
      string decodedString = utf8.GetString(fromEncrypt);
```

```
        return decodedString;

    }
    catch (CryptographicException e)
    {
        return"Error";
    }
}
```

Ejemplo de descifrado en PHP

```
<?php
//Fecha de la respuesta = "2012-10-2511:18:05"
$fechaRespuesta = date;

//Vector de inicializacion = 21025111805=
$iv = base64_decode(substr($fechaRespuesta('ymdhis'),1).'=');

//Llave del comercio
$key = base64_decode('Y8LpWzaq8MliH0PSXwijck+rh2wwsCi8');

//Security Token de Respuesta
$strToDesc =
'X3XwnzaIFu/xANl/oZ7bHPjuJO0YeWot4KO9bev7GAOX3cOgkEhO4YpnBEjAbXd3Evp11aKj8Y4='
;

$infoRespuesta = decrypt($strToDesc,$key,$iv);
//Resultado = "True4961234,51234Venta Aprobada353532012-10-2511:18:05"

function decrypt($encrypted_text, $key, $iv)
{
        $cipher = mcrypt_module_open(MCRYPT_3DES, '', MCRYPT_MODE_CBC, '');

        mcrypt_generic_init($cipher, $key, $iv);
        $decrypted = mdecrypt_generic($cipher, base64_decode($encrypted_text));
        mcrypt_generic_deinit($cipher);

        return trim($decrypted, "\x00..\x1F");
}
?>
```