

Nombre: Santiago Gaona Carvajal

# PRUEBA ANALISTA DE IMPLEMENTACIÓN PLACETOPAY

---

## PRIMER PUNTO:

---

*Realice el consumo desde postman u otra herramienta para consumo de APIs con pagos básicos y documente el código fuente usado para consumir nuestras APIs, donde suministre la información mínima necesaria para realizar el proceso transaccional (revisar parámetros de entrada del servicio). Se deben compartir las evidencias con las respuestas del flujo transaccional finales (aprobada y rechazada).*

### 1. Herramientas utilizadas

---

Como parte del consumo y envío hacia el servicio Placetopay en un entorno .NET 6, se utilizaron las siguientes herramientas:

- **Lenguaje de programación:** Se empleó C# para simular transacciones aprobadas y rechazadas desde la Tienda/Comercio.
- **Importación de espacios de nombres:** Se incluyeron las siguientes importaciones en el código:

```
using System.Security.Cryptography; // Para crear el SHA1
using Newtonsoft.Json; // Para la serialización y deserialización
// Otras importaciones necesarias
```

- **Herramientas de documentación:**
  - **Markdown:** Se utilizó Markdown como lenguaje de marcado para estructurar y formatear la documentación.
  - **Diagrams.net:** Se empleó Diagrams.net para generar el diagrama que complementa la documentación.

Además, para aceptar pagos en línea se hizo uso de **Webcheckout PlacetoPay**, una solución alojada en Placetopay. Esta implementación permitió realizar pruebas y obtener experiencia práctica en el proceso de postulación.

Para obtener las credenciales de autenticación necesarias antes de consumir la API, se utilizó el siguiente código en formato JSON:

```
{
  "locale": "",
  "auth": {
    "login": "",
    "tranKey": "",
    "nonce": "",
    "seed": ""
  }
}
```

Estos parámetros se obtienen a partir de los requisitos de entrada del servicio Placetopay.

- **Locale:** Se define de acuerdo a la implementación de la forma de pago. En este caso, se utiliza "es\_CO".

Para la autenticación, se utilizan los siguientes valores:

- **Login:** Se emplea el login proporcionado como credenciales de acceso a los servicios, tal como se indica en el documento de prueba.
- **TranKey:** Se genera utilizando una fórmula específica en el código. La operación realizada es: Base64(SHA-1(nonce + seed + secretkey)).
- **Nonce:** Se genera un número aleatorio en Base64 para cada solicitud.
- **Seed:** Se utiliza la fecha actual en formato "yyyy-MM-ddTHH:mm:ssZ".
- **SecretKey:** Se utiliza la clave secreta proporcionada en el documento de prueba.

Estos valores permiten autenticar correctamente las solicitudes a la API de Placetopay y establecer la configuración necesaria para su consumo.

## 2. Ejemplo de código

---

### 3. Parámetros de entrada del servicio

---

A continuación se presentan los parámetros de entrada utilizados en los dos endpoints: Crear sesión (CreateRequest) y Consultar sesión (getRequestInformation).

**Crear sesión (CreateRequest) - Request utilizado:**

```
https://checkout-test.placetopay.com/api/session
```

```
{
  "locale": "",
  "auth": {
    "login": "",
    "tranKey": "",
    "nonce": "",
    "seed": ""
  },
  "payment": {
    "reference": "",
    "description": "",
    "amount": {
      "currency": "",
      "total":
    }
  },
  "expiration": "",
  "returnUrl": "",
  "ipAddress": "",
  "userAgent": ""
}
```

- **Auth:** Corresponde a la autenticación proporcionada anteriormente.
- **Payment:** Contiene la información del pago.
  - **Reference:** Es una referencia de pago generada de forma aleatoria en este código.
  - **Description:** Es una descripción del pago, denominada "prueba" en este ejemplo.
  - **Amount:** Indica la moneda utilizada (COP) y el valor del pago. En los ejemplos se utilizó 10,000 como valor de referencia.

- **Expiration:** Especifica la duración de la sesión de pago. En este ejemplo, se estableció en 5 minutos.
- **ReturnUrl:** Es la URL a la cual se debe redirigir después de la operación de pago, independientemente de su estado (aprobada o rechazada).
- **IpAddress:** Representa la dirección IP utilizada para la transacción. En este caso, se utilizó la IP local como ejemplo.
- **UserAgent:** Indica el agente de usuario utilizado para la transacción. En el ejemplo se utilizó "PlacetoPay Sandbox".

Estos parámetros de entrada permiten configurar adecuadamente la creación de una sesión de pago y son utilizados para realizar las solicitudes correspondientes al servicio Placetopay.

