



Guía

INTEGRACIÓN DEL BOTÓN DE PAGO

CCR-GRIN-GUI-0002

Tabla de Contenido

1. Introducción.....	3
2. Objetivo General.....	3
3. Alcance.....	3
4. Consideraciones.....	4
5. Desarrollo.....	5
5.1 Flujo General.....	5
5.2 Pasos para la integración.....	6
5.2.1 Obtención de credenciales.....	6
5.2.2 Descarga de código base para integraciones web (Recomendada).....	7
5.2.3 Creación de un pago.....	8
I. Ruta para creación de pago.....	9
II. Headers.....	9
III. Body.....	9
IV. Response (200).....	10
5.2.4 Formulario de pago.....	12
5.2.5 Procesamiento del pago.....	13
5.2.6 Validación del estado del pago.....	14
I. Ruta para validación de pago.....	14
II. Headers.....	14
III. Parámetros del Query.....	15
IV. Response (200).....	15
V. Descripción de estados de respuesta del pago.....	15
6. Glosario.....	16
7. Acrónimos.....	18

1. Introducción

El botón de pago de Consorcio Credicard C.A., en adelante Credicard, permite a los comercios integrar un flujo de pago seguro, eficiente y dinámico. Este botón está diseñado para facilitar transacciones y gestionar respuestas a través de mensajes y eventos, brindando una experiencia de integración sencilla y funcional.

2. Objetivo General

Proporcionar una guía que facilite la integración de este medio de pago en la plataforma del cliente, indicando todos los pasos y requisitos necesarios para la integración. Si la integración es exitosa el usuario podrá recibir pagos con tarjetas de débito y crédito directo en sus cuentas.

3. Alcance

La integración funciona en cualquier tipo de aplicación sea web o móvil, y proporciona el código básico necesario para iniciar la integración en las tecnologías más populares como **React, Vue, Angular, HTML con JavaScript**. En dispositivos móviles la integración es a través de la tecnología de **WebViews** disponibles en **Android, iOS, Flutter y React Native**.

4. Consideraciones

4.1 Antes de iniciar con la integración se deben cumplir los siguientes requisitos, si alguno de ellos no se cumple la integración no se completará de forma satisfactoria.

- a) El comercio debe poseer un número de afiliado activo y válido.
- b) Cuenta activa en el portal de Credicard para el botón de pago.
- c) Un terminal activo para transacciones con tarjetas de débito y crédito.
- d) Credenciales **SecretID** y **ClientID** disponibles en la plataforma en el botón de “Mostrar Credenciales”.

Nota: La gestión de estas credenciales y el acceso al portal es gestionado a través de los canales regulares de Credicard.

4.2 Tiempo de Vida del Modal: Se limita el tiempo de sesión a 5 minutos para mayor seguridad.

4.3 Cierre Prematuro: Asegúrese de manejar los eventos generados durante el proceso de pago de forma correcta.

4.4 Flutter: Se recomienda utilizar la librería Flutter WebView para manejar el formulario de pago obtenido en la respuesta de la creación de orden de pago, ya que con esta librería se puede escuchar el evento “**message**” del formulario y enviarlo al código de Flutter utilizando el método “**addJavaScriptChannel**” de la librería.

5. Desarrollo

5.1 Flujo General

Comienza cuando el cliente crea una solicitud de pago que genera un URL de compra web, ésta contiene el formulario de pago encargado de manejar las validaciones, datos y seguridad, conectado directamente a los servidores de Credicard.

En este formulario el usuario agrega los datos de su tarjeta, y son enviados a Credicard para el procesamiento de la compra, y a través de eventos y llamadas API se le informa al cliente el estado del pago.

Los pagos poseen un único **payment_id** que es el valor encargado de identificar de forma única el pago generado. Este pago tiene un tiempo de vida de 5 minutos de validez.

En el diagrama se muestra la secuencia que sigue el proceso de pago donde el "Cliente" se refiere a la página web o aplicación móvil del integrador, el "Backend Cliente" al **backend** o servidor del integrador y por último "Backend Botón de Pago" a los servidores encargados de procesar el pago en Credicard.

