

## Payment Gateway – Manual de Integración

Documento informativo de primer nivel para desarrolladores y usuarios intermedios del servicio.

Fecha: 26/10/2015 Versión: 3.5.1-r – pgdesa@provincianet.com.ar

[Payment Gateway]

[Versión 3.5.1-r]

# Manual de Integración v3.5.1-r

Módulo: Botón de Pago

(Documento técnico-funcional para  
desarrolladores clientes del aplicativo)

## Control de cambios del documento

Versión	Fecha	Autor	Descripción
3.1.1-r	2015-05-18	Tupone, Alberto L.	Release candidate
3.5.1-r	2015-10-26	Tupone, Alberto L.	Actualización de aplicación a: 3.5. Versión inicial del documento para la presente versión del aplicativo.

# Índice

1	Introducción.....	5
1.1	Propósito.....	5
1.2	Alcance.....	5
1.3	Versionado .....	5
1.4	Términos y Definiciones.....	5
1.5	Acrónimos.....	5
2	Responsables.....	6
2.1	Contacto técnico y funcional .....	6
2.2	Documentación Online.....	6
3	Proceso funcional .....	7
3.1	Esquema de arquitectura de servicios.....	7
3.2	Sinopsis.....	7
3.3	Fase de requerimiento y carga.....	7
3.4	Fase de cierre y finalización de transacción .....	8
4	Información Técnica.....	10
4.1	Componentes.....	10
4.2	Endpoints .....	11
4.2.1	URLs por entorno .....	11
4.3	Protocolo y Error Handling .....	11
4.4	Requerimiento .....	11
4.4.1	Protocolo.....	11
4.4.2	Formatos .....	12
4.4.3	Parámetros .....	13
4.4.4	Definición de parámetros específicos .....	14
4.4.5	Valores predefinidos de ciertos parámetros.....	16
4.4.6	Validación de EPC (Electronic Payment Code).....	17
4.5	Respuesta.....	18
4.5.1	Protocolo.....	18
4.5.2	Métodos de respuesta.....	18
4.5.3	Parámetros .....	18
4.5.4	Códigos de Respuesta .....	19
5	Implementación Mobile .....	21
5.1	Envío de requerimiento de transacción.....	21

# Manual de Integración - Payment Gateway v 3.5.1-r

---

- 5.2 Retorno de control a la aplicación..... 22
- 5.3 Obtención de los datos de la transacción. .... 23
- 6 Anulaciones..... 24
  - 6.1 Disposición ..... 24
- 7 Servicios Informativos..... 25
  - 7.1 Servicio de Transacciones..... 25
  - 7.2 Servicio de Informes ..... 25
- 8 Ayuda para desarrolladores..... 26
- 9 Relativos o Dependencias ..... 26

# 1 Introducción

## 1.1 Propósito

Este documento informativo de aspecto funcional y técnico está orientado a todos los consumidores del servicio general de Payment Gateway.

## 1.2 Alcance

Documento técnico funcional de alcance semipúblico, para los siguientes roles (ya sean internos o externos a Provincia Net): desarrolladores; analistas técnicos; analistas funcionales; arquitectos; líderes de equipo de desarrollo; líderes de equipos de testeo; líderes técnicos.

## 1.3 Versionado

A partir de la versión 3.1 del servicio de Payment Gateway, el versionado de la presente documentación se asocia directamente con el versionado del servicio para evitar confusiones.

**Ejemplo:**

Versión mayor	Versión menor	Versión de documentación	Etapas de desarrollo
3	5	<b>1</b>	-r (Commercial Distribution)

## 1.4 Términos y Definiciones

El siguiente es un diccionario de términos utilizados a lo largo de este documento y su explicación:

Término	Explicación
Producto	Tarjeta o medio de pago conforme, en general VISA, MASTER, AMEX
Cliente	Comercio y/o sistema de comercialización que utiliza el Payment Gateway.
Usuario	Usuario final – Tarjetahabiente - Comprador que utiliza un sistema cliente.

## 1.5 Acrónimos

El siguiente es un diccionario de acrónimos utilizados a lo largo de este documento y su significado:

Acrónimo	Significado
TBD	To Be Define (A definir)
PG	Payment Gateway per sé
EPC	Electronic Payment Code – Código de pago electrónico, identificador único de la transacción

## 2 Responsables

Rol	Recurso
Redactor	Alberto L. Tupone
Project Manager	Mariano Lambolla
Manager	Pablo G. Araguez
Responsable Técnico	Alberto L. Tupone
Responsable Técnico Mobile	Pablo Roco, Alberto L. Tupone

### 2.1 Contacto técnico y funcional

Se recomienda que para una respuesta más rápida y eficiente, se contacte a la dirección de correo electrónico:

[pgdesa@provincianet.com.ar](mailto:pgdesa@provincianet.com.ar)

O se comuniquen al teléfono **[54] (221) 423-5731**

Rol	Contacto
Soporte técnico	Alberto L. Tupone – Mariano Lambolla Provincia NET – Edificio La Plata – La Plata, Argentina [54] (221) 423-5731 <a href="mailto:pgdesa@provincianet.com.ar">pgdesa@provincianet.com.ar</a>

### 2.2 Documentación Online

Recuerde que este documento es actualizado con regularidad. Recomendamos siempre bajar y/o chequear la última versión ubicada en:

<https://test-pgmainservices.provincianet.com.ar/help>

Asimismo, la documentación técnicas se encuentra presenta en forma online en el mismo link, incluyendo el simulador. Consulte los datos crediticios necesarios para realizar pruebas con el equipo de desarrollo, a:

[pgdesa@provincianet.com.ar](mailto:pgdesa@provincianet.com.ar)

## 3 Proceso funcional

El servicio de Payment Gateway brinda la funcionalidad necesaria para realizar pagos y cobranzas en forma online con diferentes medios de pago; en forma segura y fiable. Además, para la utilización de versiones superiores a la 3.0, es un proceso capaz de ser configurado en forma responsiva, es decir, adaptable a cualquier canal (dispositivo o tecnología). Los desarrolladores del comercio deben realizar la solicitud de la transacción y la obtención de datos, mientras que la carga de datos crediticios, se realiza desde Payment Gateway.