#### Consultar sesión (getRequestInformation) - Request utilizado:

Para consultar una sesión de pago utilizando el endpoint Consultar sesión (getRequestInformation), se realiza una solicitud POST a la siguiente URL:

```
https://checkout-test.placetopay.com/api/session/{requestId}
```

Donde el parámetro requerido es:

- **requestId:** Es el identificador de la sesión de pago, que se obtiene como resultado de la creación de la sesión.

Además, se debe enviar en el cuerpo de la solicitud la autenticación necesaria.

```
{
  "locale": "",
  "auth": {
    "login": "",
    "tranKey": "",
    "nonce": "",
    "seed": ""
  }
}
```

Esta solicitud POST devuelve la siguiente información:

- **requestId**: Corresponde al identificador de la sesión de pago establecido anteriormente.
- **status**: Proporciona un estado general de la transacción realizada, información detallada sobre el estado y resultado de la transacción.

Al realizar esta consulta, se obtiene trazabilidad y se puede obtener información actualizada sobre el estado y detalles de la transacción realizada en la sesión de pago correspondiente.

## 4. Ejemplos de flujo transaccional

Como ejemplo de flujo transaccional se tiene la ejecución de pagos, en este caso se genera un pago aprobado y un pago rechazado, el flujo transaccional se puede ver con más detalle en [Diagrama](#)

### 4.0.1. Pago Aprobado

A continuación se muestra un ejemplo de flujo transaccional para un **Pago Aprobado**:

1. Se realiza una solicitud utilizando los parámetros de *Crear sesión (CreateRequest)* - *Request*, los cuales han sido previamente configurados y completados.

Modo de pruebas. No habrá afectación monetaria a los medios de pago

sgaonacarvajal@outlook.com  
[Cambiar de cuenta](#)

Selecciona un método de pago

Tarjeta de Crédito / Débito / Prepago

Cuentas débito ahorro y corriente (PSE)

[No deseo continuar](#)

Total a pagar

S2319941-T1

\$ 10.000,00 COP

Prueba de Pago

Detalle del pago

Referencia20230527132

Fecha de solicitud2023-05-27 19:22:36

Tu proceso de pago expirará en 4 minutos

2. Se utiliza una tarjeta de prueba proporcionada para simular el pago aprobado. En este caso, se emplea una tarjeta Visa con los siguientes detalles:

- Número de tarjeta: 4110760000000008
- Autenticación 3D Secure: Aprueba 3DS-C

3. La solicitud se envía a la API de Placetopay.

4. La API procesa la solicitud y verifica los datos proporcionados.

5. La tarjeta de prueba utilizada cumple con los requisitos para una transacción aprobada.

Modo de pruebas. No habrá afectación monetaria a los medios de pago

**placetopay**  
by evertec

**Transacción Aprobada**

Método de pago  
**VISA** \*\*\*\* 0008

Total pagado  
**\$ 10.000,00**

Fecha de transacción  
**2023-05-27 19:23:50**

Dirección IP  
**190.67.58.20**

Autorización / CUS  
**000000**

Recibo  
**99933430**

Código Respuesta  
**00**

**Volver al comercio**

El comprobante será enviado a [sgaonacarvajal@outlook.com](mailto:sgaonacarvajal@outlook.com)

Total pagado S2319941-T1

**\$ 10.000,00** COP

Prueba de Pago

**Imprimir**

**Detalle del pago**

Referencia **20230527132**

Fecha de solicitud **2023-05-27 19:22:36**

El proceso de pago se terminó hace menos de un minuto

Obtén ayuda en [pruebas2p@yopmail.com](mailto:pruebas2p@yopmail.com)

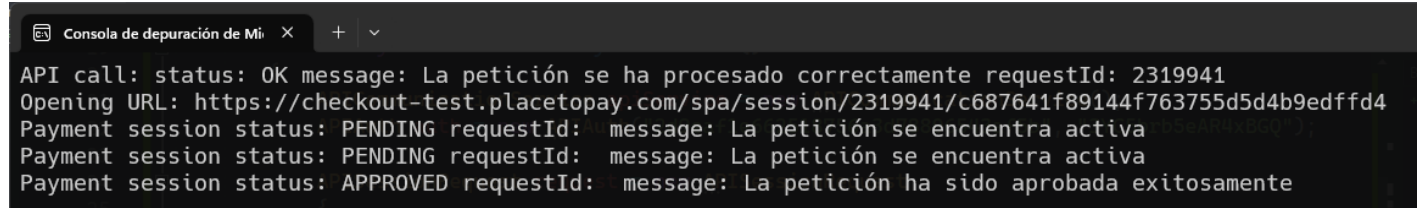
Hecho por **placetopay** by evertec Español (Colombia)

