

Integración con eCommerce Bancard

Compra Simple - versión 0.3.1

[1- Introducción](#)

[Flujo de compra](#)

[Operaciones](#)

[2- Autenticación](#)

[Token](#)

[3- Operaciones](#)

[3.1 - Operaciones de infonet-vpos](#)

[3.1.1 - Single Buy](#)

[3.1.2 - Single Buy Rollback](#)

[3.2 - Operación de confirmación al portal](#)

[3.2.1 - Buy Single Confirm](#)

[3.3 - Operación de obtener confirmación](#)

[3.3.1 - Get Buy Single Confirmation](#)

[4- Redirección hacia el servicio de VPOS](#)

[5- Página de pago del usuario](#)

[6- Restricciones del comercio](#)

[Página de respuesta](#)

[Notas generales](#)

[Formato de mensajería - JSON](#)

[Monitoreo](#)

1- Introducción

El siguiente documento presenta la información técnica necesaria para comunicarse con el servicio de eCommerce de Bancard o VPOS.

Adicionalmente a este documento el comercio o desarrollador de la integración con VPOS deberá contar con un acceso al portal de comercio de Bancard:

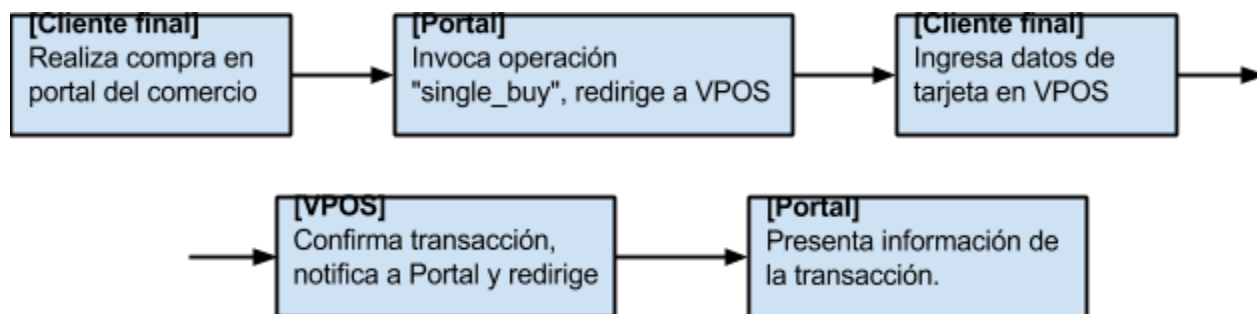
<https://comercios.bancard.com.py>

En el mismo se le brindará acceso para:

- Acceder al ambiente de staging y producción de vpos
- Acceder a la clave pública y privada. Adicionalmente podrá regenerar ambas claves.
- Modificar información del perfil: Nombre, logo y url de confirmación
- Traza de interacciones entre VPOS y el Portal desarrollado.
- Checklist con pasos para validar la integración.
- Documentación y ejemplos de códigos en distintos lenguajes.

Flujo de compra

A continuación se presenta el flujo de compra entre el cliente final, portal y VPOS.



Operaciones

El eCommerce de Bancard cuenta con tres operaciones publicadas como Web Services REST disponibles para los comercios asociados que le permitirán realizar una autorización de pago, un rollback o cancelación, y obtener información de confirmación.

El cliente debe ofrecer en una URL pública y de común acuerdo un servicio mediante el cual se notificará la aprobación o cancelación de la transacción de un cliente final.

Servicios ofrecidos por VPOS

- **single_buy** - inicia el proceso de pago.
- **single_buy_rollback** - operación que permite cancelar el pago.
- **get_single_buy_confirmation** - operación que informa si un pago fue confirmado o no.

Servicios ofrecidos por el portal del comercio

- **single_buy_confirm** - operación que será invocada por VPOS para confirmar un pago.

2- Autenticación

La clave privada y pública permitirán identificar todas las interacciones con los servicios del eCommerce de Bancard. Estas claves serán enviadas por el portal en todas sus peticiones para identificarse (la clave privada **nunca** viaja en forma plana, sino *hasheada* con otra información en forma de *token*). Ambas pueden ser generadas nuevamente en caso de vulneración.

La clave pública **será única**, y de la forma: [a-zA-Z0-9]{32}. La clave privada no tiene porqué ser única (aunque seguramente lo sea), y de la forma: [a-zA-Z0-9T1]{40}.