### 3.1 Esquema de arquitectura de servicios

El servicio de Payment Gateway cuenta con varios subproductos. El más relevante de los mismos es el llamado **"Botón de Pagos"** por el cual acceden las solicitudes de transacciones crediticias. Además, el servicio cuenta un módulo de Anulaciones, Información, y Web de Quejas y Reclamos. Para obtener más información sobre los módulos anteriormente citados consulte a la sección de Comerciales y/o Funcionales de Provincia Net.

### 3.2 Sinopsis

El proceso de Payment Gateway puede ser dividido en tres etapas generales, de las cuales el destinatario de este documento deberá intervenir en la primera y tercera. Estas tres fases son:

1. **Fase de Requerimiento y Carga** (envío de la transacción a Payment Gateway)
2. **Fase de Carga** (realización de la transacción por parte del usuario)
3. **Fase de Cierre de Transacción** (recibo de la transacción a Payment Gateway)

### 3.3 Fase de requerimiento y carga

El proceso comienza con la fase de requerimiento, donde la aplicación del comercio realiza un envío de datos por HTTP a través de POST a la dirección URL del servicio de Payment Gateway (en adelante, PG). Consulte la especificación técnica para realizar esta solicitud, ya que **NO DEBE REALIZARSE como un submit común**. Este envío de datos es un conjunto de valores llamado transacción, (ver *Parámetros de Envío*).



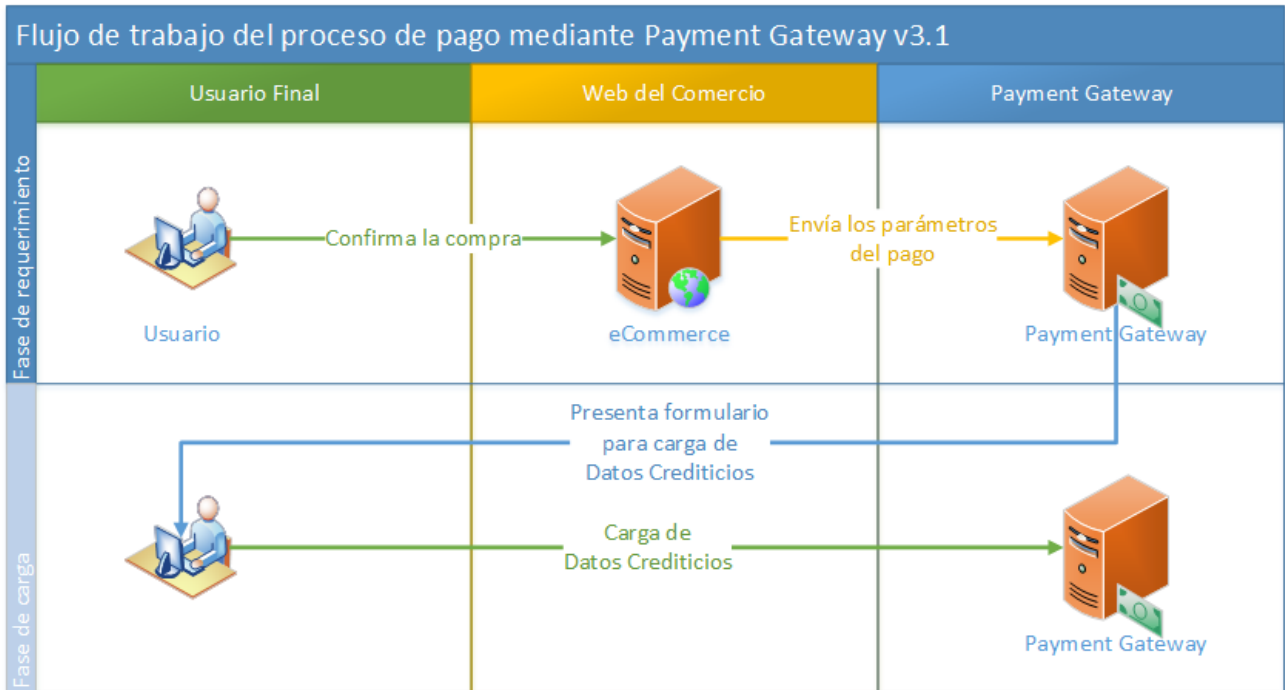
Se recuerda que los datos de la transacción, como la tarjeta a utilizarse; el monto total; el detalle de la compra; etc., deben de ser previamente solicitados al usuario final por la aplicación comercial.

El servicio de Payment Gateway no es un e-Commerce, sino un servicio de operaciones crediticias.

Provincia Net cuenta con otro sistema de e-Commerce para tal fin. Consulte.



El servicio de PG validará los datos obtenidos y, en caso de que sean válidos, presentará al cliente una pantalla donde éste podrá ingresar sus datos de comprador (Datos personales, número de tarjeta, etc.)

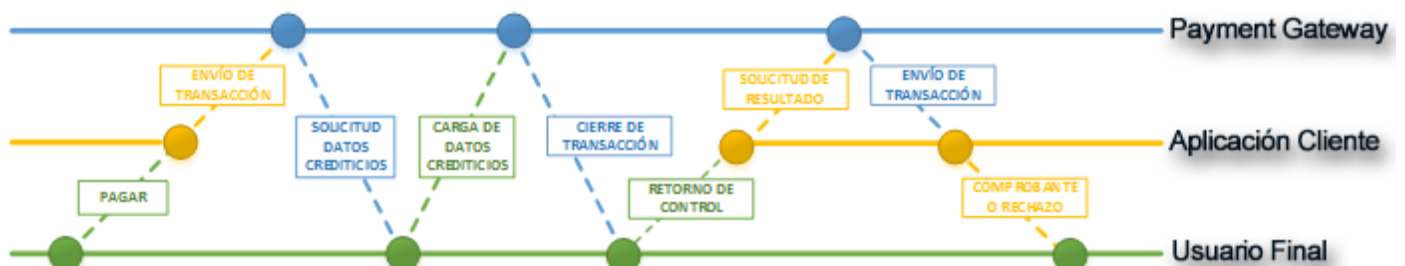


### 3.4 Fase de cierre y finalización de transacción

Una vez cargados los datos crediticios, la transacción puede tener varios resultados. En cualquier caso, ya sea una transacción aprobada o rechazada, es conveniente aclarar que el proceso debe finalizarse por ambas partes comprometidas (comercio y cliente).

