

2018

# *DOCUMENTACION SERVICIO DE TOKENIZACION*

## HISTORIAL DE VERSIONES

<b>Versión</b>	<b>Fecha</b>	<b>Autor</b>	<b>Comentario</b>	<b>Autorizado</b>
1.0	16/07/2018	José Javier A.	Creación de Documento	Marcelo Di Marco
1.1	07/09/2018	José Javier A.	Se agregan datos del ambiente de prueba	Marcelo DI Marco
1.2	27/09/2018	José Javier A.	Se agrega información sobre campos mandatorios	Marcelo DI Marco
1.3	14/12/2018	Jose Javier A.	Se agregan las url de producción	Marcelo Di Marco
1.4	11/2/2019	Jose Javier A	Se agrega detalle del servicio que permite actualizar y borrar medios de pago.	Marcelo Di Marco
1.5	03/10/2019	Jose Javier A	Se agregan ejemplos para ejecutar el formulario de captura.	Marcelo Di Marco

## Contenido

1. Preámbulo.....	4
2. Conceptos .....	4
2.1. Api.....	4
2.2. Url Base .....	4
2.3. Versiones De Api .....	4
2.4. Autenticación .....	5
2.4.1. PublicAccountKey .....	5
2.4.2. PrivateAccountKey .....	5
a. Tokenizacion.....	5
2.5. Formato de Respuesta.....	6
2.6. Identificador Único .....	6
3. Formulario de Checkout .....	6
3.1. Importación de Librería .....	7
3.2. Métodos .....	7
3.3. Proceso de Invocación Formulario Checkout .....	8
4. Flujos Transaccional .....	10
4.1.1. Proceso de Transacción para Usuarios Anónimos .....	10
4.1.2. Proceso de Transacción para Usuarios Registrados .....	12
4.1.2.1. Registro de Usuarios: .....	12
4.1.2.2. Registro de Medios de Pago para un Usuario Registrado .....	12
4.1.2.3. Proceso de Activación de Medio de Pago .....	14
5. Operaciones Permitidas.....	15
5.1. Directivas de Operaciones .....	15
6. Detalla de APIS y Objetos .....	16
6.1. Objeto Token.....	16
6.2. Objeto Purchase.....	17
6.2.1. Api para Crear una Nueva Compra .....	18
6.2.2. Obtener información de una compra .....	18
6.2.3. Anular una compra .....	19
6.2.4. Listar comprar.....	19
6.3. Objeto Customer.....	20
6.3.1. API para Crear un Nuevo Cliente .....	21

6.3.2. Actualizar datos del Cliente.....	21
6.3.3. Obtener datos de un cliente .....	21
6.3.4. Activar un perfil de pago .....	22
6.4. Objeto PaymentProfile .....	22
6.4.1. Actualizar PaymentProfile.....	22
6.4.2. Eliminar PaymentProfile .....	23
6.5. Objeto CustomerActivation .....	23
6.6. Objeto Transactions .....	23
6.7. Objeto Address.....	24
6.8. TransactinStep.....	24
6.9. CountryDataDo .....	25
6.10. Objeto Notification .....	25
7. Servicio WebHook del Comercio.....	25
7.1. Especificaciones del servicio WebHook .....	26
8. Códigos HTTP .....	27
8.1. Códigos de Error .....	27
8.2. Códigos de Respuesta de La transacción .....	28
9. Tipos de datos.....	30
9.1. Datos Básicos .....	30
9.2. String.....	30
9.3. Numeric .....	30
9.4. Amount .....	30
9.5. TimeStamp.....	31
9.6. Transaction Status.....	31
9.7. ActionType .....	31
9.8. Medios de Pago .....	32
10. Datos de Ambientes .....	32
10.1. Datos ambientes de pruebas.....	32
10.2. Datos ambiente de Produccion .....	32

## TOKENIZACION MANUAL TECNICO

### 1. Preámbulo

El siguiente documento describe los métodos y parámetros disponibles para la integración segura con la plataforma de "Card On File", utilizada para procesar transacciones de comercio electrónico brindando una experiencia de pago mejorada, permitiendo solicitar los datos de la tarjeta, como si estuviera en su carrito de compra o aplicación móvil.

Además, solicitar los datos de tarjeta de sus clientes por medio de un registro y guardar únicamente un "token" el cual estará asociado a la tarjeta ingresada de forma segura en el formulario de check-out alojado en nuestra plataforma.

### 2. Conceptos

#### 2.1. Api

La API está implementada a través de operaciones que son expuestas como servicios REST con intercambio de mensajes en formato JSON (*Content-Type: application/json*). Toda llamada a la API deberá ser realizada mediante HTTPS (TLS 1.2) y deberá controlar el [código HTTP](#) de respuesta de cada operación, que determinará el estado del procesamiento de la transacción enviada

#### 2.2. Url Base

El url base se refiere a la url donde se encuentra publicado el servicio, se menciona como ambiente a la ruta completa donde se referencia el url base y el api a utilizar, teniendo en cuenta que el url base puede cambiar de acuerdo al tipo de ambiente, producción o Certificación. Ejemplo: ambiente/v1/nombreservicios

#### 2.3. Versiones De Api

Las versiones del servicio serán identificadas en la url especificada de cada ambiente, para consumir el servicio con cada nueva version, se deberá referenciar la version de la siguiente manera.

Ejemplo: ambiente/v1 → Version 1.0

## 2.4. Autenticación

Todo comercio integrado con nuestra plataforma recibirá un par de claves por cada cuenta asociada. (*PrivateAccountKey* y *PublicAccountKey*).

Cada una de estas claves identifica al comercio en nuestro sistema en cada llamada que se hace a la API, de forma de poder establecer si está autorizado para operar y cuáles son las condiciones de dicha integración (ej. medios de pago habilitados).

Las claves serán utilizadas como forma de autenticación con la API mediante el método HTTP Basic Authentication. Es decir que cada llamada a cualquiera de las operaciones expuestas deberá ser realizada utilizando la API Key correspondiente como el dato de "username" de Basic Authentication (sin necesidad de informar "password").

### 2.4.1. PublicAccountKey

La clave pública de comercio se adjuntará en cada request público (realizado desde la interfaz web) para las operaciones de consulta o solicitud de captura de datos.

### 2.4.2. PrivateAccountKey

La clave privada de comercio se adjuntará en cada request privado (server to server) donde estarán disponibles las operaciones críticas, como iniciar o confirmar una transacción, acceder a información de transacciones del comercio, crear o eliminar planes de suscripción, etc.

**IMPORTANTE:** Esta clave privada no debe ser compartida ni expuesta públicamente en ningún momento para no comprometer la seguridad de la integración. Es responsabilidad del comercio mantenerla segura.

## a. Tokenizacion

Se refiere al proceso de sustitución de un elemento de datos sensibles con un equivalente no sensible. Este elemento no sensible (el "token") generado a través de este proceso carece de significado o valor, excepto para la entidad a quien fue entregado. De esta forma, en caso de un compromiso de seguridad que permita robar estos números, no tienen valor alguno para quien los roba ya que no podrá usarlos para su beneficio propio.

## 2.5. Formato de Respuesta

Cualquier llamada a una operación de la API, va a retornar:

- Un código HTTP que define el estado de la respuesta.
- El objeto de respuesta al request.
- Una lista de errores que contiene todos los errores que pueden haber ocurrido. En caso de no haber errores la lista está vacía.
- El código de respuesta del banco emisor.

## 2.6. Identificador Único

Es importante que, en algunos de los servicios implementados en la API, se evite la duplicación de transacciones. Por ejemplo, cuando se hacen las operaciones de compra.

Si ocurre una falla de conexión de red en el medio de una transacción de compra, la misma pudo haber sido aprobada pero el comercio no obtuvo la respuesta final.

Con el identificador único, se puede reintentar la operación con la tranquilidad de que en caso de que la plataforma ya la haya procesado, va a contestar el resultado que ya fue obtenido del medio de pago, sin duplicar la transacción.

## 3. Formulario de Checkout

El formulario de Checkout es un formulario de pago embebido en su propia página, que simplifica y asegura la captura de datos sensibles para el procesamiento de pagos en línea. La integración de este formulario en su sitio o "app", proporcionará a sus usuarios una experiencia de pago simplificada y responsiva apta para web y aplicaciones móviles.