6. Como resultado, se recibe una respuesta indicando que el pago ha sido aprobado.

7. Vuelve al comercio dada la **returnUrl**

Este ejemplo de flujo transaccional representa el caso de un pago exitoso que ha sido aprobado correctamente según los parámetros y la tarjeta utilizada para la simulación, esto se puede ver en la

consola como:

A screenshot of a web browser's developer console. The console has a dark background with light-colored text. At the top, there's a tab labeled 'Consola de depuración de Mi' with a close button (X) and expand/collapse icons (+ and -). Below the tab, there are four log entries: 1. 'API call: status: OK message: La petición se ha procesado correctamente requestId: 2319941' 2. 'Opening URL: https://checkout-test.placetopay.com/spa/session/2319941/c687641f89144f763755d5d4b9edffd4' 3. 'Payment session status: PENDING requestId: message: La petición se encuentra activa' 4. 'Payment session status: PENDING requestId: message: La petición se encuentra activa' 5. 'Payment session status: APPROVED requestId: message: La petición ha sido aprobada exitosamente' The text is left-aligned and uses a monospaced font.

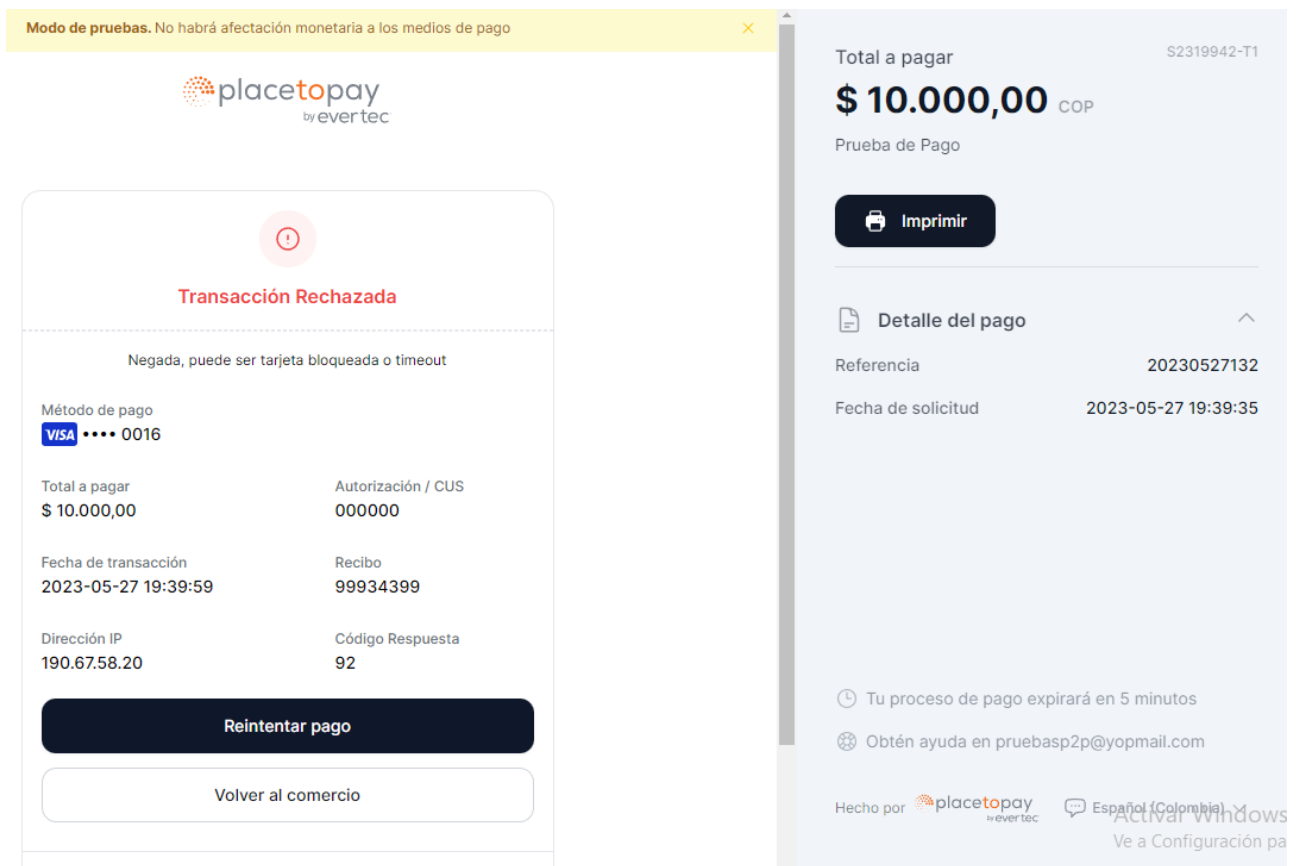
```
API call: status: OK message: La petición se ha procesado correctamente requestId: 2319941
Opening URL: https://checkout-test.placetopay.com/spa/session/2319941/c687641f89144f763755d5d4b9edffd4
Payment session status: PENDING requestId: message: La petición se encuentra activa
Payment session status: PENDING requestId: message: La petición se encuentra activa
Payment session status: APPROVED requestId: message: La petición ha sido aprobada exitosamente
```