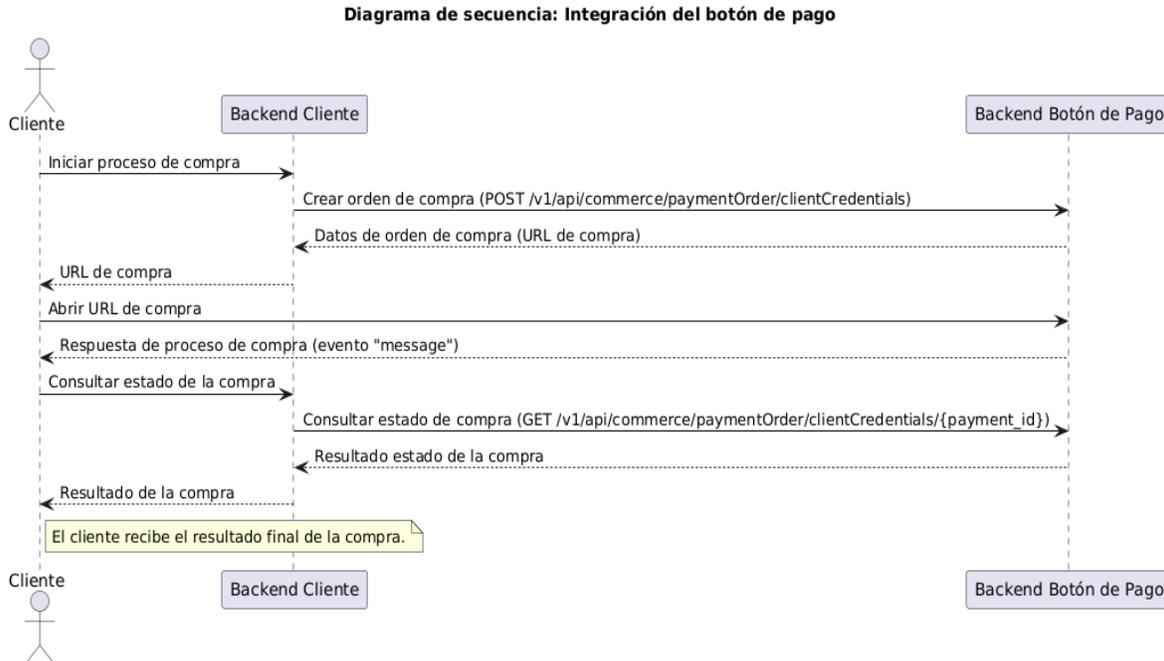


Diagrama de secuencia de las llamadas para la integración del botón de pago

5.2 Pasos para la integración

Como se comentó anteriormente, la integración se puede aplicar en cualquier aplicación móvil o web. Los pasos varían un poco dependiendo de la plataforma, pero en general se cumplen los mismos principios en todos los casos.

Variaciones de la implementación de la integración:

- a) Integración por código base descargado de la plataforma (Recomendado).
- b) Integración móvil utilizando WebViews.
- c) Integración web sin código inicial.

A continuación, se detallan los pasos que debe cumplir el integrador para lograr hacer cobros a través del botón de pago de Credicard.