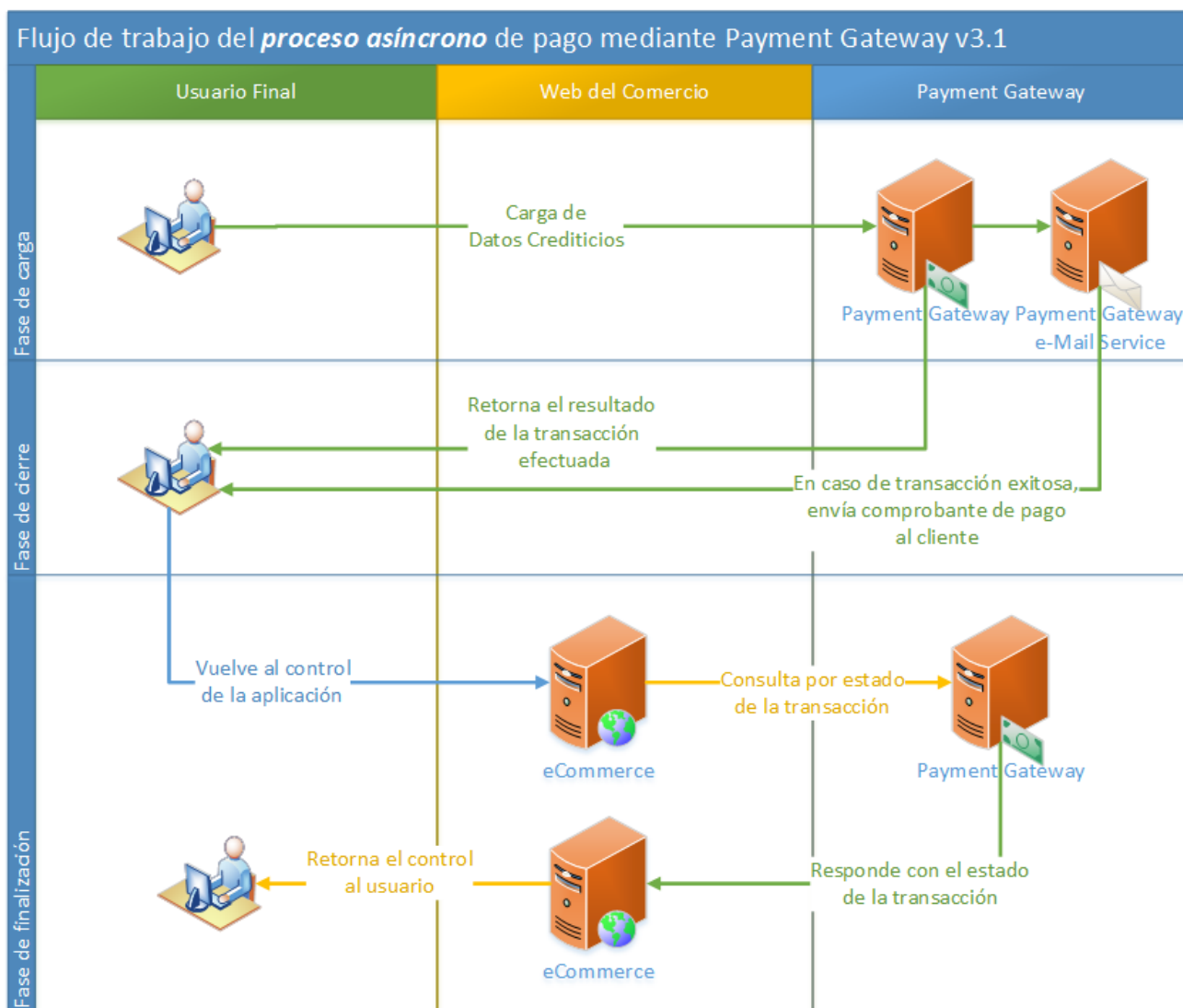
El cierre del circuito induce que la transacción ha ocurrido sin errores (***un rechazo no es un error***). Cabe destacar que, por cuestiones de seguridad y en cualquier caso, el usuario recibirá una notificación por parte de Payment Gateway sobre el resultado de esta transacción. Asimismo, en caso de ser una transacción aprobada, recibirá el comprobante de pago correspondiente.

Una vez retornado el control a la aplicación cliente, ésta se encargará de solicitar el resultado/estado de la transacción que ha realizado en paso previo. Este proceso es asíncrono (o pasivo) por razones de seguridad y estabilidad del sistema.





La solicitud para recibir los resultados de la transacción se realiza mediante otra invocación a **un servicio de consulta RESTful** en JSON, XML o URL ENCODED, para obtener las transacciones según parámetros indicados (Código de pago electrónico e identificación del comercio). Este servicio también puede ser consultado para validar/comprobar estados de transacciones pasadas o que han sido anuladas a posteriori.