Donde se observa el requestId y bajo éste requestId se muestra cada 30 segundos el status de la transacción hasta ser Aprobada.

#### 4.0.1. Pago Rechazado

A continuación se muestra un ejemplo de flujo transaccional para un **Pago Rechazado**:

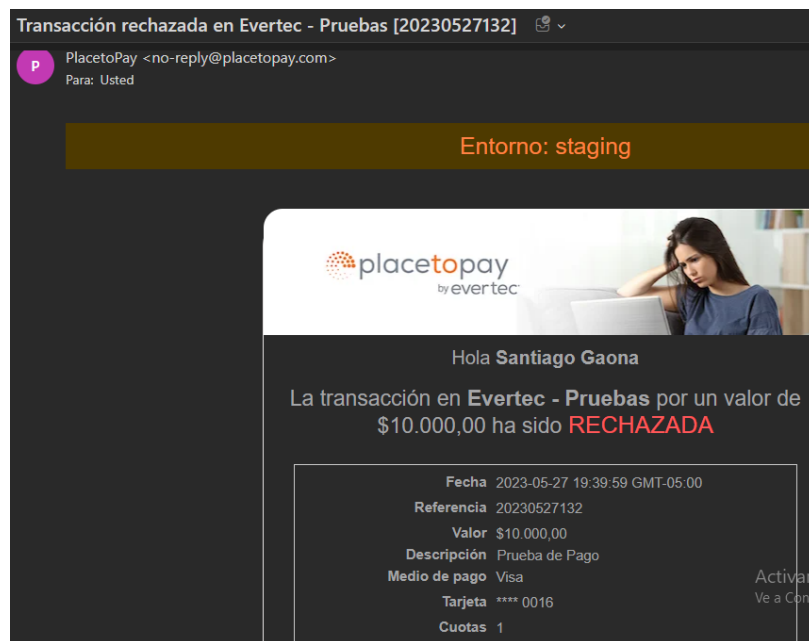
1. Se realiza una solicitud utilizando los parámetros de *Crear sesión (CreateRequest) - Request*, los cuales han sido previamente configurados y completados.
2. Se utiliza una tarjeta de prueba proporcionada para simular el pago rechazado. En este caso, se emplea una tarjeta Visa con los siguientes detalles:
  - Número de tarjeta: 4110760000000016
  - Autenticación 3D Secure: Rechaza



3. La solicitud se envía a la API de Placetopay.
4. La API procesa la solicitud y verifica los datos proporcionados.
5. La tarjeta de prueba utilizada no cumple con los requisitos para una transacción aprobada y es rechazada.
6. Como resultado, se recibe una respuesta indicando que el pago ha sido rechazado.
7. Vuelve al comercio dada la ***returnUrl***.

Este ejemplo de flujo transaccional representa el caso de un pago que ha sido rechazado debido a los detalles específicos de la tarjeta utilizada para la simulación de números de tarjeta de pruebas.





## Segundo punto:

### 4.1. ¿Qué es el RequestId y para qué sirve?

El `RequestId` es un identificador único asociado a cada transacción realizada con PlacetoPay. Este identificador sirve para rastrear y obtener información sobre el estado y los detalles de una transacción en particular. Este `RequestId` se puede utilizar en la consulta de la sesión (`getRequestInformation`) para obtener la información correspondiente a la transacción.

```
{
  "requestId": 1,
  "status": {
    "status": "APPROVED",
    "reason": "00",
    "message": "La petición ha sido aprobada exitosamente",
    "date": "2023-05-26T15:49:47-05:00"
  }
}
```

4.2. ¿Cuáles son los estados de una transacción? Explica significado de cada una de ellas.