5.2.1 Obtención de credenciales

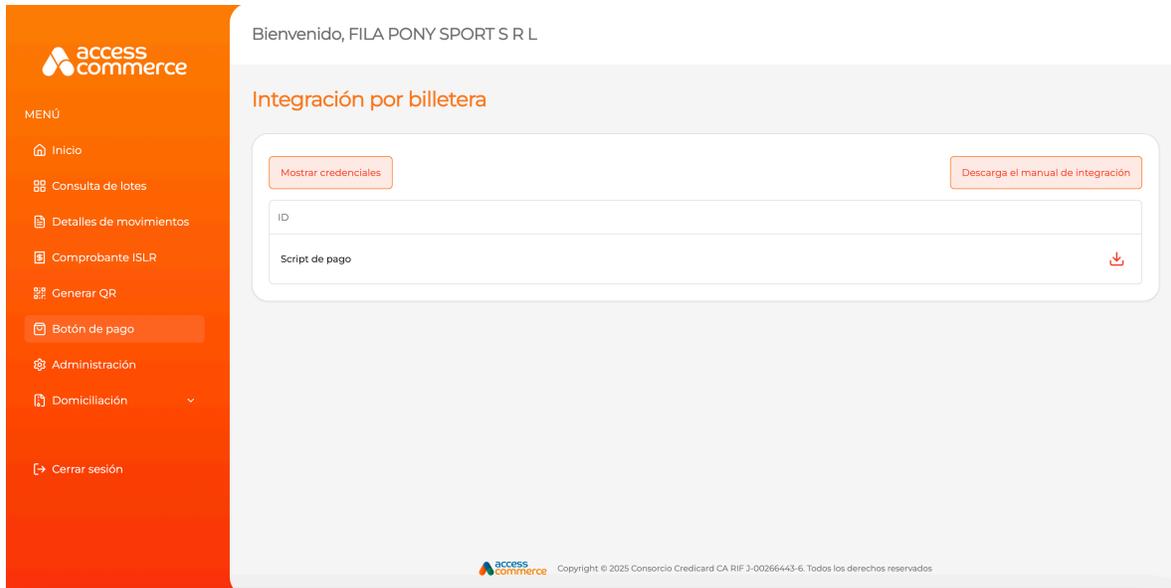
Se deben buscar y guardar las credenciales de acceso **SecretID** (*client-secret*) y **ClientID** (*client-id*), las cuales son necesarias para identificar al comercio de manera segura al momento de generar pagos. Sin estas credenciales no se podrá procesar pagos a través de la integración.

Cabe destacar que, para obtener estas credenciales, es indispensable estar afiliado al sistema de Access Commerce.

Para gestionar su afiliación y obtener sus credenciales debe seguir los siguientes pasos:

- a) En el portal, dirjase al módulo → Botón de pago.
- b) En esta sección, encontrará la opción (un botón) para visualizar las credenciales existentes. De ser la primera vez que accede a la aplicación, podrá tanto gestionar su afiliación, como generar un nuevo par de credenciales al hacer clic en el botón “Mostrar Credenciales”.

Nota: Para obtener las credenciales de acceso de forma satisfactoria es necesario que el comercio haya completado satisfactoriamente el proceso de afiliación con Credicard. Si el comercio no está afiliado, el sistema mostrará un mensaje notificando que la afiliación se encuentra pendiente y que debe ser finalizada antes de obtener las credenciales.



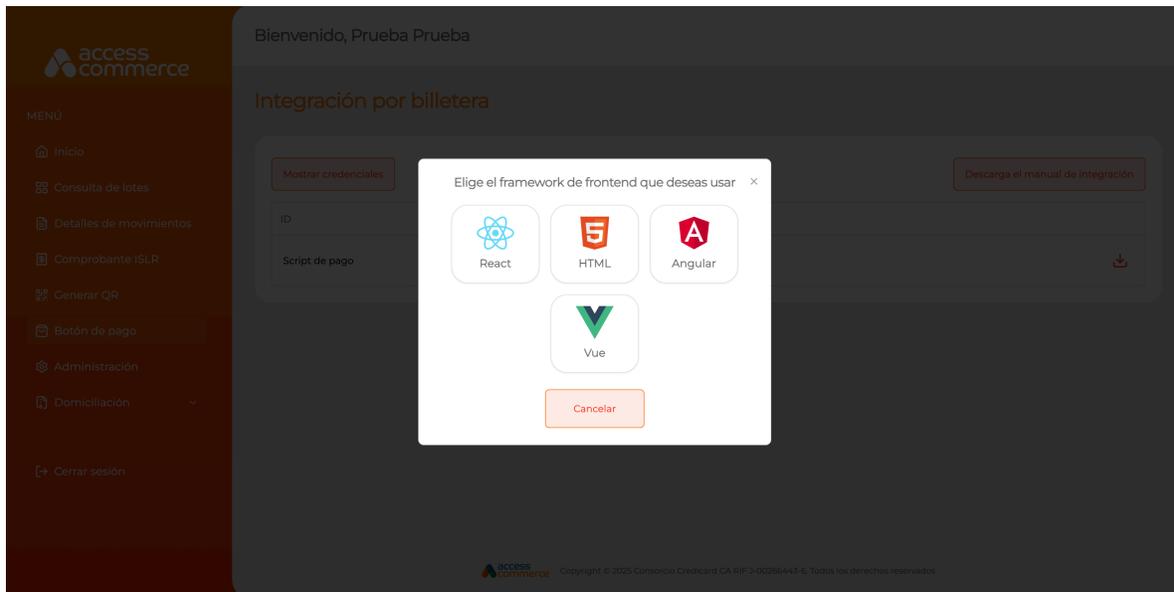
Pantalla donde presionando “Mostrar credenciales” se obtiene el **SecretID** (*client-secret*) y **ClientID** (*client-id*)