The image shows a mobile-optimized checkout form for CARDNET. At the top is the CARDNET logo and the text "Comercio Test.". Below this are several input fields: "Titular:" with a person icon and the placeholder "titular"; "Email" with an envelope icon and the placeholder "gpigni@pagosweb.com.uy"; "Número de Tarjeta:" with a card icon and a masked placeholder ".... .... ."; "Expiración:" with a calendar icon and the placeholder "MM/YY"; and "CVV:" with a lock icon and the placeholder "XXX". At the bottom is a blue button labeled "Pagar" with a checkmark icon.

### 3.1. Importación de Librería

Las funcionalidades del formulario de Checkout se encuentran en una librería JavaScript, que debe ser importada en la página web del cliente directamente desde una URL pública de la plataforma. En la llamada a dicha librería se deberá incluir (como parámetro) la clave pública de la cuenta de comercio (PublicAccountKey) la cual será utilizada para las llamadas hacia la API REST desde esta librería.

#### Ejemplo

```
<scriptsrc="{ambiente_api}/v1/Scripts/Capture.js?key={PublicAccountKey}"
type="text/javascript"></script>
```

**NOTA:** La librería debe ser importada por el comercio a través de la URL pública alojada en CARDNET. No deberá ser descargada y usada localmente desde un servidor propio del comercio o desde una URL de un tercero no autorizado por CARDNET. Esto es importante por motivos de seguridad y para mantenerse siempre actualizados con las últimas modificaciones y correcciones realizadas sobre la misma.

### 3.2. Métodos

#### SetProperties

Este método establece las propiedades visuales (textos e imágenes) a utilizarse en la ventana de Checkout. Todas las propiedades son opcionales, excepto la propiedad "form\_id". Por lo que la llamada al método SetProperties es obligatoria por lo menos para establecer esta propiedad.

Propiedad	Descripción	Presencia
name	Nombre que aparece como "Título" del Checkout.	Opcional
email	Email del cliente que se puede precargar.	Opcional
image	URL absoluta de la imagen a utilizarse dentro del Checkout (formato fijo). Imágenes aceptadas: jpg, jpeg, png	Opcional
button_lab	Texto a mostrarse en el botón de pago (el keyword #monto# es sustituido por el monto si este es informado).	Opcional
description	Descripción del pago a realizarse.	Opcional
currency	Moneda del pago (ver tabla de monedas).	Opcional
amount	Monto del pago (solo para informar al usuario, el monto real se informa server to server en la operación Purchase).	Opcional
lang	Lenguaje de la interfaz. Si no se establece se obtiene de la configuración del browser.	Opcional
form_id	Identificador del formulario web donde se manejan los datos de la compra actual. Este dato es requerido ya que se utilizará para informar el token generado por la aplicación a partir de la tarjeta o medio de pago ingresado por el usuario.	Mandatorio
checkout_card	Si se establece, el checkout pasa directamente a la captura de tarjeta.	Opcional
empty	Si esta propiedad se establece en "true" la imagen y el título en la ventana del formulario se ocultarán.	Opcional



autoSubmit	Propiedad que permite establecer el comportamiento de la librería PWCheckout una vez que se recibe el Token. Si esta propiedad es establecida en "false" el formulario no será enviado automáticamente. El valor por defecto es "true", por lo cual si esta propiedad no es establecida, el formulario será enviado automáticamente al recibir el Token del medio de pago.	Opcional
------------	--	----------

```

<script type="text/javascript">

    PWCheckout.SetProperties(
    {
        "name": "Mi tienda",
        "email": "cliente@gmail.com",
        "image": "http://mitienda.com/images/logocheckout.png",
        "button_label": "Pagar #monto#",
        "description": "Checkout de Mi tienda",
        "currency": "DOP",
        "amount": "1.00",
        "lang": "ESP",
        "form_id": "commerce_form",
        "checkout_card": "1",
        "autoSubmit": "true"
    }
    );
</script>

```

### 3.3. Proceso de Invocación Formulario Checkout

1. La página web del comercio desde la que se realizará la invocación al Formulario de Checkout deberá contener un campo de tipo oculto (input "hidden") con identificador "PWToken", cuyo valor será seteado desde la librería JavaScript luego de obtener el valor del token.

```
<input type="hidden" name="PWToken" id="PWToken" />
```

2. Se debe asociar cual es el elemento de la página que realizará la llamada al formulario de captura de la tarjeta de pago. Este elemento puede ser un botón, imagen o cualquier elemento activo que se desee asociar. Para realizar la asociación, se deberá llamar al método AddActionButton de la librería **PWCheckout** de la siguiente manera:

```

<script type="text/javascript">

    PWCheckout.AddActionButton("buttonId1", "buttonId2", "buttonId3");

</script>

```

Donde el parámetro "buttonId" se refiere al identificador del elemento elegido para desencadenar el proceso de pago. Además, se pueden asociar múltiples elementos en la misma llamada, identificando cada uno mediante su ID separados por coma. Al presionar el elemento asociado, se invocará el proceso de pago.

3. Luego la librería JavaScript se encargará de administrar el proceso de pago, mostrando al usuario el formulario donde se le solicitarán los datos de la tarjeta para realizar el pago. Una vez ingresados los datos, dicha información será tokenizada y obtenida por la librería PWCheckout. El valor del token generado, será insertado automáticamente en el campo oculto "PWToken", y finalmente la librería JavaScript PWCheckout realizará el "submit" del formulario de pago.
4. El comercio deberá procesar los datos del formulario de pago, incluyendo el valor que recibirá en el campo "PWToken" y enviará dicha información tal como se describe en la sección **Purchase** de la API.

Ejemplo completo:

```
<head>

<script type="text/javascript"
src="https://lab.cardnet.com.do/servicios/tokens/v1/Scripts/PWCheckout.js?key=mfH9CqiAFjFQh_gQR_1TQG_I56ONV
7HQ"></script>

</head>
<form id="shoppingcart_form">
<p>
    <button id="btnCheckout">ABRIR IFRAME DE CAPTURA</button>
</p>
<p>
<span class="itemName">OTT TOKEN: <input type="text" id="PWTokenAux" name="PWTokenAux" /></span>
</p>
</form>

<script>
    PWCheckout.Bind("tokenCreated", OnTokenReceived);
    PWCheckout.SetProperties({
        "name": "Demo Test.",
        "email": "gpigni@pagosweb.com.uy",
        // "image": "http://pepito.com/images/logocheckout.png",
        "button-label": "Pagar #monto#",
        "description": "Checkout Creditel Test.",
        "currency": "$",
        "amount": "100",
        "lang": "ESP",
        "form_id": "shoppingcart_form",
        "checkout_card": 1,
        "autoSubmit": "false",
        "empty": "true"
    });
    function OnTokenReceived(token) {alert(token.TokenId);
        document.getElementById("PWTokenAux").value = token.TokenId;}
        document.getElementById("btnCheckout").addEventListener("click", function(event){
            event.preventDefault(); myFunction();
        });
    function myFunction () {
```

```
//estos datos se obtienen haciendo un get del Customer , debe tener en cuenta que esto puede cambiar de acuerdo al ambiente.  
var captureUrl = "https://lab.cardnet.com.do/servicios/tokens/v1/Capture/";  
var customerUniqueld = "XXXXXXXXXXXXXXXXXXXX";  
  
PWCheckout.OpenIframeCustom(captureUrl+ "?key=" + "mfH9CqiAfjFQh_gQR_1TQG_I56ONV7HQ" + "&session_id=" +  
customerUniqueld, customerUniqueld);  
    }  
  
</script>
```

## 4. Flujos Transaccional

A continuación, se describen los flujos para autorizar una compra a través de la plataforma.