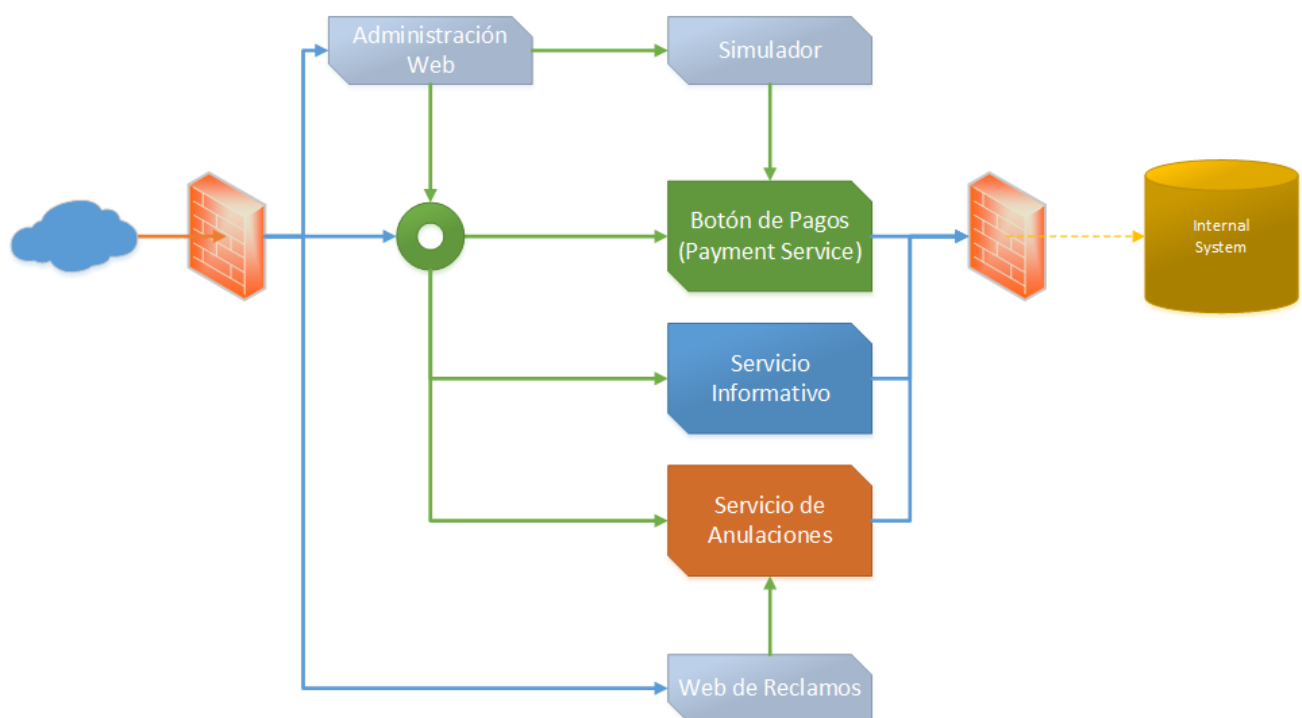


## 4 Información Técnica

El servicio de Payment Gateway es un servicio RESTful autodocumentado, por lo cual toda la ayuda técnica necesaria puede ser accedida desde cualquier momento a través del entorno de test apuntando directamente al URL principal. **(La ayuda NO se encuentra disponible en el ambiente de producción)**

### 4.1 Componentes

Algunos componentes pueden variar según el ambiente y contrato con el cliente (por seguridad varios componentes se encuentran desactivados en producción)



Componente	Descripción
Botón de pago	Servicio para realizar pagos
Servicio Informativo	Servicio para obtener información de transacciones y anulaciones.
Servicio de Anulaciones	Servicio para anulación/devolución de pagos.
Administración Web	Interfaz web para administración de diferentes tareas.
- Simulador	Parte de la administración web. Interfaz gráfica para simular pagos.
Web de Reclamos	Interfaz web para carga, autorización y proceso de reclamos de anulaciones.

## 4.2 Endpoints

### 4.2.1 URLs por entorno

Se deben utilizar los siguientes URLs según ambiente especificado.

Entorno	URL
Sandbox	<a href="https://test-pgmainservices.provincianet.com.ar/">https://test-pgmainservices.provincianet.com.ar/</a>
Producción	<a href="https://pgmainservices.provincianet.com.ar/">https://pgmainservices.provincianet.com.ar/</a>

## 4.3 Protocolo y Error Handling

El protocolo de comunicación para toda la API es HTTPS. Por este motivo, todos los mensajes (ACK, NACK, Status, Response, Error, etc.) son manejados con sus códigos nativos para facilitar el desarrollo del cliente. Estos códigos se encuentran definidos en el [RFC 2616](#).

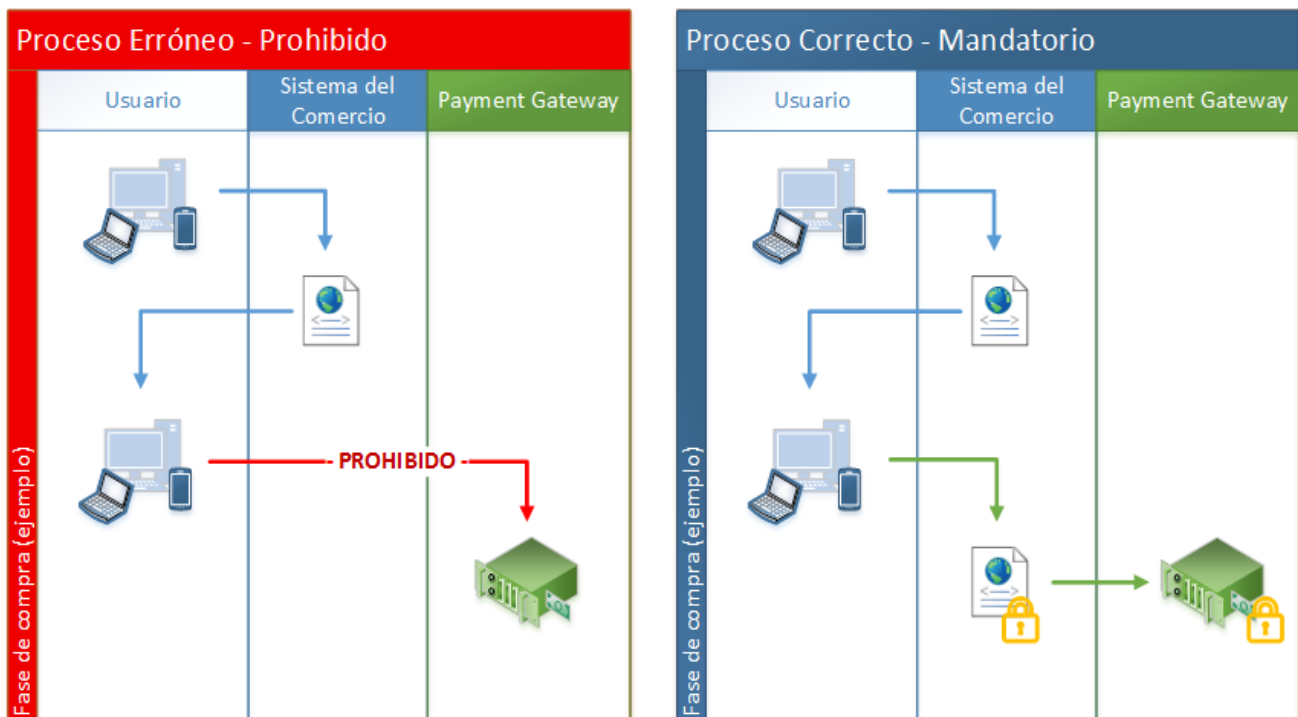
**RFC 2616:** <http://www.w3.org/Protocols/rfc2616/rfc2616-sec10.html>

## 4.4 Requerimiento

### 4.4.1 Protocolo

El requerimiento de la transacción se realiza por HTTP POST a la URLs especificadas. Las mismas están disponibles en la ayuda online en la dirección brindada en el punto [URLs por entorno](#).