Peticiones realizadas por el portal a VPOS

Las peticiones serán realizadas por POST a una interfaz REST

public_key	Clave pública del portal. En el caso de la confirmación, la key enviada es también la del portal al cual se envía.
operation	Datos de la operación que se va a llevar a cabo.

```
{
  "public_key": "[public key]",
  "operation": {
    "token": "[generated token]",
    ...
  }
}
```

```
}  
}
```

Token

El token será generado al momento de realizar la petición, dependiendo de la operación. Será siempre un md5 (32 caracteres). El orden debe ser exactamente como se indica.

single buy

```
md5( private_key + shop_process_id + amount + currency)
```

single_buy confirm

```
md5( private_key + shop_process_id + "confirm" + amount + currency )
```

single_buy get confirmation

```
md5( private_key + shop_process_id + "get_confirmation" )
```

single buy rollback

```
md5( private_key + shop_process_id + "rollback" + "0.00" )
```

El token de confirm para una acción de rollback se genera usando "0.00" para amount.

Al momento de generar el token, los números deben ser transformados en cadenas, usar dos dígitos decimales y un punto (".") como separador de decimales. ej:

```
token = md5( "[private key]" + "3332134" + "130.00" + "130.00" )
```

3- Operaciones

3.1 - Operaciones de infonet-vpos

3.1.1 - Single Buy

POST {environment}/vpos/api/0.3/single_buy
Environment

- Producción - https://vpos.infonet.com.py
- Staging - https://vpos.infonet.com.py:8888

Operación invocada por el portal para iniciar el proceso de pago.

El servicio devolverá un identificador de proceso (process_id) generado temporalmente por el servidor de VPOS.

Debe completarse con éxito un Single Buy para habilitación de la correspondiente opción en la Lista de test ->Implementación de pagos

El pedido estará compuesta por un JSON con los siguientes elementos:

Elementos de la petición

public_key	Clave pública.	String(50)
operation	Elemento Operation	Operation

Elementos Operation

token	md5 de la petición	String(32)
shop_process_id	identificador de la compra.	(Entero de largo variable definible por el cliente)
amount	Importe en Guaraníes.	Decimal(15,2) - separador decimal '.'
currency	Tipo de Moneda.	String(3) - PYG (Gs)

additional_data	Campo de servicio de uso reservado para casos especiales.	String(100)
description	Descripción del pago, para mostrar al usuario.	String(100)
return_url	URL a donde se enviará al usuario al realizar el pago. Tener en cuenta que si la tarjeta es rechazada, también se le redirigirá a esta URL.	String(255)
cancel_url	URL a donde se enviará al usuario al cancelar el pago. Opcional, se usará return_url por defecto.	String(255)

Descripción de “additional_data”

Este elemento será utilizado para enviar información adicional de la compra. Se empleara para indicar promociones o convenios realizados entre el comercio, Bancard y el emisor.

La estructura de este elemento será:

- Entidad - Int(3)
- Marca - String(3)
- Producto - String(3)
- Afinidad - Int(6)
- Agrupador de afinidad - Int(3)

Ejemplos de datos válidos:

- Entidad: 017
- Marca: VS
- Producto: ORO
- Afinidad: 000045
- Agrupador de afinidad: 001

(additional_data enviada: 017VS ORO000045001)

Ejemplo petición:

```
{
  "public_key": "[public key]",
  "operation": {
    "token": "[generated token]",
    "shop_process_id": 54322,
    "currency": "PYG",
    "amount": "10330.00",
    "additional_data": "017VS OR0000045001",
    "description": "Ejemplo de pago",
    "return_url": "http://www.example.com/finish",
    "cancel_url": "http://www.example.com/cancel"
  }
}
```

La respuesta estará compuesta por un JSON con los siguientes elementos:

status	Estado de respuesta	String(20)
process_id	Identificador de la compra	String(20)

Ejemplo respuesta:

```
{
  "status": "success",
  "process_id": "i5fn*lx6niQel0QzWK1g"
}
```

Nota:

"El mensaje de respuesta se enviará en el cuerpo (body) de la petición HTTP"

3.1.2 - Single Buy Rollback

POST {environment}/vpos/api/0.3/single_buy/rollback
Environment

- Producción - <https://vpos.infonet.com.py>
- Staging - <https://vpos.infonet.com.py:8888>

La operación de Rollback deberá ser enviada en los siguientes casos:

- Para realizar un reverso de pago.
- Para transacciones canceladas por el usuario en la página de vPos.
- Para transacciones abandonadas o no culminadas por el usuario.

La operación del Rollback será satisfactoria mientras la transacción no haya sido CUPONADA (confirmada en el extracto del cliente). Si el JSON devuelve status: error y key: "TransactionAlreadyConfirmed", el comercio deberá realizar el proceso manual de pedido de reversión de una transacción cuponada a tramitar en el Área Comercial de Bancard.