Existen 2 flujos posibles de compras:

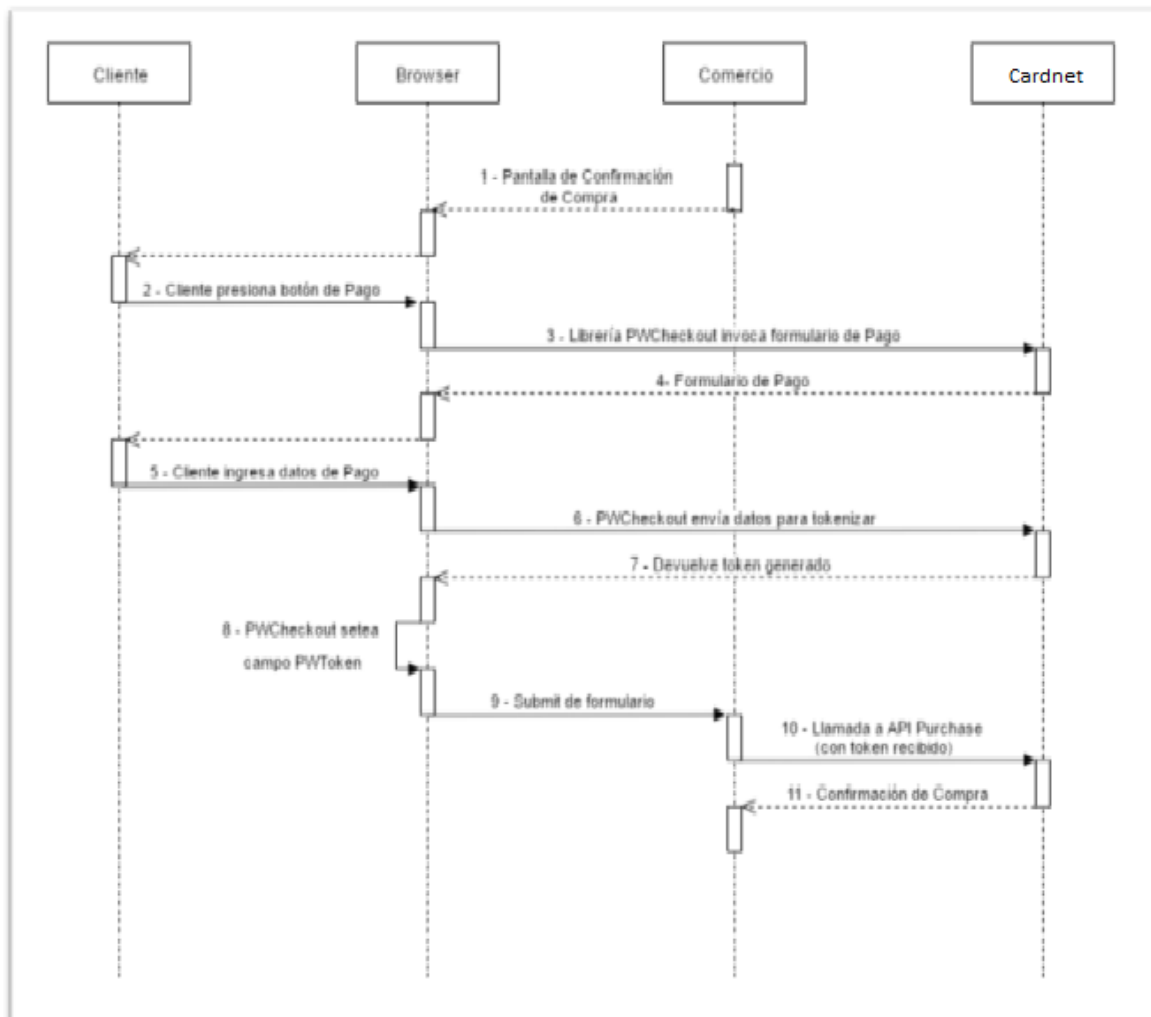
- **Usuarios anónimos**, que no están registrados en el sitio y hacen una compra por única vez. En este caso se debe pedir siempre los datos de tarjetas para poder realizar la transacción.
- **Usuarios registrados**, para clientes que sí tienen un registro dentro del sitio web, se pueden identificar y por lo tanto se pueden asociar los datos de tarjeta que se ingresan una vez para luego realizar otras compras sin necesidad de ingresar nuevamente los datos.

### 4.1.1. Proceso de Transacción para Usuarios Anónimos

1. Desde el carrito de compra del comercio invocar el "Formulario de Checkout" a través de la librería JavaScript de la plataforma de CARDNET. Este formulario se muestra dentro de un iFrame de su propia página y solicita los datos de tarjeta.
2. Una vez que se invoca el formulario de checkout, y el cliente llena los datos, el comercio podrá obtener el "token" asociado a la tarjeta del cliente. Este token es un "One Time Token" y tiene validez por una única vez y está vigente durante 10 minutos
3. El token obtenido debe ser enviado, desde el navegador o la app móvil, al servidor de la aplicación para que éste haga la transacción de compra. Desde el servidor se llama por HTTP POST a: **{ambiente api}/v1/api/purchase** adjuntando el objeto *Purchase* que contendrá el token y otros datos de la transacción.

```
{  
  'TrxToken': 'OT_01_kYv0qTHckRiZ4wjCz5NguZRuwFLSIrQc4jiYpVJ8SzQ_',  
  'Order': '17030613595101621fb',  
  'Amount': 123456,  
  'Currency': 'DOP',  
  'Capture': true  
}
```

## Flujo De Transacción Con Usuarios Anónimos



## 4.1.2. Proceso de Transacción para Usuarios Registrados

### 4.1.2.1. Registro de Usuarios:

1. Para poder operar con usuarios registrados, se debe primero registrar dicho usuario en el Sistema de Pagos, para ello se deberá llamar a la interfaz de Customers informando los datos del usuario que se desea registrar. Ver Referencia [Customers](#)
2. Como resultado del registro de dicho usuario, se recibirá como respuesta un objeto Customer, el cual se deberá procesar y almacenar como mínimo el CustomerId, con el cual se podrán realizar otras operaciones sobre dicho Customer registrado.

### 4.1.2.2. Registro de Medios de Pago para un Usuario Registrado

1. (El usuario (debidamente autenticado en la web del comercio) solicita el registro de un medio de pago nuevo en su cuenta dentro del comercio.
2. El Browser navega hacia la página de registro de medios de pago del comercio.
3. El comercio, debe acceder a la cuenta del usuario realizando la llamada (GET) al servicio de Customer, informando el CustomerId que debió previamente haber almacenado (tal cual lo indica el flujo de Registro de Usuarios).
4. El objeto Customer devuelto, contendrá la información conocida del usuario, incluyendo los Medios de Pago que pueden haber sido registrados previamente. Además, este objeto Customer informa un "CaptureURL" y un "UniqueID" que debe ser utilizado luego para mostrarle al usuario la interfaz de captura de datos de la tarjeta de pago.
5. El comercio, deberá presentar al usuario la pantalla de registro de Medios de Pago, donde por ejemplo podrá disponer un botón de tipo "Agregar nuevo Medio de Pago" el cual iniciará el proceso de registro.
6. El cliente presiona este botón (o el método elegido por el comercio) para iniciar el método de captura de la tarjeta de pago.
7. La interfaz del sitio del comercio deberá abrir y mostrar al usuario la página de captura de tarjetas. Para esto puede por ejemplo utilizar el método OpenIframeCustom de la librería javascript PWCheckout, a la cual le deberá informar 2 parámetros:
  - **URL:** donde deberá concatenar el dato CaptureURL obtenido en el paso 4, con la clave pública del comercio y el UniqueID (también obtenido en el paso 4). El formato de el parámetro URL es el siguiente:

```
<script type="text/javascript">  
  {CaptureURL}?key={publicKey}&session_id={UniqueID}  
</script>
```

- **UniqueID:** como segundo parámetro de la función, se deberá informar nuevamente el UniqueID obtenido en el paso 4.

8. El cliente en este paso deberá interactuar con la interfaz de captura de tarjeta, ingresando los datos de la misma.
9. El formulario de captura enviará la información ingresada por el usuario directamente a los sistemas de tokenización del Sistema de Pagos, donde se almacenarán dichos datos y se generará un Token para representar dicha tarjeta.
10. Una vez que el proceso de capturar la tarjeta del usuario haya finalizado, la librería PWCheckout recibirá una notificación, la cual puede ser manejada por la web del comercio, suscribiéndose al evento "tokenCreated", el cual se desencadenará en el momento de que el token esté listo.
11. Cuando el comercio recibe la notificación de que el Medio de Pago ha sido correctamente registrado, deberá volver a llamar (GET) al servicio de Customer de la API para obtener una versión actualizada de la información del dicho usuario.
12. Como respuesta, el objeto Customer devuelto tendrá todos los medios de pago del usuario, incluyendo el nuevo recién registrado. Con esta información el comercio deberá actualizar el perfil del usuario en su sistema, almacenando el **Token** registrado y los datos asociados al mismo.

Ej. **PaymentProfileId, Brand, Expiration, Last4, etc.**).

Además, deberá verificar que el objeto PaymentProfile (dentro del Customer recibido) podrá estar marcado como deshabilitado (Enabled=false) por lo que para que dicho perfil (Token) pueda ser utilizado, deberá ser activado.