El método mandatorio para realizar cualquier consulta siempre será desde el servidor, para ocultar el envío de los parámetros, utilizando cualquier librería de cliente de HTTP (cURL, HttpClient, WebClient, etc) disponible para el lenguaje que esté utilizando.



## 4.4.2 Formatos

Los parámetros del requerimiento pueden ser enviados en varios formatos. Para especificar el formato de la solicitud recuerde cambiar el atributo "Content-Type" de su requerimiento HTTP POST. Los formatos disponibles son:

Formato	Ejemplo
<b>URL ENCODED (Común)</b> application/x-www-form-urlencoded	variable1=valor1&variable2=valor2
<b>XML</b> application/xml, text/xml	<pre>&lt;PaymentInput xmlns:i="http://www.w3.org/2001/XMLSchema-instance" xmlns="http://schemas.datacontract.org/2004/07/PGMainService.Models"&gt;   &lt;variable1&gt; valor1 &lt;/variable1&gt; &lt;variable2&gt; valor2 &lt;/variable2&gt; &lt;/PaymentInput&gt;</pre>
<b>JSON (Recomendado)</b> application/json, text/json	<pre>{   "variable1": "valor1",   "variable2": "valor2" }</pre>

### 4.4.3 Parámetros

Este requerimiento debe de ser enviado con los parámetros de entrada que se encuentran detallados a continuación. Recuerde siempre que, toda la documentación aquí disponible, se encuentra actualizada al día en la auto-documentación del servicio, y se recomienda su uso anteriormente al presente documento.

Más información en: <https://test-pgmainservices.provincianet.com.ar/Help/Api/POST-Payment>

Nombre	Descripción	Tipo	Info. Adicional	Ejemplo
MerchantId	Identificación del comercio. Este dato es provisto al comienzo del proceso.	cadena	<b>Requerido</b>	TiendaDeEjemplo
ElectronicPaymentCode	Código del Pago Electrónico (EPC) que identifica al pago (o factura) del cliente. Este campo es devuelto por el proceso de Callback.	cadena	<b>Requerido</b>	7000142114464886
CurrencyCode	Código internacional ISO 4217 de la moneda de pago	cadena	<b>Requerido</b>	Requerido
Amount	Importe total del pago. Máximo 10 dígitos en la parte entera y 2 dígitos fijos para la parte fraccionaria; separados por coma.	cadena	<b>Requerido</b> Patrón de RegEx: <code>^\d+(, \.)\d{1,2})?\$</code>	123,00 ó 123,45
ProductId	Identificador único de tarjeta de crédito de Provincia Net (Ver Información, Listado de Tarjetas).	entero	<b>Requerido</b>	1 (American Express), 4 (Mastercard), 8 (VISA)
Payments	Cantidad de cuotas a realizarse en el pago.	entero	<b>Requerido</b>	1 ó 4 ó 12 (etc.)
MailAddress	Dirección de correo electrónico del usuario final (se enviará el comprobante al mismo).	cadena	<b>Requerido</b> Patrón de RegEx: <code>^[A-Za-z0-9._%+-]+@[A-Za-z0-9.-]+\.[A-Za-z]{2,4}\$</code>	<a href="mailto:consumidor@cliente.com">consumidor@cliente.com</a>
CallbackUrl	Dirección de retorno para control de aplicación. Una vez concluida la operación se redireccionará a través de un POST los resultados de la transacción a este URL especificado.	cadena	<b>Requerido</b>	<a href="http://app/procesarResultado">http://app/procesarResultado</a>
ValidateEPCreturnUrl	Dirección URL a donde se volverá automáticamente si el ElectronicPaymentCode está repetido. Si es vacío o nulo, no se valida el Electronic Payment Code.	cadena	<b>Requerido</b>	<a href="http://aplicación/regenerarEPC">http://aplicación/regenerarEPC</a>
BarCode	Código de barras de la factura, sticker o transacción, hasta 60 dígitos.	cadena		83576108935698
Channel	Especifica el canal para considerar opciones de interfaz responsive. Por defecto, Web.	cadena	<b>Requerido</b>	web o mobile
ClientLanguage	Código internacional ISO 639-1 que se utilizará para mostrar el sitio. Por defecto, "es"	cadena	Por defecto, "es" Se recomienda enviarlo en el header	es (español), en (inglés), pt (portugués)
Metadata	Representación string de un JSON que almacena cualquier información. La información de este campo es devuelto por el callback.	cadena	<b>Requerido</b>	Ver Commerce Items

#### 4.4.4 Definición de parámetros específicos

Algunos de los parámetros precisan una definición de formato específico. Estos parámetros son:

##### 4.4.4.1 Commerce Items

Este parámetro **es requerido** para ser procesado por diferentes servicios encargados de reportería y conciliación crediticia. El mismo es el detalle de los items asociados a la transacción; de modo que puedan relacionarse más tarde los detalles de la transacción con la cancelación de la deuda.