5.2.2 Descarga de código base para integraciones web (Recomendada)

Para las tecnologías webs más populares (**React**, **Vue**, **Angular**, **HTML con JavaScript**) se puede descargar un **script** con el código base inicial que simplifica el flujo de integración, ya que, el integrador solo debe descargar el código de su tecnología, pegarlo en su proyecto y sustituir los llamados a su servidor en los lugares marcados en el código.

Los códigos proporcionados están diseñados para simplificar el proceso de integración, minimizar errores y resolver posibles dudas que puedan surgir de manera estandarizada, por esto, es la manera recomendada de hacer la integración.

Para descargar el código debe ir a la sección “Integración por Billetera” y seleccione el ícono de descarga de Script de Pago, se abrirá un modal preguntando por la tecnología usada en su aplicación.



Tecnologías disponibles para la integración rápida utilizando código base

5.2.3 Creación de un pago

Indiferentemente, si la integración ocurre iniciando el código base o se decide hacer de forma manual, se debe crear una orden de pago llamando a los servidores de Credicard, si la respuesta es exitosa el servidor devolverá la URL de pago junto con otros valores relacionados al pago.

En el caso del código inicial, se muestra dónde debe modificarse el **script** para hacer este llamado directo a los servidores del cliente integrador.

Integración del Botón de Pago

I. Ruta para creación de pago

POST /v1/api/commerce/paymentOrder/clientCredentials

II. Headers

Parámetro	Descripción	Tipo	Ejemplo
client-secret	SecretID obtenido en la plataforma de Access Commerce	String	"2963a620-cfc8-11ef-998b-01010506d"
client-id	ClientID obtenido en la plataforma de Access Commerce	String	"2963a620-cfc8-11ef-998b-74896531"

III. Body

Parámetro	Descripción	Tipo	Ejemplo
email	Correo del comercio que recibe el pago	String	"comerciotest@yopmail.com"
amount	Monto en bolívares de la orden de pago.	Number	34.43
concept	Concepto de la orden de pago generada	String	"Pago de prueba"
identification	Campo opcional utilizado en pagos donde se requiere que el titular de la tarjeta coincida con un número de identificación específico	String (opcional)	V18913353

IV. Response (200)

Parámetro	Descripción	Tipo	Ejemplo
url	Ruta que contiene el formulario de pago junto con el id de pago	String	"https://{base_url}/open/pay/fast/checkout?payment=7aa68c5e-db8e-4edf-a706-4ac84124bb91"
paymentId	Id del pago	String	"7aa68c5e-db8e-4edf-a706-4ac84124bb91"
message	Mensaje de respuesta al crear la orden	String	"Orden creada con éxito"

Este endpoint devolverá la URL con un pago único el cual tendrá un tiempo de expiración de 5 minutos y solo podrá ser usado una vez; en caso contrario, será necesario crear un pago nuevo en el sistema.