El rollback devolverá un estado general "status". "success" indica que el pedido será notificado para cancelar. "error" indica que por alguna razón el pedido no puede continuar.

Las posibles causas de error son:

- **InvalidJsonError** - Error en el JSON enviado
- **UnauthorizedOperationError** - Las credenciales enviadas no tienen permiso para la operación rollback.
- **ApplicationNotFoundError** - No existen permisos para las credenciales enviadas.
- **InvalidPublicKeyError** - Existe un error sobre la clave pública enviada.
- **InvalidTokenError** - El token se generó en forma incorrecta.
- **InvalidOperationError** - El JSON enviado no es válido. No cumple con los tipos o límites definidos.
- **BuyNotFoundError** - No existe el proceso de compra seleccionado
- **PaymentNotFoundError** - No existe un pedido de pago para el proceso seleccionado. Esto quiere decir que el cliente no pagó este pedido.
- **AlreadyRollbackedError** - Ya existe un pedido de rollback previo.
- **PosCommunicationError** - Existen problemas de comunicación con el componente de petición de rollback.
- **TransactionAlreadyConfirmed** - Transacción Cuponada (Confirmada en el extracto del cliente)

En el caso de que una compra sea iniciada por el portal pero no se finalice por el usuario o no se obtenga respuesta de parte de vpos luego de 10 minutos, se debería invocar un **Get Buy Single Confirmation** para conocer el estado del pedido. Si el pago todavía no ha sido realizado, el comercio puede optar por realizar un rollback del pedido invocando a la operación **Single Buy Rollback**.

Nota:

Debe completarse con éxito un Single Buy Rollback manual en un caso de transacción aprobada para habilitación de la correspondiente opción en la Lista de test ->Recibir rollback

El pedido estará compuesta por un JSON con los siguientes elementos:

Elementos de la petición

public_key	Clave pública	String
operation	Elemento Operation	Elemento

Elemento Operation

token	md5 de la petición	String
shop_process_id	Identificador interno del portal	String

Ejemplo petición:

```
{
  "public_key": "[public key]",
  "operation": {
    "token": "[generated token]",
    "shop_process_id": "12313"
  }
}
```

La respuesta estará compuesta por un JSON con los siguientes elementos:

Elementos de la respuesta

status	Estado de respuesta	String <u>Valores posibles:</u> - success - error
messages	Array de elemento	Array

	Message	
--	---------	--

Elemento Message

key	Clave de respuesta	String <u>Valores posibles:</u> - InvalidJsonError - UnauthorizedOperationError - ApplicationNotFoundError - InvalidPublicKeyError - InvalidTokenError - InvalidOperationError - BuyNotFoundError - PaymentNotFoundError - AlreadyRollbackedError - PosCommunicationError - RollbackSuccessful - TransactionAlreadyConfirmed
level	Nivel de despliegue del mensaje	String <u>Valores posibles:</u> - info - error
dsc	Descripción de respuesta	String

Ejemplo respuesta:

<pre> { "status": "success", "messages": [{ "key": "RollbackSuccessful", "level": "info", "dsc": "Rollback correcto.", }] } </pre>
--

3.2 - Operación de confirmación al portal

3.2.1 - Buy Single Confirm

POST [URL] (Definida por el comercio)

Esta acción es invocada por VPOS al finalizar una transacción. Tiene como objetivo confirmar o cancelar un pago. Este será el único medio por el cual el portal tendrá la certeza de que el usuario completó satisfactoriamente una transacción.

Bancard realizará una petición POST enviando el JSON en el cuerpo del pedido o body. El comercio deberá responder con status 200 a la operación, como se muestra más abajo en el ejemplo de respuesta. Si el comercio no responde con status 200 dentro de los siguientes 60 segundos, vPOS cerrará la conexión y se marcará como inválida la confirmación en la traza y con una indicación del timeout en reemplazó de lo que debió ser la respuesta del comercio.

Nota:

Si el Portal del Comercio inicia la operación de compra (single buy) y no recibe la confirmación (single buy confirm) por parte del VPOS, deberá invocar a la operación de reversa (single buy rollback) para evitar inconsistencias en su sistema. El tiempo de espera recomendado es de 10 minutos.

Debe completarse con éxito un Buy Single Confirm para habilitación de la correspondiente opción en la Lista de test ->Confirmamos correctamente al comercio.