El mismo es un string (cadena de texto) que representa un objeto JSON (Javascript Object Notation - <http://www.w3schools.com/json/> ). Este objeto contiene un array de elementos (Commerceltem) con tres atributos: Code, Description y Price. El contenedor puede contener n elementos, uno o más. Puede probar la validez del mismo usando un validador como: <http://jsonlint.com/>

- **Code:** Código de barras identificador del producto. No está relacionado al código de barras de la transacción, el cual identifica a la transacción. Puede ser, por ejemplo, el código de barras propio de una deuda, o impuesto. En los casos en que cada producto tenga su correspondiente código unívoco, es preferible que se procese de esta manera. En los casos en donde se especifiquen varios productos que no posean un código unívoco, sería ideal que se generara uno, pero el sistema no requiere que los códigos sean irrepetibles.
- **Description:** Descripción del elemento (meramente informativo)
- **Price:** Precio del elemento sin comas y/o separador de miles con punto flotante de dos decimales (Ejemplo: 12345.78).
  - **Este monto NO puede ser negativo.**
  - **La suma de todos los atributos "Price" de todos los elementos, debe coincidir con el total de la transacción.**
  - **Si es necesario realizar un ajuste de tipo descuento, recargo o bonificación, tenga en cuenta que deben ser realizados antes del envío del pago.**

##### Ejemplo del dato con dos elementos:

```
{
  "Commerceltems": [
    {
      "Code": " ABCD1234",
      "Description": "PAGO DE IMPUESTO AUTOMOTOR ",
      "Price": "40.00"
    },
    {
      "Code": "00060372691000184590105215084150400001412000013611",
      "Description": "Nombre del producto que usted quiera probar",
      "Price": "1234.56"
    }
  ]
}
```



#### 4.4.5 Valores predefinidos de ciertos parámetros

Algunos de los parámetros precisan valores que se encuentran predefinidos por el sistema. Se recomienda mapear estos valores en una tabla hash / base de datos / sistema de almacenamiento de manera que puedan ser actualizados correspondientemente. Los parámetros bajo estas regulaciones son:

##### 4.4.5.1 Listado de Tarjetas Crediticias (ProductId)

Estos son los productos (tarjetas de crédito) disponibles en el sistema. Consulte disponibilidad. Recuerde que solo podrá emplear los valores de las tarjetas que fueron solicitadas y aprobadas en el proceso administrativo (ver punto 4). Tenga en cuenta que

Valor	Descripción
1	American Express
2	Diners
3	JCB
4	Mastercard
5	Cabal
6	Naranja
7	Kadicard
8	Visa
9	Nevada
10	Visa Naranja
11	Tarjeta Shopping
12	Italcared
13	Mas
14	Naranja MO
15	Pyme Nación
16	Visa Débito
17	Nativa
18	Argencard
19	Consumax
20	Discover
21	Elo
22	Magna
23	Aura
24	Hipercard
25	Credencial Col
26	AR RapiPago

Valor	Descripción
27	AR PagoFacil
28	AR BaproPagos
29	AR Ripsa
30	AR Cobro Express
31	CL ServiPag
32	BR Boleto Itaú
33	MX Seven Eleven
34	MX Oxxo
35	MX Bancomer
36	MX Santander
37	MX IXE
38	MX Scotiabank
39	PagoMisCuentas
40	SafetyPay
41	Favacard
42	Lider
43	Credimas
44	Club La Nacion
45	Club Personal
46	Club Arnet
47	Clarín 365
48	Club Speedy
49	Club La Voz
50	Nexo
51	Coopeplus
52	Arcash



#### 4.4.5.2 Listado de Canales (Channel)

Estos son los canales disponibles por el momento

Valor	Descripción
MOBILE	Canal Responsivo – Celular o Tablet
WEB	Canal básico (por defecto). Aplicación o Servicio web.

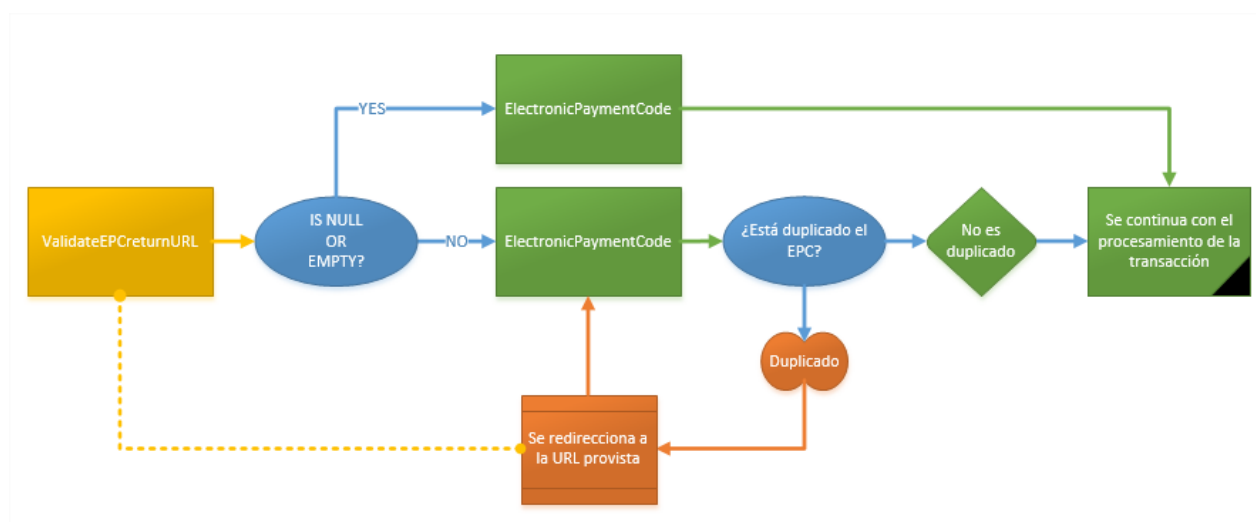
#### 4.4.5.3 Listados especificados por ISO

Estos listados se encuentran especificados en normativas internacionales vigentes que pueden ser consultadas en cualquier momento.