Nota: la URL base `{base_url}` viene definida por: gateway.credicard.com.ve

Para poder acceder a este endpoint es necesario enviar en el header de la solicitud las credenciales (client-id y client-secret), de lo contrario arrojará un error **401 de unauthorized**.

Importante: Por motivos de seguridad el llamado al endpoint anterior debe ser realizado de servidor a servidor, evitando así pasar por el frontend cualquier tipo de data sensible de ser alterada.

Solo se contará con un intento de procesar el pago, de manera que si la ventana de pagos es cerrada, o el pago se procesa de forma exitosa o falla, ya no podrá usarse la URL de pago generada, y se requiere comenzar el flujo desde cero para hacer la creación de una nueva orden de pago con una URL nueva.

Ejemplo de request

```
{
  "email": "comerciotest@yopmail.com",
  "amount": 10.01,
  "concept": "Pago tienda"
}
```

Ejemplo de respuesta exitosa

```
{
  "url": "https:{base_url}/open/pay/fast/checkout?payment=7aa68c5e-db8e-4edf-a706-4ac84124bb91",
  "paymentId": "7aa68c5e-db8e-4edf-a706-4ac84124bb91",
  "message": "Orden creada con éxito"
}
```

Integración del Botón de Pago

5.2.4 Formulario de pago

La URL obtenida en la respuesta contiene la dirección del formulario de pago asociado al monto creado que debe abrir el usuario en su aplicación para iniciar el flujo de pago. Si es a través del código inicial se le debe pasar a la función que maneja la respuesta de la llamada los datos de la solicitud anterior.

Dependiendo del tipo de integración el usuario podrá ver dos tipos de ventanas al iniciar el proceso de pago en su página web o aplicación:

- Si la integración es con el código inicial descargado de la plataforma, al usuario se le abrirá una nueva ventana en forma de PopUp que mostrará el formulario para ingresar sus datos de tarjeta, y así completar el pago. Este PopUp solo durará activo 5 minutos.
- Si la integración es manual en web o en una aplicación móvil, el usuario verá el formulario en la forma que prefiera el integrador, ya que, es el integrador el que decide cómo abrir la URL de respuesta del pago.

Nota: Si bien, en web el portal de pagos se abre a través de un PopUp, la integración con aplicaciones móviles se debe realizar por medio de un webview.



Ejemplo de formulario de pago

5.2.5 Procesamiento del pago

Una vez que el usuario proporcione la información de su tarjeta y continúe con el pago, Credicard validará los datos y notificará el resultado del mismo. El integrador deberá escuchar el resultado de la transacción que ocurre en el formulario de pago.

El resultado de la transacción es emitido a través del evento **“message”** y debe ser escuchado por el integrador para obtener el estado del pago.

- Si se está haciendo la integración recomendada, el integrador solo debe sustituir el código en los eventos pre-creados en el **script**, y allí escuchar la respuesta inicial sobre el estado del pago.
- Si la integración es manual o en dispositivos móviles, el integrador debe crear sus propios eventos escuchando el evento **“message”** emitido por el formulario de pago.

El estado tendrá los siguientes valores:

- **payment-success-cc**: El pago fue exitoso.
- **payment-fail-cc**: El pago falló.
- **payment-cancelled-cc**: Cierre del modal antes de completar el pago.

Para el manejo de evento **“message”** se puede implementar algo como lo siguiente:

```
addEventListener("message", function (event) {  
  if (event.data.status === "payment-success-cc") {  
    // TODO: Función a ejecutar en caso de pago exitoso  
    handlePaymentSuccessful()  
  }  
});
```

En algunos casos puede ser **window.addEventListener** y en otros **addEventListener** dependiendo de la implementación.

Ejemplo de respuesta en el evento “message”

```
{
  "status": "payment-success-cc"
}
```

Luego de esto, es recomendable validar el estado del pago, como se explica en el siguiente paso, contra el servidor de Credicard, el estado real del pago, ya que, al ser un evento ejecutado en front, es susceptible a posibles modificaciones.