El pedido estará compuesta por un JSON con los siguientes elementos:

Elementos de la petición

operation	elemento Operation	Elemento
-----------	--------------------	----------

Elementos Operation

token	md5 de la petición	String(32)
-------	--------------------	------------

shop_process_id	identificador interno del portal	(Entero de largo variable definible por el cliente)
response	Indicador de detalle procesado	String(1) - S o N
response_details	Descripción del proceso	String(60)
amount	Importe en Guaraníes.	Decimal(15,2) - separador decimal '.'
currency	Tipo de Moneda.	String(3) - PYG (Gs)
authorization_number	Código de autorización.	String(6) - Solo si la transacción es aprobada.
ticket_number	Identificador de autorización.	Int(15)
response_code	Código de respuesta de la transacción.	String(2) - 00 (transacción aprobada) - 05 (Tarjeta inhabilitada) - 12 (Transacción inválida) - 15 (Tarjeta inválida) - 51 (Fondos insuficientes)
response_description	Descripción de la respuesta de transacción.	String(40)
extended_response_description	Descripción extendida de la respuesta de transacción.	String(100)
security_information	Elemento SecurityInformation	Elemento

Elementos SecurityInformation

card_source	Local o Internacional	String(1) - L (Local) - I (Internacional)
customer_ip	Ip del cliente que ingresa los datos de pago	String(15)
card_country	País de origen de la tarjeta	String(30)
version	Version de la API	String(5)

risk_index	Indicador de riesgo.	Int(1)
-------------------	----------------------	--------

Ejemplo petición:

```
{
  "operation": {
    "token": "[generated token]",
    "shop_process_id": "12313",
    "response": "S",
    "response_details": "respuesta S",
    "extended_response_description": "respuesta extendida",
    "currency": "PYG",
    "amount": "10100.00",
    "authorization_number": "123456",
    "ticket_number": "123456789123456",
    "response_code": "00",
    "response_description": "Transacción aprobada.",
    "security_information": {
      "customer_ip": "123.123.123.123",
      "card_source": "I",
      "card_country": "Croacia",
      "version": "0.3",
      "risk_index": "0"
    }
  }
}
```

Notas:

- El atributo de “risk_index” será parte de la respuesta pero aún no se enviará un dato válido. Momentáneamente se representará con el valor “0”.
- El JSON enviado cumple con los estándares de codificación y encoding. Ver anexo por más información.

Ejemplo respuesta:

```
{
  "status": "success"
}
```


3.3 - Operación de obtener confirmación

3.3.1 - Get Buy Single Confirmation

POST {environment}/vpos/api/0.3/single_buy/confirmations

Esta acción es invocada por el portal para consultar si existió o no una confirmación.

Debe completarse con éxito un Get Buy Single Confirm para habilitación de la correspondiente opción en la Lista de test -> Consultamos confirmación correctamente.

El pedido estará compuesta por un JSON con los siguientes elementos:

Elementos de la petición

public_key	Clave pública.	String(50)
operation	Elemento Operation	Operation

Elementos Operation

token	md5 de la petición	String(32)
shop_process_id	identificador de la compra.	(Entero de largo variable definible por el cliente)

La respuesta estará compuesta por un JSON con los siguientes elementos:

Elementos de la respuesta

status	Estado de respuesta	String <u>Valores posibles:</u> - success - error
confirmation	Informacion de confirmación	Elemento SingleBuyConfirmation
messages	Array de elemento	Array

	Message	
--	---------	--

Elemento Message

key	Clave de respuesta	String <u>Valores posibles:</u> - InvalidJsonError - UnauthorizedOperationError - ApplicationNotFoundError - BuyNotFoundError - InvalidPublicKeyError - InvalidTokenError - InvalidOperationError - PaymentNotFoundError - AlreadyRollbackedError
level	Nivel de despliegue del mensaje	String <u>Valores posibles:</u> - info - error
dsc	Descripción de respuesta	String

Elemento SingleBuyConfirmation

token	MD5 de la petición	String(32)
shop_process_id	Identificador interno del portal	(Entero de largo variable definible por el cliente)
response	Indicador de detalle procesado	String(1) - S o N
response_details	Descripción del proceso	String(60)
amount	Importe en Guaraníes.	Decimal(15,2) - separador decimal ‘.’
currency	Tipo de Moneda.	String(3)

		- PYG (Gs)
authorization_number	Código de autorización.	String(6) - Solo si la transacción es aprobada.
ticket_number	Identificador de autorización.	Int(15)
response_code	Código de respuesta de la transacción.	String(2) - 00 (transacción aprobada) - 05 (Tarjeta inhabilitada) - 12 (Transacción inválida) - 15 (Tarjeta inválida) - 51 (Fondos insuficientes)
response_description	Descripción de la respuesta de transacción.	String(40)
extended_response_description	Descripción extendida de la respuesta de transacción.	String(100)
security_information	Elemento SecurityInformation	Elemento