A continuación, se muestra un ejemplo de respuesta que incluye un PaymentProfile

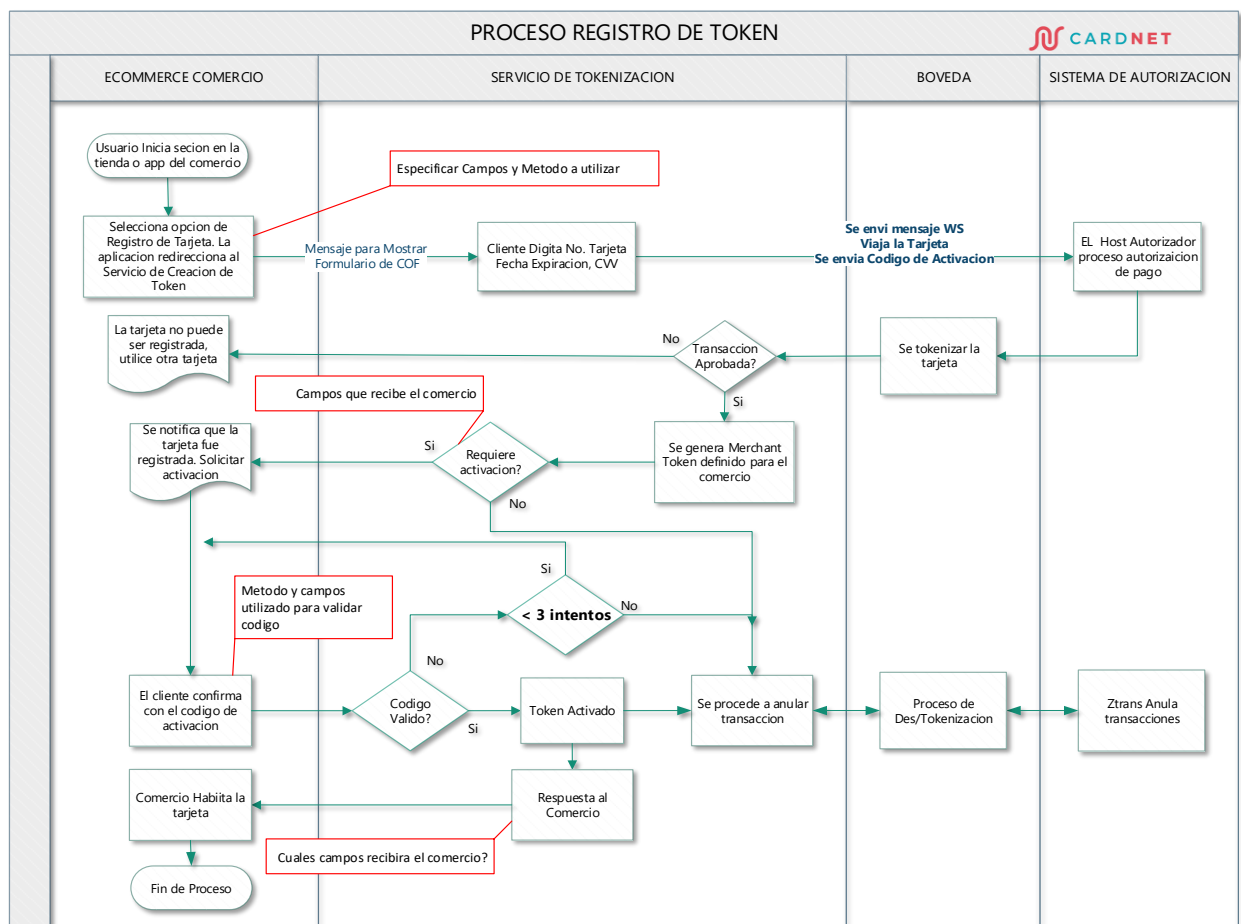
```
{
  "Response": {
    "CustomerId": 50329,
    "Created": "2018-01-24T12:41:35.503",
    "CommerceCustomerId": "test@email.com",
    "Owner": "Commerce",
    "Email": "test@email.com",
    "Enabled": true,
    "ShippingAddress": null,
    "BillingAddress": null,
    "Plans": null,
    "AdditionalData": null,
    "PaymentProfiles": [
      {
        "PaymentProfileId": 50330,
        "PaymentMediaId": 2,
        "Brand": "MasterCard",
        "IssuerBank": null,
        "Type": "CreditCard",
        "Token": "CT__3RIuqDhQmyeg7RUCF_xdtZelY5Ua2zXniwnDMVFoTWs_",
        "Expiration": "202211",
        "Last4": "0001",
        "Enabled": false
      }
    ],
    "CaptureURL": "http://api.localsiemprepago.com/Capture",
    "UniqueID": "UI_4e9ab4f2-d007-4663-815a-6a78962238e3",
    "URL": "http://api.localsiemprepago.com/api/Customer/50329",
    "FirstName": "Test",
    "LastName": "Account",
    "DocNumber": "12345678",
    "DocumentTypeId": 2,
    "PhoneNumber": null
  },
  "Errors": []
}
```

**NOTA:** La información de un **"Token"** no debe ser expuesta públicamente ya que representa la tarjeta capturada y puede ser utilizada múltiples veces para realizar transacciones.

### 4.1.2.3. Proceso de Activación de Medio de Pago

1. El comercio deberá mostrarle al usuario los distintos Medios de Pago registrados por él, indicando en los casos que corresponda que dicho medio de pago está deshabilitado y que se necesita "Activar".
2. El usuario selecciona la opción de activar este Medio de Pago.
3. El comercio deberá presentarle al usuario un campo donde el mismo pueda ingresar el código de activación que oportunamente recibió (el método de entrega de dicho código queda fuera de este alcance).
4. Una vez que el comercio obtuvo el código ingresado por el usuario, se deberá realizar una llamada a la operación "Activate" de la API de Customer del Sistema de Pago, informando el CustomerId del usuario registrado, y enviando la información del Token que se desea activar junto con el código de activación que fue obtenido.
5. Como respuesta a este proceso de activación, la API retornará nuevamente el objeto Customer completo, con los datos actualizados, incluyendo el perfil de pago habilitado en caso de que el proceso haya sido exitoso. Tener en cuenta que en caso de errores, estos se informarán como parte de la respuesta del servicio

### Flujo de Clientes Registrados



## 5. Operaciones Permitidas

La plataforma de Card On File permitirá a los comercios realizar diferentes tipos de operaciones que pueden ser realizadas durante todo el proceso de compra.

Objeto	Operaciones	Tipo Transacción	Método
<b>Customer</b>	Crear Customer	Registro de Cliente	POST
	Consultar		GET
	Actualizar	Activar Perfil de Pago	POST
<b>Purchase</b>	Crear	Realizar compra	POST
	Consultar	Consulta de compra	GET
	Lista	Listar Compras	GET
	Actualizar	Anular compra	POST
<b>Customer Activation</b>	Actualizar	Activar Medio de Pago	POST
<b>PaymentProfile</b>	Actualizar	Permite activar o desactivar un medio de pago	POST
	Remover	Permite borrar medio de pago	POST

### 5.1. Directivas de Operaciones

Cada objeto tiene operaciones que permiten crear, consultar o actualizar.

#### Creación de Objetos.

La creación de objetos se hace a través de un método POST a la URL de formato:

[{ambiente\\_api}/v1/api/{object-name}](#)

#### Actualización de Objetos

Cada objeto puede tener distintas operaciones de actualización. La llamada básica se hace a través de un método POST a la URL de formato:

[{ambiente\\_api}/v1/api/{object-name}/{object-id}/{operation}](#)

#### Consulta de Objetos

Hay 2 tipos de consultas sobre los objetos:

- La consulta que devuelve un objeto a través de su ID. Se hace a través de un método GET a la URL de formato:

[{ambiente\\_api}/v1/api/{object-name}/{object-id}](#)

- La consulta que devuelve una lista de objetos de acuerdo a diversos parámetros. Se hace a través de un método GET a la URL de formato:

[{ambiente\\_api}/v1/api/{object-name}?{query-parameters}](#)



## Nomenclatura de Objetos

Todos los objetos están descritos en tablas con el siguiente formato:

<b>Campo</b>	Es el nombre del campo en el objeto JSON.
<b>Tipo</b>	Es el tipo de datos del campo (ver en Tipos de datos la descripción de cada uno).
<b>Descripción</b>	Descripción del campo indicado, con comentarios de ayuda
<b>Presencia</b>	Define si el campo es mandatorio, opcional o solo lectura
<b>Default</b>	Para los campos opcionales, se indica el valor que se toma por default

## 6. Detalla de APIS y Objetos

### 6.1. Objeto Token

El objeto **Token** es utilizado para informar el token generado junto con otra información relativa al medio de pago utilizado por el cliente.