5.2.6 Validación del estado del pago

A continuación, le corresponderá al integrador hacer el llamado a la ruta de validación de pago utilizando el id del pago ejecutado (**payment_id** obtenido en la respuesta de creación del pago), aunque en el paso anterior ya se tiene un estado del pago, al ser este un evento del **frontend** es susceptible a modificaciones malintencionadas y lo correcto es hacer la llamada al servidor de Credicard sobre el estado del pago.

I. Ruta para validación de pago

GET /v1/api/commerce/paymentOrder/clientCredentials/{payment_id}

II. Headers

Parámetro	Descripción	Tipo	Ejemplo
client-secret	SecretID obtenido en la plataforma de Access commerce	String	"2963a620-cfc8-11ef-998b-01010506d"
client-id	ClientID obtenido en la plataforma de Access commerce	String	"2963a620-cfc8-11ef-998b-74896531"

III. Parámetros del Query

Parámetro	Descripción	Tipo	Ejemplo
payment_id	Id del pago al que se le quiere conocer el estado obtenido al momento de la creación del pago	String	"2963a620-cfc8-11ef-998b-01010506d"

IV. Response (200)

Parámetro	Descripción	Tipo	Ejemplo
status	Estado del pago	String	Posibles respuestas: "PENDING", "PAID", "DECLINED", "CANCELLED"
paymentId	Id del pago	String	"7aa68c5e-db8e-4edf-a706-4ac84124bb91"
message	Mensaje de respuesta al crear la orden	String	"Orden creada con éxito"

V. Descripción de estados de respuesta del pago

Estado	Descripción
PENDING	El pago sigue en estado pendiente y no se ha procesado
PAID	El pago se procesó de manera exitosa
DECLINED	La transacción de pago falló
CANCELLED	El proceso de pago fue cancelado

6. Glosario

API:	Interfaz de programación de aplicaciones que permite la comunicación entre sistemas.	A
Backend:	Parte de la aplicación que maneja la lógica del servidor y la base de datos.	B
Body:	Cuerpo de una solicitud HTTP que contiene los datos enviados.	
CANCELLED:	Estado de un pago que ha sido cancelado.	C
ClientID (client_id):	Identificador único del comercio en la plataforma.	
DECLINED:	Estado de un pago que ha sido rechazado.	D
Endpoint:	Punto de acceso a una API o servicio web.	E
Evento "message":	Evento que se dispara cuando se recibe un mensaje en el frontend.	
Flutter:	Framework de desarrollo de aplicaciones móviles.	F
Frontend:	Parte de la aplicación con la que interactúa el usuario.	
Headers:	Encabezados de una solicitud HTTP que contienen metadatos.	H
Integración:	Proceso de conectar y configurar sistemas para que funcionen juntos.	I
Modal:	Ventana emergente que se muestra sobre la página principal.	M

		P
PAID:	Estado de un pago que ha sido procesado exitosamente.	
Payment_id:	Identificador único de un pago.	
PENDING:	Estado de un pago que aún no ha sido procesado.	
PopUp:	Ventana emergente que se abre sobre la página principal.	
		R
Response:	Respuesta de una solicitud HTTP.	
		S
SecretID (client_secret):	Credencial secreta utilizada para autenticar al comercio.	
		W
WebView:	Componente que permite mostrar contenido web dentro de una aplicación móvil.	

7. Acrónimos

API:	Application Programming Interface.
GET:	Método HTTP para solicitar datos del servidor.
HTTP:	Hypertext Transfer Protocol.
ID:	Identificador.
JSON:	JavaScript Object Notation (formato de intercambio de datos).
POST:	Método HTTP para enviar datos al servidor.
SDK:	Software Development Kit.
UI:	User Interface (Interfaz de Usuario).
URL:	Uniform Resource Locator.
UX:	User Experience (Experiencia de Usuario).