

Configuración de Mercado Pago

Índice

1. [Requisitos Previos](#)
 2. [Paso 1: Crear Cuenta de Mercado Pago](#)
 3. [Paso 2: Obtener Credenciales de Test](#)
 4. [Paso 3: Configurar Webhooks](#)
 5. [Paso 4: Probar en Sandbox](#)
 6. [Paso 5: Credenciales de Producción](#)
 7. [Paso 6: Configurar en la Plataforma](#)
 8. [Flujo de Pago](#)
 9. [Webhooks](#)
 10. [Testing](#)
 11. [Troubleshooting](#)
-

Requisitos Previos

Antes de comenzar, asegúrate de tener:

- ☒ Una cuenta de Mercado Pago (gratuita)
 - ☒ Documentación legal de tu empresa (para cuentas de producción)
 - ☒ Acceso a la plataforma de incentivos
 - ☒ Capacidad para recibir webhooks (URL pública)
-

Paso 1: Crear Cuenta de Mercado Pago

Para Chile

1. Ve a [Mercado Pago Chile](https://www.mercadopago.cl/) (<https://www.mercadopago.cl/>)
2. Haz clic en **“Crear cuenta”**
3. Completa el registro:
 - Email
 - Contraseña
 - Datos personales
4. Verifica tu email
5. Completa el perfil de tu cuenta

Tipos de Cuenta

Tipo	Para quién	Características
Personal	Individuos	Límites de transacción menores
Empresarial	Negocios	Límites mayores, certificación requerida

Paso 2: Obtener Credenciales de Test

Las credenciales de test te permiten probar sin procesar pagos reales.

Acceder a Credenciales

1. Inicia sesión en [Mercado Pago](https://www.mercadopago.cl/) (<https://www.mercadopago.cl/>)
2. Ve a **“Tus integraciones”** > **“Credenciales”**
3. Selecciona **“Credenciales de prueba”** en el selector

Credenciales de Test

Encontrarás dos tipos de credenciales:

Public Key (Clave pública):

```
TEST-xxxxxxxx-xxxxxx-xx-xxxxxxxx-xxxxxxxx
```

- Se usa en el frontend
- No es sensible (puede ser pública)

Access Token (Token de acceso):

```
TEST-123456789012345-102030-xxxxxxxxxxxxxxxxxxxxxxxx-123456789
```

- Se usa en el backend/API calls
- **DEBE mantenerse privada**

Copia las Credenciales

```
# Credenciales de TEST para desarrollo
PUBLIC_KEY: TEST-xxxxxxxx-xxxxxx-xx-xxxxxxxx-xxxxxxxx
ACCESS_TOKEN: TEST-123456789012345-102030-xxxxxxxxxxxxxxxxxxxxxxxx-123456789
```

Paso 3: Configurar Webhooks

Los webhooks notifican a tu plataforma cuando ocurren eventos (pagos aprobados, rechazados, etc.).

¿Qué es un Webhook?

Un webhook es una URL en tu servidor que Mercado Pago llama cuando ocurre un evento:

Pago aprobado → Mercado Pago llama tu URL → Tu sistema actualiza la deuda

Configurar URL de Webhook

1. En **“Tus integraciones”** > **“Webhooks”**
2. Haz clic en **“Configurar webhooks”**
3. Agrega tu URL:

`https://tudominio.com/api/webhooks/mercadopago`

4. Selecciona los eventos:
 - ☒ `payment` (Pagos)
 - ☒ `merchant_order` (Órdenes, opcional)

Para Testing Local

Si estás desarrollando localmente, usa [ngrok](https://ngrok.com/) (<https://ngrok.com/>):

```
# 1. Instala ngrok
npm install -g ngrok

# 2. Ejecuta tu servidor local (ej: puerto 5173)
npm run dev

# 3. Crea un túnel público
ngrok http 5173

# 4. Usa la URL generada como webhook
https://abc123.ngrok.io/api/webhooks/mercadopago
```

Paso 4: Probar en Sandbox

Usuarios de Prueba

Mercado Pago proporciona usuarios de prueba para simular pagos:

1. Ve a **“Tus integraciones”** > **“Usuarios de prueba”**
2. Haz clic en **“Crear usuario de prueba”**
3. Configura:
 - **País:** Chile
 - **Dinero disponible:** 10000 (o el monto que necesites)

Tarjetas de Prueba

Usa estas tarjetas para simular diferentes escenarios:

Pago Aprobado:

Número: 5031 7557 3453 0604
 CVV: 123
 Vencimiento: 11/25
 Nombre: APRO

Pago Rechazado:

Número: 5031 4332 1540 6351
 CVV: 123
 Vencimiento: 11/25
 Nombre: OTHE

Pago Pendiente:

Número: 5031 7557 3453 0604
 CVV: 123
 Vencimiento: 11/25
 Nombre: CONT

Flujo de Prueba

1. Crea una preferencia de pago desde tu app
2. Redirige al checkout de Mercado Pago
3. Usa una tarjeta de prueba
4. Completa el pago
5. Verifica que tu webhook reciba la notificación
6. Confirma que la deuda se actualice en tu BD

Paso 5: Credenciales de Producción

Requisitos para Producción

Para usar credenciales de producción, debes:

1. ☒ Completar la información de tu empresa
2. ☒ Verificar tu identidad
3. ☒ Agregar información bancaria (para recibir fondos)
4. ☒ Aceptar términos y condiciones

Certificación de Cuenta

1. Ve a **“Tus integraciones” > “Credenciales”**
2. Completa el formulario de certificación:
 - Razón social
 - RUT de la empresa
 - Documentación legal
 - Cuenta bancaria

Obtener Credenciales de Producción

1. Una vez certificado, ve a **“Credenciales de producción”**

2. Copia las credenciales:

```
PUBLIC_KEY: APP-xxxxxxx-xxxxxx-xx-xxxxxxx-xxxxxxx
```

```
ACCESS_TOKEN: APP-123456789012345-102030-xxxxxxxxxxxxxxxxxxxxxx-123456789
```

Paso 6: Configurar en la Plataforma

Variables de Entorno

Edita el archivo `.env` :

```
# Para desarrollo (usa credenciales TEST)
VITE_MERCADOPAGO_ACCESS_TOKEN=TEST-123456789012345-102030-
xxxxxxxxxxxxxxxxxxxxxx-123456789
VITE_MERCADOPAGO_PUBLIC_KEY=TEST-xxxxxxx-xxxxxx-xx-xxxxxxx-xxxxxxx
VITE_MERCADOPAGO_SANDBOX=true

# Para producción (usa credenciales APP)
VITE_MERCADOPAGO_ACCESS_TOKEN=APP-123456789012345-102030-
xxxxxxxxxxxxxxxxxxxxxx-123456789
VITE_MERCADOPAGO_PUBLIC_KEY=APP-xxxxxxx-xxxxxx-xx-xxxxxxx-xxxxxxx
VITE_MERCADOPAGO_SANDBOX=false
```

Verificar Configuración

```
import mercadoPagoService from './services/integrations/mercadopago.service';

const status = mercadoPagoService.isConfigured();
console.log(status);
// { configured: true, message: "Mercado Pago configurado (SANDBOX)" }
```

Flujo de Pago

1. Crear Preferencia de Pago

```
import { useMercadoPago } from './hooks/integrations';

function PaymentButton({ debt }) {
  const { createDebtPayment } = useMercadoPago();

  const handlePay = async () => {
    const result = await createDebtPayment(debt);

    if (result.success) {
      // Redirigir al checkout de Mercado Pago
      window.location.href = result.initPoint;
    }
  };

  return <button onClick={handlePay}>Pagar con Mercado Pago</button>;
}
```

2. Usuario Paga en Mercado Pago

El usuario es redirigido al checkout de Mercado Pago donde:

1. Elige método de pago (tarjeta, efectivo, etc.)
2. Completa los datos
3. Confirma el pago

3. Mercado Pago Procesa el Pago

- Si es aprobado → Estado: `approved`
- Si es rechazado → Estado: `rejected`
- Si está pendiente → Estado: `pending`

4. Webhook Notifica a la Plataforma

Mercado Pago llama tu webhook con información del pago:

```
{
  "action": "payment.created",
  "api_version": "v1",
  "data": {
    "id": "123456789"
  },
  "date_created": "2025-10-15T10:00:00Z",
  "id": 123456789,
  "live_mode": false,
  "type": "payment",
  "user_id": "987654321"
}
```

5. Tu Sistema Procesa el Webhook

```
// Este código ya está implementado en mercadopago.service.js
async processWebhook(webhookData) {
  // 1. Obtener información del pago
  const payment = await this.getPayment(webhookData.data.id);

  // 2. Guardar transacción en BD
  await this.saveTransaction(payment);

  // 3. Si está aprobado, actualizar deuda
  if (payment.status === 'approved') {
    await this.processApprovedPayment(payment);
  }
}
```

6. Usuario es Redirigido

Según el resultado, el usuario es redirigido a:

- **Éxito:** `/payment/success`
 - **Pendiente:** `/payment/pending`
 - **Fallo:** `/payment/failure`
-