Campo	Descripción	Tipo	Presencia Default
<b>TokenId</b>	Es el token generado.	string	Solo Lectura
<b>Created</b>	Fecha y hora de creación del token	TimeStamp	Solo Lectura
<b>Type</b>	Tipo de token generado, valores posibles: "OneTime" "Commerce"	string	Solo Lectura
<b>Brand</b>	Marca de la tarjeta o medio de pago utilizado	string	Solo Lectura
<b>IssuerBank</b>	Banco emisor de la tarjeta (en caso que pueda ser determinado).	string	Solo Lectura
<b>Owner</b>	Nombre del tarjetahabiente	string	Solo Lectura
<b>Last4</b>	Últimos cuatro dígitos de la tarjeta.	Numeric[4]	Solo Lectura
<b>CardExpMonth</b>	Mes de expiración de la tarjeta	Numeric[2]	Solo Lectura
<b>CardExpYear</b>	Año de expiración de la tarjeta	Numeric[2]	Solo Lectura
<b>URL</b>	URL donde se puede acceder a la información del Token (ej. {ambiente_api}/v1/api/token/{token-id}).	string	Solo Lectura

## 6.2. Objeto Purchase

El objeto Purchase es utilizada para la acción de autorización, anulación o consulta de compras.

Campo	Descripción	Tipo	Presencia Default
<b>PurchaseId</b>	Identificador de la compra.	Numeric	Solo Lectura
<b>Created</b>	Fecha y hora del momento de la creación de la compra. Este campo está presente en la respuesta a consultas. No se incluye o valida en la creación o actualizaciones del objeto.	TimeStamp	Solo Lectura
<b>TrxToken</b>	Token que identifica la tarjeta del cliente	Token	Mandatorio
<b>Order</b>	Número de orden, generado por el comercio	string	Mandatorio
<b>Transaction</b>	Contiene la información que resulta de la transacción realizada contra el medio de pago (por ej: código de respuesta, número de autorización). Ver objeto Transaction.	Transaction	Solo Lectura
<b>Capture</b>	Establece si la compra se debe realizar en uno o dos pasos. Es de tipo booleano cuyo valor por defecto es "true". Si es false, solo se procesa la autorización y la compra queda pre-autorizada a la espera de la confirmación final a través de las llamadas commit (confirmar) y rollback (anular). Si es true, la transacción queda autorizada y capturada (confirmada).	Boolean	Opcional TRUE. Nota: Por el momento no está disponible.
<b>Amount</b>	Monto total de la compra. El valor debe ser mayor a cero.	Amount	Mandatorio
<b>Currency</b>	Moneda de la compra, de acuerdo a ISO-4217 (códigos alfanuméricos).	String[3]	Mandatorio
<b>Tip</b>	Propina	Amount	Opcional
<b>Installments</b>	Cantidad de cuotas de la compra.	Numeric[3]	Opcional 1
<b>Description]</b>	Descripción opcional de la compra	String[50]	Opcional
<b>Customer</b>	Información del cliente que realiza el pago. Algunos medios de pago pueden requerir información adicional del cliente para poder tramitar la autorización.	Customer	Opcional
<b>RefundList</b>	Lista de devoluciones realizadas a la compra.	List<RefundData>	Solo Lectura
<b>PlanID</b>	Reservado		
<b>UniqueID</b>	Identificador único de la compra. Este valor opcional permite identificar una compra única y evitar la duplicación de transacciones en caso de errores de comunicación (ver más en Conceptos / Identificador único).	String[50]	Opcional
<b>AdditionalData</b>	Información adicional que el comercio puede agregar a la transacción (Ej.: lista de datos de tipo "Clave:Valor").Dicha información será devuelta al procesar la compra y cada vez que dicha compra sea consultada.	String[300]	Opcional

<b>CustomerUserAgent</b>	User Agent del cliente que utiliza el servicio, en la Web debería ser el UserAgent reportado por el navegador, en el caso de móviles información acerca del dispositivo, S.O. utilizado, nombre de la App.	String	Opcional
<b>CustomerIP</b>	IP del cliente que utiliza el servicio.	String	Opcional
<b>DataDo</b>	Datos específicos para la Rep. Dominicana. Ver la definición del objeto CountryDataDo. Detalle del objeto en la sección <a href="#">6.9 CountryDataDo</a> .	CountryDataDo	Mandatorio
<b>CommerceAction</b>	Utilizado para indicar al comercio una acción que deba ser realizada por él o por el customer para completar el proceso de la compra actual. Si la transacción se devolvió en estado Pending, se deberá revisar este objeto para determinar las acciones siguientes.	CommerceAction	Solo Lectura
<b>URL</b>	URL donde se puede acceder a la información de la Compra. Ej: {ambiente_api}/v1/api/purchase/{purchase-id}.	String	Solo Lectura

### 6.2.1. Api para Crear una Nueva Compra

<b>URL:</b>	{ambiente_api}/v1/api/purchase	
<b>Tipo API:</b>	Pública	
<b>Method:</b>	POST	
<b>Autenticación:</b>	PrivateAccountKey	
<b>Parámetros:</b>	Purchase Purchase	Objeto que contiene la información de la compra a realizarse.
<b>Respuesta:</b>	HTTP Code	Código HTTP.
	Response Purchase	Objeto que contiene la información actualizada de la compra realizada. Se le asigna el PurchaseID que será utilizado como identificador de la compra a partir de este momento.
	Errors List<Error>	Lista de errores que se pueden haber generado.

### 6.2.2. Obtener información de una compra

<b>URL:</b>	{ambiente_api}/v1/api/purchase/{purchase-id}	
<b>Tipo API:</b>	Pública	
<b>Método:</b>	GET	
<b>Autenticación:</b>	PrivateAccountKey	
<b>Parámetros</b>	PurchaseID Numeric	Identificador de la compra
<b>Respuesta:</b>	HTTP Code	Código HTTP.
	Response Purchase	Objeto que contiene la información de la compra solicitada.
	Errors List<Error>	Lista de errores que se pueden haber generado.

### 6.2.3. Anular una compra

<b>URL:</b>	{ambiente_api}/v1/api/purchase/{purchase-id}/refund	
<b>Tipo API:</b>	Pública	
<b>Método:</b>	POST	
<b>Autenticación:</b>	PrivateAccountKey	
<b>Parámetros</b>	PurchaseID Numeric	Identificador de la compra
<b>Respuesta:</b>	HTTP Code	Código HTTP.
	Response Purchase	Objeto que contiene la información de la compra actualizada.
	Errors List<Error>	Lista de errores que se pueden haber generado.

### 6.2.4. Listar comprar

Este método permite obtener una lista de compras de acuerdo a los filtros que se pueden establecer en los parámetros

<b>URL:</b>	{ambiente_api}/v1/api/purchase			
<b>Tipo API:</b>	Pública			
<b>Método:</b>	GET			
<b>Autenticación:</b>	PrivateAccountKey			
<b>Parámetros</b>	<b>Nombre</b>	<b>Descripción</b>	<b>Tipo</b>	<b>Valor por def.</b>
	CustomerID	Identificador del cliente que realizó la compra.	Numeric	Ninguno
	From	Fecha inicial del filtro, Formato: yyyyMMdd	Date	La fecha del día
	To	Fecha final del filtro, Formato: yyyyMMdd	Date	La fecha del día
	PaymentMediaId	Identificador del Medio de Pago con el que se realizaron las compras.	Numeric	Ninguno
	Authorized	Si el valor es true devuelve solo las compras que hayan finalizado exitosamente	Boolean	false
	OrderNumber	Número de orden del comercio.	String	Ninguno
<b>Respuesta:</b>	HTTP Code	Código HTTP.		
	Response	Lista de Compras que coinciden con los filtros de búsqueda.	List<Purchase>	
	Errors	Lista de errores que se pueden haber generado.	List<Error>	

### 6.3. Objeto Customer

El objeto Customer es utilizada para la acción de creación y mantenimiento de los clientes.