Valor	Descripción
CurrencyCode	ISO 4217 <a href="http://www.iso.org/iso/home/standards/currency_codes.htm">http://www.iso.org/iso/home/standards/currency_codes.htm</a>
Language	ISO 639-2 <a href="http://www.iso.org/iso/home/standards/language_codes.htm">http://www.iso.org/iso/home/standards/language_codes.htm</a>

#### 4.4.6 Validación de EPC (Electronic Payment Code)

Es posible (y recomendable) validar el Electronic Payment Code generado por el comercio. De esta manera el comercio se asegura de que Payment Gateway solamente aceptare un único código electrónico de pago por comercio. Para que esta verificación se realice, debe enviarse el parámetro opcional "**ValidateEPCreturnUrl**". Si el parámetro no se envía, o se envía una cadena vacía (empty), el EPC no se valida. Si el parámetro tiene algún valor, se valida que el valor "ElectronicPaymentCode" no se haya enviado anteriormente. De no existir una transacción anterior con el EPC especificado, el proceso sigue con normalidad. De existir una transacción (cualquiera sea su estado) con el mismo EPC, el proceso **redireccionará** a la URL especificada en el parámetro "ValidateEPCreturnUrl". Esta metodología se encuentra diseñada para que el comercio pueda retomar el control de la aplicación y generar un nuevo EPC diferente al repetido.



## 4.5 Respuesta

### 4.5.1 Protocolo

La respuesta en modo asíncrono, es la respuesta de un HTTPS a través de GET con los mismos parámetros, que contienen los valores de la transacción realizada.

### 4.5.2 Métodos de respuesta

#### 4.5.2.1 Por GET

Este método se realiza consultando al controlador "Transaction" con el ElectronicPaymentCode y el MerchantId (ver Requerimiento). Existen otros métodos que devuelven información complementaria sobre la transacción, los cuales pueden ser consultados en la sección de servicios informativos.

Ejemplo:

<https://test-pgmainservices.provincianet.com.ar/>

Transaction?electronicPaymentCode=7000142114464886&merchantCode=TiendaDeEjemplo

### 4.5.3 Parámetros

Los parámetros devueltos son los mismos para cualquier método de recepción. Los mismos son:

Nombre	Descripción	Info. Adicional	Ejemplo
ResponseCode	Código del resultado de la transacción	Específico. Se recomienda utilizar este código como parámetro de manejo. <a href="#">Ver tabla de códigos de respuesta</a>	2
ResponseMessage	Mensaje del resultado de la transacción.	Descriptivo. <a href="#">Ver tabla de códigos de respuesta</a>	TRANSACCIÓN APROBADA
ResponseExtended	Código interno del resultado de la transacción	Informativo. Se recomienda no utilizarlo como parámetro ni mostrarlo al usuario.	MOP GPAY: -1 - Aprobada
ElectronicPaymentCode	Código de Pago Electrónico	Identificación de la transacción que fue enviado en el momento del requerimiento	31463461346
TransactionId	Identificador único de la transacción	Identificador de Payment Gateway	
MetaData	Representación string de un JSON que almacena cualquier información. La información de este campo es devuelto por el callback.	Dato que fue enviado en el momento del requerimiento, provisto "AS IS".	Ver Requerimiento
ResponseGenericCode	Código del tipo del mensaje	Específico. Se recomienda utilizar este código como parámetro de evento. <a href="#">Ver tabla de códigos de respuesta</a>	1
ResponseGenericCodeMessage	Mensaje del tipo del mensaje	Descriptivo. <a href="#">Ver tabla de códigos de respuesta</a>	OK

#### 4.5.4 Códigos de Respuesta

Los códigos de respuesta son los especificados a continuación. Se debe mapear estos códigos desde el lado del cliente y considerar como identificador único el código numérico. Estos mensajes varían conforme el Language especificado. Además se encuentran agrupados en 4 tipos genéricos, para su mejor identificación ante un evento, del tipo ACK/NACK

Código Genérico	Mensaje corto	Descripción
<b>1</b>	<b>OK</b>	ACK – Solicitud aceptada / aprobada / acción correcta
<b>2</b>	<b>NO</b>	NACK – Solicitud rechazada / denegada / acción incorrecta
<b>3</b>	<b>ERROR</b>	ERR – Error en el proceso / Excepción del sistema
<b>4</b>	<b>NA</b>	NA – Error indefinido / Solicitud sin respuesta / Respuesta desconocida o en duda / Intente nuevamente

Código	Mensaje	Tipo
1	Indefinido	NA
2	Transacción aprobada	OK
3	Transacción rechazada	NO
4	Transacción reversada	OK
5	Transacción reversada y aprobada	OK
6	Límite de tarjeta excedido	NO
7	Tarjeta vencida	NO
8	Código de seguridad inválido	NO
9	Tarjeta inválida	NO
10	Error en el proceso	NO
11	Pendiente de pago (Cupón de pago emitido)	NO
12	Pendiente de pago (alta de factura y alta de adhesión efectuadas)	NO
13	Secret Key cambiada satisfactoriamente	OK
14	Respuesta pendiente del adquirente	NO
15	Factura vencida	NO
16	Renovación de PAN o fecha de expiración	NO
17	Transacción esperando datos crediticios	OK
18	Solicitud extiosa	OK
19	Sospecha de fraude	NO
20	Transacción en progreso	OK
101	Reclamo abierto	OK
102	Reclamo aprobado para anular	OK
103	Reclamo rechazado para anular	NO
104	Reclamo enviado al validador	OK
105	Reclamo enviado al validador sin respuesta (Time Out)	NA
106	Operación de anulación/devolución aprobada	OK
107	Operación de anulación/devolución rechazada	NO
108	La transacción ya ha sido anulada/devuelta con anterioridad	OK
109	Reclamo cancelado	NO

110	Error en proceso de reclamo	ERROR
111	Error no definido en proceso de reclamo	NA
210	Transacción anulada	NO
200	Transacción de pago creada	NO
201	Transacción enviada al validador	NO
202	Error al enviar la transacción al validador	ERROR
203	Error no definido al enviar la transacción al validador	NA
204	Tiempo de espera agotado al intentar comunicarse con el validador	NA
112	Reclamo bloqueado	NO
301	En progreso - En comprador	NA
302	En progreso - Transacción en progreso	NA
303	En progreso - En comercio	NA
304	Tarjeta no autorizada	NO

## 5 Implementación Mobile

La presente es un ejemplo de cómo implementar Payment Gateway en Phonegap. El código siguiente asume conocimientos básicos de HTML y JAVASCRIPT. Se encuentra disponible un ejemplo de código (POC o prueba de concepto) realizado en AngularJS y Ionic para Cordova 3.5, en plataforma Android, pero el presente código puede ser utilizado en cualquier configuración.

