



# GUÍA DE INTEGRACIÓN

---

PUNTO WEB 2.0

# Índice

<b>1. INTRODUCCIÓN.....</b>	<b>3</b>
1.1    Objetivo.....	3
1.2    Catálogo de APIs .....	3
1.3    Pasos para integrar .....	4
<b>2. INTEGRACIÓN.....</b>	<b>5</b>
2.1    Paso 1: Credenciales .....	5
2.2    Paso 2: Accesos al panel.....	5
2.3    Paso 3: Generar token de sesión .....	5
2.4    Paso 4: Invocar al formulario.....	7
<b>3. TARJETAS DE PRUEBAS.....</b>	<b>12</b>

## 1. INTRODUCCIÓN

### 1.1 Objetivo

Este documento detalla los lineamientos técnicos necesarios para la integración con la nueva plataforma de pagos PuntoWeb (APIs y Formulario Web).

### 1.2 Catálogo de APIs

APIs	Descripción de APIs
<b>Seguridad</b>	Generación de Token de Sesión
<b>Autorización</b>	Autorización (incluye Pre-Autorización)
<b>Validación decuenta</b>	Validación de Cuenta
<b>Confirmación</b>	Confirmación
<b>Anulación</b>	Anulación
<b>Tokenización</b>	Creación de Token de tarjeta
	Obtención de Token de tarjeta
	Obtención de Tokens de comprador
	Eliminación de Token
<b>Deposito</b>	Deposito
<b>Info. Orden</b>	Consulta de información de la Orden
<b>Planes y Cuotas</b>	Consulta de Planes y Cuotas
<b>Link de Pago</b>	Consultar Detalle Link de pago
	Crear Link de Pago
	Actualizar Link de Pago
	Activar/Desactivar Link de Pago

### 1.3 Pasos para integrar

Para realizar la integración se tiene que seguir los siguientes pasos:

1. Obtener las credenciales (Merchant Code y ApiKey)
2. Accesos al panel (ambiente sandbox/producción)
3. Generar token de sesión
4. Invocar al formulario
5. Identificar otras Apis a utilizar (según corresponda el flujo interno del comercio)



## 2. INTEGRACIÓN

### 2.1 Paso 1: Credenciales

```
MERCHANT_CODE = '4004353'
PUBLIC_KEY = "VErethUtraQuxas57wuMuquprADrAHAb"
CURRENCY = "PEN"
```

### 2.2 Paso 2: Accesos al panel

URL: <https://sandbox-panel.izipay.pe/login/>  
 USUARIO: sandboxizipay@gmail.com  
 CLAVE: L1m@2025

### 2.3 Paso 3: Generar token de sesión

El token de sesión generado es enviado a cada API como medida de autenticación.

#### Definiciones

Método HTTP	POST
Protocolo	REST
Formato	JSON
Ambiente de Sandbox	<a href="https://sandbox-api-pw.izipay.pe/security/v1/Token/Generate">https://sandbox-api-pw.izipay.pe/security/v1/Token/Generate</a>
Ambiente de producción	<a href="https://api-pw.izipay.pe/security/v1/Token/Generate">https://api-pw.izipay.pe/security/v1/Token/Generate</a>

#### Cabecera de entrada (HTTP Headers)

Nombre	Descripción	Tipo y Longitud	Obligatorio
transactionId	Id único por cada transacción.	Alfanumérico Longitud: 5-40	SI

#### Parámetros de entrada (HTTP Body)

Nombre	Descripción	Tipo y Longitud	Obligatorio
requestSource	Origen de la petición, Valor por defecto "ECOMMERCE".	Alfanumérico Longitud: 7-15	SI
merchantCode	Código del comercio.	Alfanumérico Longitud: 7-15	SI
orderNumber	Número de Pedido de la operación.	Alfanumérico Longitud: 5-15	SI
publicKey	Llave pública.	Alfanumérico Longitud: 16 -400	SI
amount	Monto de la operación. (Valor por defecto=0.00)	Decimal Longitud: 4-10	SI

## Ejemplo petición

```
{
  "RequestSource": "ECOMMERCE",
  "merchantCode": "4004345",
  "OrderNumber": "R202211101518",
  "PublicKey": "VErethUtraQuxas57wuMuquprADrAHAb",
  "Amount": "0.00"
}
```

## Parámetros de salida

Nombre	Descripción	Tipo y Longitud	Obligatorio
code	Código de respuesta.	Numérico Longitud: 2-3	SI
message	Mensaje de respuesta.	Alfanumérico Longitud: 5-30	SI
response	Objeto Response	Objeto	NO
response.token	Token generado, por transacción.	Alfanumérico Longitud: 255	NO
response.userOrg	(De uso futuro)	-	NO
response.userScoring	(De uso futuro)	-	NO