Campo	Descripción	Tipo	Presencia Default
<b>CustomerId</b>	Identificador del cliente.	Numeric	Solo Lectura
<b>CommerceCustomerId</b>	Identificador del cliente en el comercio. Este valor es generado y utilizado internamente por el comercio para identificar al cliente dentro de la plataforma.	String	Opcional
<b>Created</b>	Fecha y hora del momento de la creación del cliente. Este campo está presente en la respuesta a consultas. No se incluye o valida en la creación o actualizaciones del objeto.	TimeStamp	Solo Lectura
<b>Owner</b>	Determina si el usuario fue registrado por el comercio, a través de Cardnet o anónimo. Valores posibles: "Our", "Commerce", "Anonymous". Este campo está presente en la respuesta a consultas. No se incluye o valida en la creación o actualizaciones del objeto.	String	Solo Lectura
<b>FirstName</b>	Nombre del cliente	String	Opcional
<b>LastName</b>	Apellido del cliente	String	Opcional
<b>Email</b>	Email del cliente	String	Mandatorio
<b>PhoneNumber</b>	Teléfono de contacto del cliente.	String	Opcional
<b>Enabled</b>	Indica si el usuario está habilitado o no a operar.	Boolean	Opcional true
<b>ShippingAddress</b>	Información de dirección de entrega (si está disponible).	Address	Opcional
<b>BillingAddress</b>	Información de dirección de facturación (si está disponible y el usuario habilitó que se compartiera esa información).	Address	Opcional
<b>Plans</b>	Reservado		
<b>AdditionalData</b>	Lista de datos de tipo "Clave:Valor" para almacenar información extra.		Opcional
<b>PaymentProfiles</b>	Lista de objetos PaymentProfile con información de los medios de pago registrados por el Customer.	List<PaymentProfile>	Solo Lectura
<b>CaptureURL</b>	URL de captura de datos de tarjeta (es la URL que se debe abrir en un iframe para iniciar el proceso de captura de datos sensibles). Solo es válida para Customers de tipo "Commerce".	String	Opcional
<b>UniqueID</b>	Identificador único temporal utilizado para registrar medios de pago externos. Cada vez que se solicita la información del Customer se genera un identificador nuevo.		Solo Lectura
<b>URL</b>	URL donde se puede acceder a la información del Cliente (ej. /v1/customer/{customer-id}).	String	Solo Lectura
<b>DocumentTypeId</b>	Tipo de documento del cliente	Numeric	Opcional
<b>DocNumber</b>	Documento del cliente	String	Opcional

### 6.3.1. API para Crear un Nuevo Cliente

<b>URL:</b>	{ambiente_api}/v1/api/customer		
<b>Tipo API:</b>	Pública		
<b>Method:</b>	POST		
<b>Autenticación:</b>	PrivateAccountKey		
<b>Parámetros:</b>	<b>Campo</b>	<b>Tipo</b>	<b>Descripción</b>
	Customer	Customer	Objeto que contiene la información del cliente
<b>Respuesta:</b>	HTTP Code		Código HTTP.
	Response	Customer	Objeto que contiene la información del Cliente actualizada. Se asigna el valor de CustomerID.
	Errors	List<Error>	Lista de errores que se pueden haber generado.

### 6.3.2. Actualizar datos del Cliente

<b>URL:</b>	{ambiente_api}/v1/api/customer/{customer-id}/update		
<b>Tipo API:</b>	Pública		
<b>Method:</b>	POST		
<b>Autenticación:</b>	PrivateAccountKey		
<b>Parámetros:</b>	<b>Campo</b>	<b>Tipo</b>	<b>Descripción</b>
	CustomerID	Numeric	Identificador del Cliente a modificar.
<b>Respuesta:</b>	Customer	Customer	Objeto que contiene la información del cliente
	HTTP Code		Código HTTP.
	Response	Customer	Objeto que contiene la información actualizada del Cliente.
	Errors	List<Error>	Lista de errores que se pueden haber generado.

### 6.3.3. Obtener datos de un cliente

<b>URL:</b>	{ambiente_api}/v1/api/customer/{customer-id}		
<b>Tipo API:</b>	Pública		
<b>Method:</b>	GET		
<b>Autenticación:</b>	PrivateAccountKey		
<b>Parámetros:</b>	<b>Campo</b>	<b>Tipo</b>	<b>Descripción</b>
	CustomerID	Numeric	Identificador del Cliente a modificar.
<b>Respuesta:</b>	HTTP Code		Código HTTP.
	Response	Customer	Objeto que contiene la información del Cliente.
	Errors	List<Error>	Lista de errores que se pueden haber generado.

### 6.3.4. Activar un perfil de pago

<b>URL:</b>	{ambiente_api}/v1/api/customer/{customer-id}/activate		
<b>Tipo API:</b>	Pública		
<b>Method:</b>	POST		
<b>Autenticación:</b>	PrivateAccountKey		
<b>Parámetros:</b>	<b>Campo</b>	<b>Tipo</b>	<b>Descripción</b>
	Activation	CustomerActivation	Objeto que contiene la información relativa al perfil que se desea activar y el código de activación relacionado.
<b>Respuesta:</b>	HTTP Code		Código HTTP.
	Response	Customer	Objeto que contiene la información actualizada del Cliente.
	Errors	List<Error>	Lista de errores que se pueden haber generado.

## 6.4. Objeto PaymentProfile

Este objeto es obtenido luego de ingresar un medio de pago, en este se informan los datos de un medio de pago registrado por el cliente al que está asociado, o sea el número de la tarjeta. La información ser obtenida a través de la consulta del cliente.

Campo	Descripción	Tipo	Presencia Default
<b>PaymentProfileId</b>	Identificador del perfil de pago registrado para el Customer.	Numeric	Solo Lectura
<b>PaymentMediaId</b>	Identificador del medio de pago asociado al perfil.	Numeric	Solo Lectura
<b>Brand</b>	Nombre asociado a la marca de la tarjeta de pago, por ejemplo "VISA".	String	Solo Lectura
<b>IssuerBank</b>	Identificador del Banco Emisor. (en caso que pueda ser determinado)	String	Solo Lectura
<b>Type</b>	Tipo de medio de pago, por ejemplo "CreditCard"	String	Solo Lectura
<b>Token</b>	Dato que representa al medio de pago registrado sin exponer los datos sensibles del mismo. Este dato será utilizado para realizar transacciones de pago mediante el medio de pago registrado.	String	Solo Lectura
<b>Expiration</b>	Fecha de expiración del medio de pago (si corresponde). Formato: yyyyMM	String	Solo Lectura
<b>Last4</b>	Últimos 4 dígitos de la tarjeta de pago (si corresponde).	Numeric	Solo Lectura
<b>Enabled</b>	Determina si el perfil se encuentra habilitado para realizar pagos.	Boolean	Solo Lectura

### 6.4.1. Actualizar PaymentProfile

<b>URL:</b>	{ambiente_api}/v1/api/Customer/{customer-id}/PaymentProfileUpdate		
<b>Tipo API:</b>	Pública		
<b>Method:</b>	POST		
<b>Autenticación:</b>	PrivateAccountKey		
<b>Parámetros:</b>	<b>Campo</b>	<b>Tipo</b>	<b>Descripción</b>
	PaymentProfileID	Numeric	Identificador del medio de pago a modificar
	Expiration	String	Fecha de expiración de la tarjeta a modificar
	Enable	Boolean	Valor para actualizar estado del medio de pago, donde true es activo y false es inactivo.
<b>Respuesta:</b>	HTTP Code		Código HTTP.
	Response	Customer	Objeto que contiene la información del Cliente.
	Errors	List<Error>	Lista de errores que se pueden haber generado.

### 6.4.2. Eliminar PaymentProfile

<b>URL:</b>	{ambiente_api}/v1/api/Customer/{customer-id}/PaymentProfileDelete		
<b>Tipo API:</b>	Pública		
<b>Method:</b>	POST		
<b>Autenticación:</b>	PrivateAccountKey		
<b>Parámetros:</b>	<b>Campo</b>	<b>Tipo</b>	<b>Descripción</b>
	PaymentProfileID	Numeric	Identificador del medio de pago a modificar
	HTTP Code		Código HTTP.
	Response	Customer	Objeto que contiene la información del Cliente.
<b>Respuesta:</b>	Errors	List<Error>	Lista de errores que se pueden haber generado.

### 6.5. Objeto CustomerActivation

**CustomerActivation** es obtenido luego de ingresar el código de activación, se utiliza para informar los datos de activación de un perfil de pagos registrado por el cliente.

Campo	Descripción	Tipo	Presencia Default
<b>Token</b>	Identificador del Token asociado al perfil que se desea activar.	String	Mandatorio
<b>ActivationCode</b>	Código de activación recibido por el cliente, asociado al perfil que se desea activar.	String	Mandatorio

### 6.6. Objeto Transactions

El objeto **Transaction** está asociado a un objeto **Purchase** y contiene la información de requerimiento enviado y la respuesta obtenida del medio de pago correspondiente. Además, va asociado a los diferentes estados por los cuales puede pasar una transacción (autorizada, anulada, etc.)