Los estados de una transacción en el sistema PlacetoPay son los siguientes:

- **APPROVED** : Indica que la transacción ha sido aprobada exitosamente.
- **PENDING** : El estado de la transacción está pendiente, probablemente esperando la confirmación del pago.
- **REJECTED** : La transacción ha sido rechazada.
- **APPROVED\_PARTIAL** : Parte de la transacción ha sido aprobada, posiblemente debido a una transacción parcialmente exitosa.
- **PARTIAL\_EXPIRED** : Una transacción parcialmente exitosa ha expirado.
- **FAILED** : La transacción ha fallado, es posible que por algún error en el proceso.

### 4.3. Explicar las diferencias entre cobro por suscripción (token) y cobro por recurrencia.

El **cobro por suscripción (token)** en PlacetoPay se refiere a una forma de procesar los pagos en la que se obtiene un token para un instrumento de pago específico, como una tarjeta de crédito o débito. Este token es almacenado y utilizado para futuros cobros sin necesidad de volver a ingresar los datos de la tarjeta. Es útil para los clientes que desean realizar compras frecuentes o suscripciones en un sitio específico sin tener que ingresar la información de su tarjeta cada vez.

Ejemplo de una solicitud de cobro usando un token:

```
{
  "instrument": {
    "token": {
      "token": "e07ca9986cf0ecacEjempl057fa11c07bf37ea35e9e3e3a4180c49"
    }
  }
}
```

El **cobro por recurrencia**, por otro lado, se refiere a cobros que se realizan en un intervalo específico (diario, semanal, mensual, anual, etc.) a un cliente. Puede servir como método para las suscripciones de servicios que requieren pagos regulares. En este caso, es necesario especificar la periodicidad, el intervalo, la fecha del próximo pago y otras condiciones específicas de la recurrencia.

Ejemplo de una solicitud de cobro recurrente:

```
{
  "recurring": {
    "periodicity": "M",
    "interval": 1,
    "nextPayment": "2023-06-27",
    "maxPeriods": 12
  }
}
```

---

## Tercer Punto

---

### 5. Explicación de la situación

El error "Autenticación fallida 103" generalmente ocurre debido a una discrepancia de tiempo. Según la documentación proporcionada, este error se genera cuando la 'semilla' (timestamp o fecha y hora) que se envía para la autenticación tiene una diferencia mayor a 5 minutos respecto al tiempo real GMT o local con zona horaria. Es esencial que la sincronización del reloj del servidor esté correctamente configurada para evitar este problema.

Un aspecto importante que encontré durante las pruebas es la necesidad de utilizar el formato de fecha y hora correcto. Durante una prueba, experimenté un problema similar que se resolvió cuando cambié el formato de fecha y hora a:

```
seed = DateTime.UtcNow.ToString("yyyy-MM-ddTHH:mm:ssZ");
```

#### 5.1. Respuesta a remitir (Tipo Correo)

Estimado cliente,

Lamentamos los inconvenientes que ha estado experimentando con las transacciones. Tras revisar el problema, hemos identificado que se está produciendo un error de autenticación, específicamente "Autenticación fallida 103".

Este error ocurre cuando la 'semilla' o timestamp que se envía durante el proceso de autenticación presenta una discrepancia de tiempo mayor a 5 minutos respecto al tiempo real GMT o local con

zona horaria. Es vital que la sincronización del reloj de su servidor esté correctamente configurada para evitar este problema.

Para identificar y solucionar la causa raíz del problema, le recomendamos realizar las siguientes validaciones:

1. **Sincronización del reloj del servidor:** Verifique que la hora de su servidor esté correctamente sincronizada con un servicio de tiempo de red (NTP) para mantener la precisión del reloj. La semilla o timestamp enviada durante la autenticación debe ser una fecha en formato ISO 8601 y debe estar dentro de los 5 minutos de la hora actual GMT o local con zona horaria.
2. **Generación del hash tranKey:** Asegúrese de que el hash tranKey se esté generando correctamente. Este valor se genera utilizando la operación: Base64(SHA-1(nonce + seed + secretKey)). Debe garantizar que el nonce, la seed (fecha y hora) y la secretKey se estén combinando correctamente para generar el hash tranKey.
3. **Datos de autenticación:** Revise que los datos de login, tranKey, nonce y seed se estén enviando correctamente en la estructura auth. Todos estos elementos deben estar presentes y ser correctos para autenticarse con éxito.

Por favor, realice estas validaciones y si el problema persiste, no dude en ponerse en contacto con nosotros para que podamos asistirle más a fondo.

Saludos cordiales,

Santiago Gaona Carvajal Analista de implementación - Evertec Placetopay - Pasarela de pagos digitales

---

## Diagrama de flujo del proceso de pago

---