## Ejemplo respuesta

```
{
  "Code": "00",
  "Message": "OK",
  "Response": {
    "Token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9eyJtZXJjaGFudENvZGUiOiI0MDAwOTAxliwiZmFjaWxpdGF0b3JDb2RljoimCIsInRyYW5zYWN0aW9uSWQiOijSMjAyMjExMTAxNTE4IiwiT3JkZXJodW1iZXliOijSMjAyMjExMTAxNTE4IiwiQW1vdW50IjoiMC4wMClsIRva2VuSWQiOijkOGVkNmM2NS05OGQ4LTQ3OWEtOGQ1MC05NWMM0Mml0ZjhmnjAiLCJuYmyiQjE2Nju2MTYzMzYslmV4cCl6MTY2NTcwMjczNiwiWF0ljoxNjY1NjE2MzM2fQ.pPe3HgZwmxn9M90jSX_PWb5QeUhwDj_DeOJX59Q2Pk",
    "UserOrg": "",
    "UserScoring": ""
  }
}
```

## 2.4 Paso 4: Invocar al formulario

Agrega la SDK *Izipay* a tu proyecto:

SDK Izipay	
Ambiente de Sandbox	<script src="https://sandbox-checkout.izipay.pe/payments/v1/js/index.js"></script>
Ambiente de Producción	<script src="https://checkout.izipay.pe/payments/v1/js/index.js"></script>

### 2.4.1 Preparar el objeto de config

La configuración básica está compuesta por los siguientes datos de entrada:

```
const iziConfig = {
    publicKey: PUBLIC_KEY,
    config: {
        transactionId: TRANSACTION_ID,
        action: 'pay',
        merchantCode: MERCHANT_CODE,
        order: {
            orderNumber: ORDER_NUMBER,
            currency: ORDER_CURRENCY,
            amount: ORDER_AMOUNT,
            payMethod: "CARD,QR",
            processType: 'AT',
            merchantBuyerId: 'jparteta',
            installments: '00',
            dateDateTimeTransaction: TIEMPO.toString(),
        },
        card: {
            brand: '',
            pan: '',
        },
        billing: {
            firstName: 'Jean Pierre',
            lastName: 'Arteta',
            email: 'jpierre.arteta@gmail.com',
            phoneNumber: '943541279',
            street: 'calle de ópruebas',
            city: 'lima',
            state: 'lima',
            country: 'PE',
            postalCode: '00001',
            document: '70155398',
            documentType: 'DNI',
        },
        render: {
            typeForm: Izipay.enums.typeForm.POP_UP,
            container: '#your-iframe-payment',
            showButtonProcessForm: true,
            redirectUrls: {
                onSuccess: 'https://mierror.com/onSuccess',
                onError: 'https://mierror.com/onError',
                onCancel: 'https://micancel.com onCancel',
            }
        },
        urlRedirect:'https://www.google.com/',
        appearance: {
            logo:'https://demo-izipay.azureedge.net/test/img/millasb.svg',
        },
    },
};
```

#### 2.4.2 Crear una instancia a la clase Izipay

Se requiere enviar como argumento el objeto configuración previamente creado

```
try {
    const izi = new Izipay({config: iziConfig});
}
catch (error) { }
```

#### 2.4.3 Levantar el formulario

El método LoadForm, recibe los siguientes parámetros:

Parámetros del método LoadForm	Descripción	Obligatorio
authorization	Valor usado para autenticar al sistema de desee consumir un servicio de PuntoWeb.	SI
Llave publica RSA	Llave pública	SI
callbackResponse	Valor usado para recibir las respuestas de la transacción.	NO

- Si se desea obtener todas las respuestas de la transacción, esto se puede hacer pasando como argumento un callback.
- Otra manera de Leer la respuesta de la transacción es accediendo a la Clase global Izipay de la siguiente manera:

```
const callbackResponsePayment = response =>
console.log(response);

try {
    izi && izi.LoadForm({
        autorization: 'TU_TOKEN_SESSION',
        keyRSA: 'RSA',
        callbackResponse: callbackResponsePayment
    });
} catch (error) {
//console.log(error.message, error.Errors, error.date);
}
```

**Observación:** En el ejemplo anterior, se mostró un Formulario de pago, que debería ser invocado desde tu Front-end y puedes visualizar los errores en el apartado del 'catch'.