Campo	Descripción	Tipo	Presencia Default
<b>TransactionID</b>	Identificador de la Transacción	Numeric	Solo Lectura
<b>Created</b>	Fecha y hora del momento de creación de la transacción.	TimeStamp	Solo Lectura
<b>TransactionStatusId</b>	Identificador del Status de la transacción al momento de la consulta.	Numeric	Solo Lectura
<b>Status</b>	Estado de la transacción al momento de la consulta	TransactionStatus	Solo Lectura
<b>Description</b>	Descripción del resultado de la transacción	String	Solo Lectura
<b>ApprovalCode</b>	Código de aprobación devuelto por el medio de pago.	String	Solo Lectura
<b>Steps</b>	Lista de los estados intermedios desde que se creó la transacción.	List<TransactionStep>	Solo Lectura



Los campos que forman el Transaction son:

- TransactionID – Identificador de la Transacción.
- Created – TimeStamp del momento de creación de la Transacción.
- Status – Estado de la Transacción al momento de la consulta.
- Steps – Lista de “estados intermedios” desde que se creó la Transacción. Es una lista de objetos “TransactionStep”

## 6.7. Objeto Address

Este objeto se utiliza para obtener los datos de la dirección de un cliente tanto de envío como dirección de facturación. (Shipping Address o Billing Address)

Campo	Descripción	Tipo	Presencia Default
<b>AddressID</b>	Identificador de la Dirección	Numeric	Mandatorio
<b>AddressType</b>	Tipo de Dirección (billing, shipping, customer, etc.).	String	Opcional
<b>Country</b>	País	String	Opcional
<b>State</b>	Estado / Departamento / Provincia	String	Opcional
<b>City</b>	Ciudad	String	Opcional
<b>AddressDetail</b>	Calle, número, etc.	String	Mandatorio
<b>PostalCode</b>	Código postal	String	Opcional

## 6.8. TransactionStep

El objeto TransactionStep se utiliza para informar los datos de los distintos estados por los que paso la Transacción hasta su estado actual.

Campo	Descripción	Tipo	Presencia Default
<b>Step</b>	Nombre del paso ejecutado.	String	Solo Lectura
<b>Created</b>	Fecha y hora del momento de ejecución del paso.	TimeStamp	Solo Lectura
<b>Status</b>	Estado final luego de la ejecución del paso.	Transaction Status	Solo Lectura
<b>ResponseCode</b>	Código de respuesta obtenido luego de la ejecución del paso actual. Contiene por ejemplo el código de respuesta del medio de pago.	String	Solo Lectura
<b>ResponseMessage</b>	Mensaje de respuesta asociado al ResponseCode.	String	Solo Lectura
<b>Error</b>	Código de error generado (si corresponde) en la ejecución del paso.	String	Condiciona
<b>AuthorizationCode</b>	Código de aprobación devuelto por el medio de pago.	String	Solo Lectura
<b>Uniqueld</b>	Identificador único de la llamada (request) que desencadenó la ejecución de este paso. Utilizado para relacionar operaciones Refund y Rollback.	String	Solo Lectura

### 6.9. CountryDataDo

Objeto utilizado para obtener datos de la República Dominicana (Código ISO-3166 = DO).

Campo	Descripción	Tipo	Presencia Default
<b>Invoice</b>	Número de factura asociado a la venta	String	Mandatorio
<b>Tax</b>	Monto de impuestos pagados	Amount	Opcional

### 6.10. Objeto Notification

El objeto Notification se utiliza para enviar actualizaciones de estado u otras notificaciones hacia el comercio sin esperar que este lo solicite. Por ejemplo, en flujos de transacciones donde la misma queda en estado Pendiente y se requieren otras acciones para terminar su procesamiento.

Estas notificaciones serán entregadas al servicio WebHook que debe ser implementado por el comercio de forma de poder procesar estas llamadas.

Campo	Descripción	Tipo	Presencia Default
<b>NotificationId</b>	Identificador de la notificación.	Numeric	Solo Lectura
<b>Created</b>	Fecha y hora de creación de la notificación.	TimeStamp	Solo Lectura
<b>ResourceType</b>	Tipo de Recurso informado. Por ejemplo "Purchase" para informar el estado final de la compra realizada.	ResourceType	Solo Lectura
<b>ResourceUrl</b>	URL donde se puede consultar el recurso informado.	String	Solo Lectura
<b>ResourceObject</b>	Objeto devuelto. Puede contener cualquier tipo de objeto, como un Purchase o un Customer, dependiendo del tipo de notificación y el proceso que la desencadenó.	String	Solo Lectura

## 7. Servicio WebHook del Comercio

El comercio podrá implementar un servicio para recibir y procesar notificaciones enviadas desde el sistema.

Este servicio es necesario en algunos flujos de transacciones donde parte del proceso no puede ser realizado de forma sincrónica, por lo que el estado final de dicha transacción será informado asincrónicamente una vez que la plataforma puedan responder al comercio. El comercio deberá informar a CARDNET la URL en la que será publicado dicho servicio, de forma de que esta sea configurada en nuestros sistemas para el envío de las notificaciones.

## 7.1. Especificaciones del servicio WebHook

El servicio WebHook se trata de un Servicio Web que deberá procesar un request HTTP con las siguientes características:

<b>URL:</b>	<Debe especificarlo el comercio>		
<b>Tipo API:</b>	Pública		
<b>Method:</b>	POST		
<b>Autenticación:</b>	PrivateAccountKey		
<b>Parámetros:</b>	<b>Campo</b>	<b>Tipo</b>	<b>Descripción</b>
	Notification	Notification	Objeto que contiene la información de notificación enviada. Formato: JSON
<b>Respuesta:</b>	HTTP Code		Código HTTP.

La implementación de dicho servicio depende de la plataforma y lenguaje elegidos por el comercio. Los únicos requisitos técnicos son:

- Deberá aceptar mensajes en formato JSON (application/json)
- Deberá validar la clave privada (PrivateAccountKey) enviada como header de autenticación en el mismo formato (Basic Authentication) que el comercio debe enviar en cada uno de los requests.
- Deberá responder solo un código HTTP, donde:
  - Si se responde el código 200 (OK), SiemprePago asumirá que el procesamiento de la notificación fue exitoso.
  - Si se responde cualquier otro código distinto a 200, CARDNET asumirá que el procesamiento fue fallido, por lo que la notificación se reintentará.
- La propiedad ResourceObject del objeto Notification recibido puede contener uno de varios tipos de objetos. Se deberá validar la propiedad ResourceType para saber cómo tratar el ResourceObject. Por ejemplo, si se envía una notificación relacionada al cambio de estado de una compra, el ResourceType corresponderá a Purchase y el ResourceObject se deberá tratar según la definición de dicho tipo de datos.

## 8. Códigos HTTP

Código	Descripción	Descripción
200	Ok	La solicitud fue procesada correctamente.
400	Bad Request	La solicitud está mal formada o falta algún parámetro obligatorio.
401	Unauthorized	Fallo de autenticación.
403	Forbidden	No tiene permisos para realizar la operación solicitada.
404	Not Found	El recurso solicitado no fue encontrado.
405	Method not Allowed	Request por método incorrecto (ej. GET en lugar de POST).
408	Request Timeout	No se pudo completar el pedido en el tiempo máximo configurado.
500	Internal Server Error	Ocurrió un error en el servicio.
503	Service Unavailable	El servicio está en mantenimiento o experimentando problemas de acceso.