Para implementaciones nativas (Android SDK, no uso de Phonegap o Cordova) se recomienda la utilización de clientes y servicios propios con el proceso síncrono, o toma de datos en servicios cerrados (por ejemplo, servicios privados del comercio) con procesos asíncronos por razones de performance y escalabilidad del sistema.

### 5.1 Envío de requerimiento de transacción

El envío del requerimiento se realiza en su forma habitual, considerando lo visto en los puntos anteriores. La única salvedad sobre el presente, es que debe realizarse apuntando a un target particular, ya sea un iframe o un marco similar provisto por un plugin (como InAppBrowser), para que se puedan mostrar las pantallas de carga de los datos crediticios. Ejemplo del cabecal de un formulario:

```
<form
  name="nombre"
  id="identificador"
  action=https://test-pgmainservices.provincianet.com.ar/Payment
  method="POST"
  target="my_frame"
  accept-charset="UTF-8">
```

Dado el ejemplo del iframe, debería indicarse el contenedor que obtendrá el resultado del posteo del formulario anterior.

```
<iframe name="my_frame" style="width:100%;height:90%"></iframe>
```

## 5.2 Retorno de control a la aplicación

El envío del formulario anterior mostrará entonces, dentro del contenedor, las páginas de Payment Gateway donde el usuario podrá cargar sus datos crediticios. Una vez cargados estos datos, y realizada la operación, en caso de aceptación o rechazo de la transacción, la aplicación redigirá a una página específica (llamada "TransactionTerminal"). Esta página se encuentra vacía y es solamente aplicable a la funcionalidad de retornar el control de la aplicación sin necesidad de utilizar un frame gráfico. El algoritmo es muy sencillo, si el terminal se carga, el contenedor desaparece o se redirecciona.

Esto se realiza gracias a dos elementos. El primero es una función de escucha que se ejecuta ante cada evento donde el iframe (adaptable fácilmente a InAppBrowser). Esta función se bindea en el contenedor:

```
<iframe name="my_frame" style="width:100%;height:90%" onLoad="checkForClose(this);"></iframe>
```

Y se define de la siguiente forma:

```
function checkForClose(pageURL) {  
  var urlFrame = pageURL.contentWindow.location;  
  if (urlFrame.href.indexOf('TransactionTerminal') > -1) {  
    window.location = "#/transaction/result";  
  }  
}
```

En verde, se encuentra resaltado el procedimiento a realizar cuando el sitio especificado es cargado (y por ende ha finalizado la transacción). Aquí se hace una redirección de manera local a un sitio específico de la aplicación cordova (el ejemplo se encuentra sobre la base de un template ionic de angular, pero ciertamente puede ser cualquier tipo de redirección base cordova).



En caso de no poder cerrarse definitivamente el frame, se sugiere que se especifique una URL a un archivo vacío y utilizar el nombre de este archivo para identificar el cierre (en vez de TransactionTerminal)

El servicio de Payment Gateway dispone de uno: [servicio/endpoints/autoclose.html](#)



## 5.3 Obtención de los datos de la transacción.

Luego de retomar el control de la aplicación, será necesario consultar lo sucedido en forma asíncrona. Para esto se puede utilizar una llamada común al servicio de información de transacciones, como figura en el siguiente ejemplo (de test). Los valores de las variables del ElectronicPaymentCode y de merchantId deben de ser las mismas que fueron enviadas para la transacción original por medio del requerimiento por POST.

***Si bien el ejemplo se encuentra en AngularJS-Ajax, se recomienda no utilizar este método directamente y utilizarlo a través de un servicio propio.***

```
var url = "https://test-pgmainservices.provincianet.com.ar/Transaction?electronicPaymentCode="
+ ElectronicPaymentCodeSent + "&merchantCode=" + merchantID;

var responseTransactionResult = $http.get(url);
```

El objeto resultante podrá ser tomado e interpretado para ser guardado y mostrado ante el cliente de la siguiente forma (parseándolo como un objeto JSON según punto 5.4.3):

```
responseTransactionResult .success(function(data, status, headers, config) {
    $scope.transaccionId = data.TransactionId;
    $scope.transaccionCode = data.ResponseCode;
    $scope.transaccionResult = data.ResponseMessage;
    $scope.transaccionReason = data.ResponseExtended;
    $scope.transaccionEPC = data.ElectronicPaymentCode;
});
```

## 6 Anulaciones

### 6.1 Disposición

Por razones de seguridad y de comercio verifique previamente con ProvinciaNET la disposición de este servicio.



## 7 Servicios Informativos

El servicio de Payment Gateway cuenta con varios servicios para consultar información asociada.

Los mismos son:

Servicio Informativo	Descripción
Servicio de Transacciones	Servicio de consulta sobre transacciones.
Servicio de Informes	Servicio de consulta sobre diversas configuraciones y estados.

### 7.1 Servicio de Transacciones

**El servicio de consulta de transacciones para configuraciones Mobile se consulta en forma diferente que el servicio para Web. Para transacciones a través de Web, deberá utilizarse el servicio previo.**

Para realizar la consulta se precisa el identificador de compra (EPC) y el identificador de comercio (Merchant ID):

```
Servicio/Transaction  
?  
electronicPaymentCode=valor1  
&  
merchantCode=valor2
```

*A partir de la versión Payment Gateway v3.2 este servicio informará cualquier tipo de transacción.*

### 7.2 Servicio de Informes

Servicio Informativo	Descripción
Information/creditCards	Devuelve todas las tarjetas disponibles por Payment Gateway
Information/creditCards/merchantId	Devuelve todas las tarjetas disponibles por Payment Gateway para el comercio especificado.

## 8 Ayuda para desarrolladores

Debido a que la tecnología REST puede ser nueva o desconocida para algunos desarrolladores, sugerimos la consulta del siguiente [link](#) donde podrán encontrar ejemplos para PHP, Node.js, Python, Objective-C, Rails, C# o .NET, Java y Android. Recomendamos también el uso del sitio [Stack Overflow](#).

## 9 Relativos o Dependencias