Elementos SecurityInformation

card_source	Local o Internacional	String(1) - L (Local) - I (Internacional)
customer_ip	Ip del cliente que ingresa los datos de pago	String(15)
card_country	País de origen de la tarjeta	String(30)
version	Version de la API	String(5)
risk_index	Indicador de riesgo.	Int(1)

Ejemplo petición:

<pre>{ "public_key": "[public key]", "operation": {</pre>

```
    "token": "[generated token]",
    "shop_process_id": "12313"
  }
}
```

Ejemplo respuesta:

```
{
  "status": "success"
  "confirmation": {
    "token": "[generated token]",
    "shop_process_id": "12313",
    "response": "S",
    "response_details": "respuesta S",
    "extended_response_description": "respuesta extendida",
    "currency": "PYG",
    "amount": "10100.00",
    "authorization_number": "123456",
    "ticket_number": "123456789123456",
    "response_code": "00",
    "response_description": "Transacción aprobada.",
    "security_information": {
      "customer_ip": "123.123.123.123",
      "card_source": "I",
      "card_country": "Croacia",
      "version": "0.3",
      "risk_index": "0"
    }
  }
}
```

Notas:

- El atributo de “risk_index” será parte de la respuesta pero aún no se enviará un dato válido. Momentáneamente se representará con el valor “0”.
- El JSON enviado cumple con los estándares de codificación y encoding. Ver anexo por más información.

4- Redirección hacia el servicio de VPOS

GET {environment}/payment/single_buy

Environment

- Producción - <https://vpos.infonet.com.py>
- Staging - <https://vpos.infonet.com.py:8888>

Luego de llamar a la acción “single_buy”, el portal deberá redireccionar al usuario a VPOS.

Parámetros

process_id	Identificador devuelto por la acción “single_buy”
-------------------	---

ejemplo:

https://vpos.infonet.com.py/payment/single_buy?process_id=1veY2NMX3Od751Lu0sK5

5- Página de pago del usuario



**Bienvenido, completá éstos
datos y finalizá tu compra**

Tu compra

Conseguí un afeitado... Gs. 748.000

Ingresá los datos de tu Tarjeta



Nro. de Tarjeta

Código de Seguridad

Fecha de Expiración

01

/

2013

Cuotas (Válido sólo para Paraguay)

No

Confirmar compra

Cancelar



Todos los derechos reservados. Pago seguro © 2012 - un servicio de Bancard

6- Restricciones del comercio

A continuación se presentan restricciones que debe contemplar el comercio que desarrolle la integración con el eCommerce de Bancard.

Página de respuesta

Luego de que el usuario ingresa sus datos de tarjeta y se confirma al comercio por medio de la operación "Buy Single Confirm" el comercio debe desplegar una página de respuesta con la aprobación de la transacción.

En esta página se deben respetar las siguientes restricciones:

- Se deben indicar los datos de la transacción:
 - Fecha y Hora
 - Número de pedido (shop_process_id)
 - Importe (amount)

Descripción de la Respuesta (response_description)
Código de autorización si fue aprobada (authorization_number)

- **No** debe mostrarse al usuario:
 - Código de respuesta (response_code)
 - Respuesta extendida (extended_response_description)
 - Información de seguridad (security_information)

Notas generales

- El Comercio debe incluir en su aplicación la sección de **Contacto**, de manera que el cliente pueda evacuar consultas referentes a las compras del ecommerce.
- La aplicación podrá registrar todos datos del cliente que requiera el comercio, salvo todos aquellos que se relacionen a sus tarjetas de crédito (Número de tarjeta, código de seguridad, vencimiento, etc.)
- El logo a ser utilizado por el comercio debe idealmente tener un ancho de 173 píxeles y un alto 55 píxeles. El ancho mínimo es de 85 píxeles.

Formato de mensajería - JSON

Para enviar y recibir información se empleará el formato JSON (JavaScript Object Notation).

Información sobre JSON - <http://json.org/>

Validación de JSON - <http://jsonlint.com/>

Al consumir un JSON enviado por el VPOS debe prestarse especial atención a los caracteres especiales (ej. tildes). El mismo será enviado utilizando el standard “\uXXXX” (Donde X es un dígito hexadecimales)

Los JSON enviados y recibidos por Bancard y el comercio deberán realizarse mediante una petición POST enviando el JSON en el cuerpo del pedido o body.

Monitoreo

Para cumplir con nuestros estándares de calidad el servicio del cliente será monitoreado para analizar el uptime.

Cada 5 minutos se realizará una petición post a la url de confirmación con datos un JSON vacío.