### 8.1. Códigos de Error

Código	Mensaje	Descripción
TK...		Errores servicio de Tokenización
TK001	INVALID_CARD_PAN	El número de tarjeta ingresado es incorrecto.
TK002	INVALID_CVV	El número de CVV ingresado es incorrecto.
TK003	INVALID_EXPIRATION_DATE	La fecha de vencimiento de la tarjeta es incorrecta.
TK004	INVALID_SESSION_IDENTIFIER	Se envió un identificador de sesión inválido en una solicitud de token.
TK005	INVALID_EMAIL	Se ingresó un email con formato incorrecto.
TK006	EXPIRED_TOKEN	El token (de tipo One-Time) ya fue utilizado o está expirado.
TK007	INVALID_PAYMENT_MEDIUM	Medio de pago no coincide con el esperado.
TK008	ISSUER_BANK_NOT_MATCH	Banco emisor no coincide con el esperado.
TK009	INVALID_ACTIVATION_CODE	Código de activación de Token inválido.
TK999	UNKNOWN_ERROR	Error desconocido, el detalle puede especificar el error real.
PR...		Errores servicio de Purchase
PR001	INVALID_TOKEN	El token informado es inválido, está vencido o no corresponde al comercio.
PR002	INVALID_ORDER	El número de orden es inválido.
PR003	INVALID_AMOUNT	El monto informado es inválido.
PR004	INVALID_CURRENCY	La moneda informada es inválida.
CS...		Errores servicio Customers.
CS001	INVALID_EMAIL	El email informado es inválido.
TR...		Errores servicio Transactions
TR001	COMMUNICATION_ERROR	Error de comunicación con el servicio adquirente.
TR002	INVALID_TRANSACTION_STATE	La transacción asociada a la compra se encuentra en un estado que no permite la ejecución de la operación actual.

		Esto sucede por ejemplo cuando se quiere realizar una operación "Commit" sobre una Purchase que ya se encuentra autorizada o que fue rechazada.
TR003	ACQUIRER_ACCOUNT_PROBLEM	Problemas con la cuenta de comercio en el Adquirente.
TR004	ACQUIRER_PROXY_ERROR	Error al enviar transacción al Adquirente mediante Proxy.
TR005	ACQUIRER_PROBLEM	Error interno del Adquirente.
TR006	ACQUIRER_DUPLICATED_ORDER	Número de orden duplicada en el Adquirente.
TR007	INVALID_PAYMENT_METHOD	Error con algún dato del medio de pago (número de tarjeta, código de verificación y/o fecha de expiración).
TR008	COMMIT_AMOUNT_GREATER_THAN_AUTHORIZED	El monto que se intenta confirmar es superior al autorizado previamente.
TR009	ACQUIRER_UNKNOWN_ERROR	Error desconocido de Adquirente.
TR999	UNKNOWN	Error no determinado al ejecutar la transacción.
ER...		Errores genéricos.
ER999	UNKNOWN	No determinado.

## 8.2. Códigos de Respuesta de La transacción

Este es el código de respuesta recibo del banco el cual indica si la transacción fue aprobada o rechazada. Este se encuentra en el objeto **TransactionStep**.

RESPONSE CODE	DESCRIPCION CLIENTE
00	Aprobada
01	Llamar al Banco
02	Llamar al Banco
03	Comercio Invalido
04	Rechazada
05	Rechazada
06	Error en Mensaje
07	Tarjeta Rechazada
08	Llamar al Banco
09	Request in progress
10	Aprobación Parcial
11	Approved VIP
12	Transaccion Invalida
13	Monto Invalido
14	Cuenta Invalida
15	No such issuer
16	Approved update track 3
17	Customer cancellation
18	Customer dispute

19	Reintentar Transaccion
20	No tomo accion
21	No tomo acción
22	Transaccion No Aprobada
23	Transaccion No Aceptada
24	File update not supported
25	Unable to locate record
26	Duplicate record
27	File update edit error
28	File update file locked
30	File update failed
31	Bin no soportado
32	Tx. Completada Parcialmente
33	Tarjeta Expirada
34	Transaccion No Aprobada
35	Transaccion No Aprobada
36	Transaccion No Aprobada
37	Transaccion No Aprobada
38	Transaccion No Aprobada
39	Tarjeta Invalida
40	Funciono no Soportada
41	Transacción No Aprobada
42	Cuenta Invalida
43	Transacción No Aprobada
44	No investment account
51	Fondos insuficientes
52	Cuenta Invalididad
53	Cuenta Invalididad
54	Tarjeta vencida
56	Cuenta Invalididad
57	Transaccion no permitida
58	Transaccion no permitida en terminal
60	Contactar Adquirente
61	Excedio Limte de Retiro
62	Tarjeta Restringida
65	Exedio Cantidad de Intento
66	Contactar Adquirente
67	Hard capture
68	Response received too late
75	Pin excedio Limte de Intentos
77	Captura de Lote Invalida

78	Intervencion del Banco Requerida
79	Rechazada
81	Pin invalido
82	PIN Required
85	Llaves no disponibles
89	Terminal Invalida
90	Cierre en proceso
91	Host No Disponible
92	Error de Ruteo
94	Duplicate Transaction
95	Error de Reconciliación
96	Error de Sistema
97	Emisor no Disponible
98	Excede Limete de Efectivo
99	CVV or CVC Error response

## 9. Tipos de datos

### 9.1. Datos Básicos

#### 9.2. String

Es una cadena de caracteres que puede contener cualquier carácter Unicode. Puede definirse un largo máximo que está expresado entre corchetes, por ejemplo string[30] significa que el string puede contener como máximo 30 caracteres. Salvo que en la documentación del dato se especifique lo contrario, si un string tiene más caracteres que el máximo definido se truncará y se continuará el procesamiento.

#### 9.3. Numeric

Campo numérico de valor entero. Puede definirse un largo máximo que está expresado entre corchetes, por ejemplo, Numeric [3] significa que el número puede ser de 3 cifras como máximo. Si el dato posee un valor mayor a la especificación, se devolverá un error.

#### 9.4. Amount

Campo numérico que incluye decimales, para expresar los montos relacionados a una transacción. Estos campos son siempre expresados con la parte entera más 2 decimales sin signos de puntuación entre ambos.

A continuación, se plantean ejemplos de cómo deben codificarse los valores:

Valor	Codificación
<b>100</b>	10000
<b>1.237,52</b>	123752
<b>3.200,5</b>	320050
<b>0.01</b>	1

### 9.5. TimeStamp

Campo que contiene un valor fecha/hora que debe ser expresado en el siguiente formato "YYYY-MM-DDTHH:mm:ss.ttt" donde

Detalle Formato	
YYYY indica el año MM indica el mes – DD indica el día – T indica el inicio de la sección de tiempo requerida hh indica la hora (de 0 a 23) mm indica los minutos ss indica los segundos ttt indica los milisegundos	
Valor	Codificación
2016/01/12 13:21:48.354	2016-01-12T13:21:48.354
2016/03/31 05:17:00.000	2016-03-31T05:17:00.000
2016/11/28 22:59:59.970	2016-11-28T22:59:59.970

### 9.6. Transaction Status

Indica el estado final de la transacción

TransactionStatusId	TransactionStatus
<b>1</b>	Approved
<b>2</b>	Pending
<b>3</b>	Preauthorized
<b>4</b>	Rejected

### 9.7. ActionType

Indica el tipo de acción a realizarse por parte del comercio (parte del CommerceAction).

Valores posibles

Actiontype	Accion
<b>1</b>	Redirect



### 9.8. Medios de Pago

La siguiente tabla identifica los medios de pago aceptados, incluyendo las funcionalidades soportadas por cada uno.

PaymentMediaId	Medio de Pago	Compra	Pre autorización
1	Visa	Si	No
2	MasterCard	Si	No
3	American Express	Si	NO

## 10. Datos de Ambientes

### 10.1. Datos ambientes de pruebas

Para realizar pruebas se deben utilizar los siguiente url:

- **API Servicios:** <https://lab.cardnet.com.do/servicios/tokens/v1/api/objeto> donde objeto se refiere al nombre del objeto que se desea consultar.
- **Formulario de Captura:**
  - <https://lab.cardnet.com.do/servicios/tokens/v1/Capture>
  - <https://lab.cardnet.com.do/servicios/tokens/v1/Scripts/PWCheckout.js>
- **Private Account Key:** 9kYH2uY5zoTD-WBMEoc0KNRQYrC7crPRJ7zPegg3suXguw\_8L-rZDQ\_\_
- **Public Account Key:** mfH9CqiAFjFQh\_gQR\_1TQG\_I56ONV7HQ

**Nota:** Estos datos deben ser utilizados únicamente para pruebas de los servicios, una vez que su aplicación sea certificada por los técnicos de CARDNET, deberá solicitar los datos para procesar en el ambiente de producción.

### 10.2. Datos ambiente de Produccion

Para realizar pruebas se deben utilizar los siguiente url:

- **API Servicios:** <https://servicios.cardnet.com.do/servicios/tokens/v1/api/objeto> donde objeto se refiere al nombre del objeto que se desea consultar.
- **Formulario de Captura:**
  - <https://servicios.cardnet.com.do/servicios/tokens/v1/Capture>
  - <https://servicios.cardnet.com.do/servicios/tokens/v1/Scripts/PWCheckout.js>