Aplicación web:

**Repositorio de ejemplo** <https://bitbucket.org/mc1768/izipay-examples-checkout/src/main/>

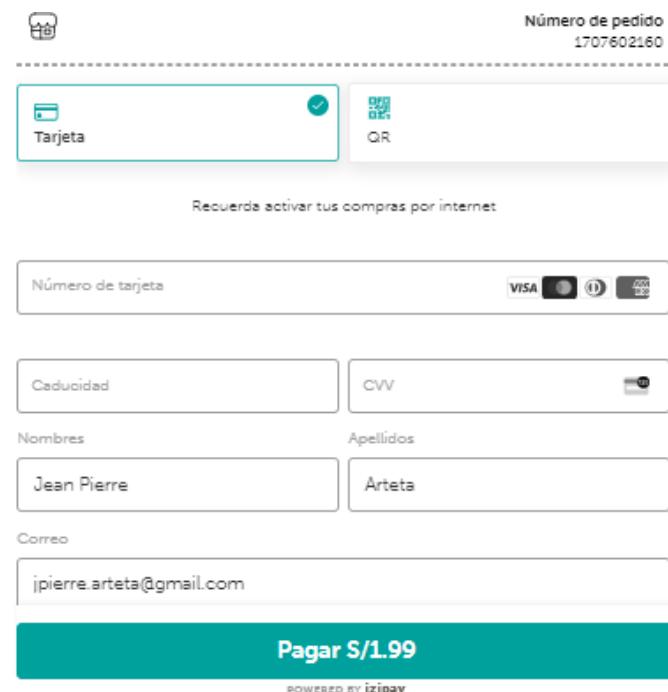
#### 2.4.4 Tipos de Formularios:

El SDK actual soporta los siguientes tipos de formulario: **Embebido, Pop-up y Redirect.**

##### Embebido

Se tendría que enviar de la siguiente forma:

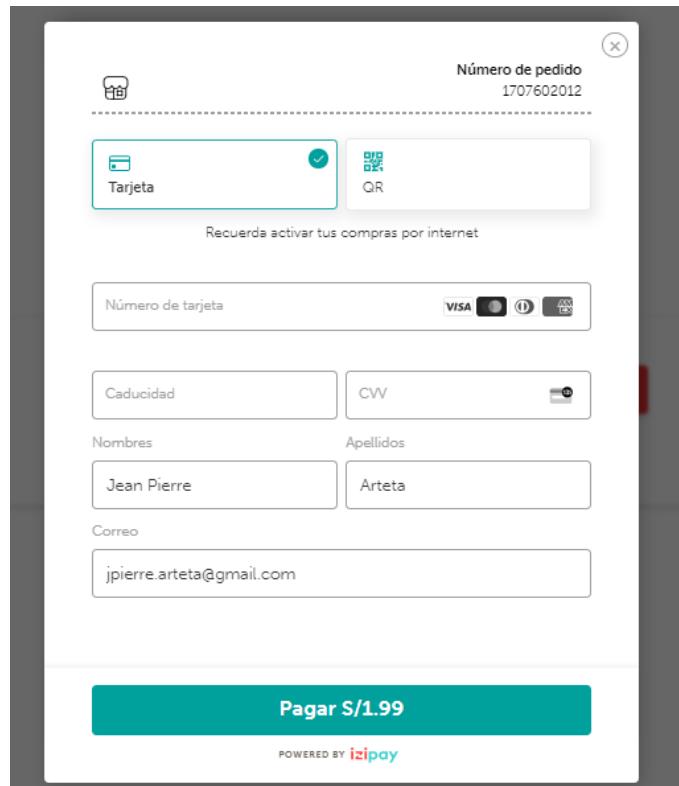
```
const iziConfig = {  
  config: {  
    ....  
    render: {  
      typeForm: Izipay.enums.typeForm.EMBEDDED,  
      container: '#your-iframe-payment',  
      showButtonProcessForm: true,  
    },  
    ...  
  }};
```



**Pop-up**

Se tendría que enviar de la siguiente forma:

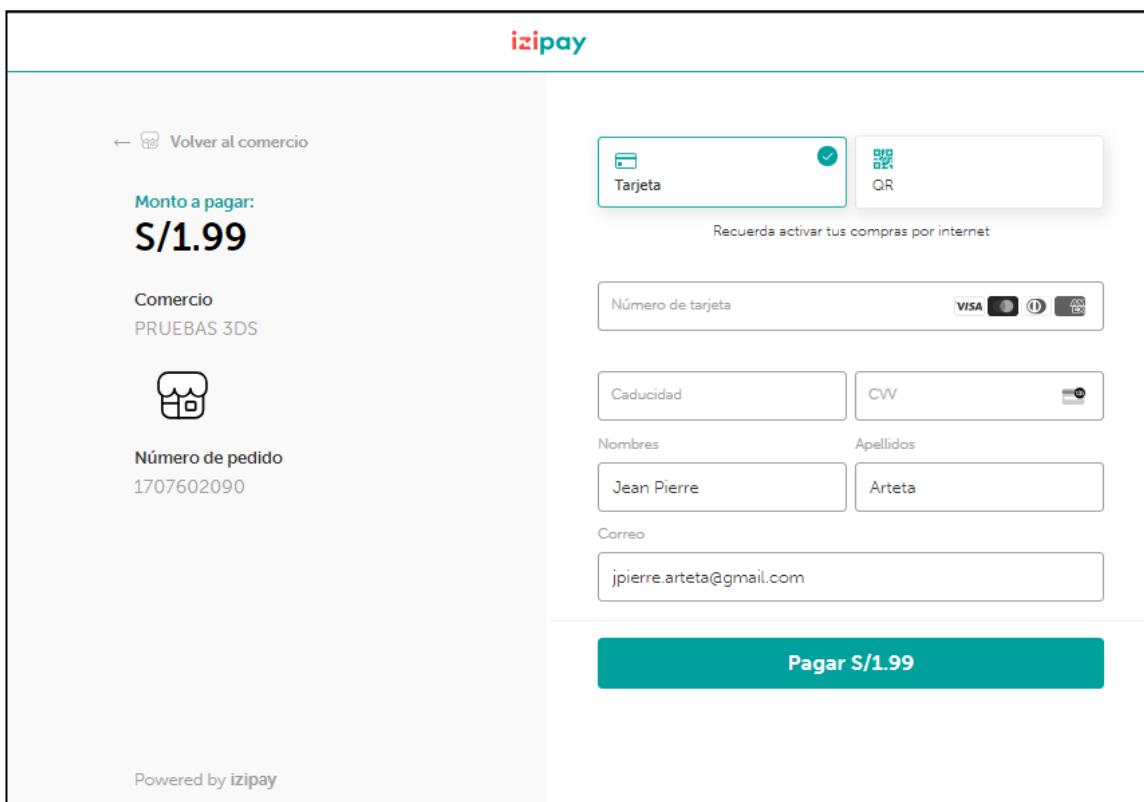
```
const iziConfig = {  
  config: {  
    ...  
    render: {  
      typeForm: Izipay.enums.typeForm.POP_UP,  
      container: '#your-iframe-payment',  
      showButtonProcessForm: true,  
      ...  
    },  
    ...  
  }};
```



**Redirect**

Se tendría que enviar de la siguiente forma:

```
const iziConfig = {  
  config: {  
    ...,  
    render: {  
      typeForm: Izipay.enums.typeForm.REDIRECT,  
      container: '#your-iframe-payment',  
      showButtonProcessForm: false,  
      redirectUrls: {  
        onSuccess: 'https://mierror.com/success',  
        onError: 'https://mierror.com/error',  
        onCancel: 'https://micancel.com/backcommerce',  
      }  
    },  
    ...  
  }};
```



### 3. TARJETAS DE PRUEBAS

<b>Tarjetas aceptadas</b>					
4970100000000055	12	2025	123	Visa	
4970100000000014	12	2025	123	Visa	
5100010000002045	12	2025	123	Mastercard	
5204740000001127	03	2026	261	Mastercard	
<b>Tarjetas rechazadas</b>					
4019138210019096	12	2025	123	Visa	
14 - Número de tarjeta inválido					
5103225852930567	12	2025	123	Mastercard	
14 - Número de tarjeta inválido					
4014818108389710	12	2025	123	Visa	
41 - Tarjeta perdida					
5102546179105000	12	2025	123	Mastercard	
41 - Tarjeta perdida					
4003331978380689	12	2025	123	Visa	
51 - Fondos insuficientes / Límite de crédito excedido					
5102495814688105	12	2025	123	Mastercard	
51 - Fondos insuficientes / Límite de crédito excedido					
4008069640361357	10	2022	123	Visa	
54 - Tarjeta expirada					
5106659582128224	12	2022	123	Mastercard	
54 - Tarjeta expirada					
4002913238833267	12	2025	123	Visa	
61 - Excede el límite de monto de retiro					
5114803122406407	12	2025	123	Mastercard	
61 - Excede el límite de monto de retiro					
4004412132480232	12	2025	123	Visa	
63 - Violación de Seguridad					
5112929646850374	12	2025	123	Mastercard	
63 - Violación de Seguridad					