Webhooks

Endpoint de Webhook

Crea un endpoint en tu backend para recibir notificaciones:

```
// Express.js example
app.post('/api/webhooks/mercadopago', async (req, res) => {
  try {
    const webhookData = req.body;

    // Procesar webhook
    await mercadoPagoService.processWebhook(webhookData);

    // Responder 200 OK
    res.status(200).send('OK');
  } catch (error) {
    console.error('Error procesando webhook:', error);
    res.status(500).send('Error');
  }
});
```

Validar Webhooks

Para mayor seguridad, valida que el webhook venga de Mercado Pago:

```
const crypto = require('crypto');

function validateWebhook(req) {
  const xSignature = req.headers['x-signature'];
  const xRequestId = req.headers['x-request-id'];

  // Construir string a firmar
  const dataID = req.body.data.id;
  const stringToSign = `id:${dataID};request-id:${xRequestId}`;

  // Calcular HMAC
  const hmac = crypto
    .createHmac('sha256', process.env.MERCADOPAGO_WEBHOOK_SECRET)
    .update(stringToSign)
    .digest('hex');

  return hmac === xSignature;
}
```

Reintentos

Mercado Pago reintentará enviar el webhook si no recibe 200 OK:

- Reintento 1: Inmediato
- Reintento 2: 15 minutos después
- Reintento 3: 1 hora después
- Hasta 12 reintentos

Testing

Test de Integración Completa

```
describe('Mercado Pago Integration', () => {
  test('Crear preferencia de pago', async () => {
    const { createPaymentPreference } = useMercadoPago();

    const result = await createPaymentPreference({
      debtId: 'test-123',
      debtorId: 'user-456',
      debtorEmail: 'test@example.com',
      debtorName: 'Test User',
      amount: 50000,
      description: 'Pago de deuda test'
    });

    expect(result.success).toBe(true);
    expect(result.initPoint).toContain('mercadopago');
  });

  test('Procesar webhook de pago aprobado', async () => {
    const webhookData = {
      type: 'payment',
      data: { id: '123456789' }
    };

    const result = await mercadoPagoService.processWebhook(webhookData);
    expect(result.success).toBe(true);
  });
});
```

Test Manual con Tarjetas

1. Crea un pago de prueba
2. Usa tarjeta `APRO` para aprobar
3. Verifica que:
 - ☒ Deuda se marca como pagada
 - ☒ Incentivo se otorga al deudor
 - ☒ Se registra en historial
 - ☒ Se envía notificación

Troubleshooting

Error: “invalid_token”

Problema: Access Token inválido

Solución:

```
# Verifica que el token esté correcto
# Para TEST debe empezar con TEST-
# Para PRODUCCIÓN debe empezar con APP-
```


Error: “payment_method_not_found”

Problema: Método de pago no disponible

Solución:

- Verifica que el método esté habilitado en tu cuenta
- Algunos métodos requieren certificación

Webhook no se recibe

Problema: Tu servidor no recibe notificaciones

Checklist:

1. ☒ URL de webhook configurada correctamente
2. ☒ URL es accesible públicamente (no localhost)
3. ☒ Endpoint responde 200 OK
4. ☒ Firewall permite requests de Mercado Pago

IPs de Mercado Pago (whitelist si es necesario):

- 209.225.49.0/24
- 216.33.196.0/24
- 216.33.197.0/24

Pagos en estado “pending”

Problema: Pago queda pendiente

Causas comunes:

- Usuario seleccionó pago en efectivo (se procesa cuando paga)
- Tarjeta con 3DS (requiere autenticación bancaria)
- Revisión manual de Mercado Pago

Acción:

- Notifica al usuario del estado pendiente
- Monitorea cambios de estado vía webhooks

Error: “amount_exceeds_limits”

Problema: Monto excede límites de transacción

Solución:

- Cuenta personal: Máximo ~\$500,000 CLP por transacción
- Certifica tu cuenta para límites mayores
- Divide en cuotas si es necesario

Mejores Prácticas

1. Idempotencia

Usa claves de idempotencia para evitar pagos duplicados:

```
const idempotencyKey = `${debtId}-${timestamp}`;  
// Ya implementado en el servicio
```

2. Metadata

Incluye metadata para rastrear pagos:

```
const preference = {
  // ...
  metadata: {
    debt_id: debtId,
    debtor_id: debtorId,
    platform_version: '2.0',
    payment_type: 'debt_payment'
  }
};
```

3. Statement Descriptor

Personaliza cómo aparece el cargo en el extracto:

```
const preference = {
  // ...
  statement_descriptor: 'PLATAFORMA INCENTIVOS' // Max 11 caracteres
};
```

4. Expires

Configura tiempo de expiración para preferencias:

```
const preference = {
  // ...
  expires: true,
  expiration_date_from: new Date().toISOString(),
  expiration_date_to: new Date(Date.now() + 48 * 60 * 60 * 1000).toISOString() // 48
  horas
};
```

5. Monitoreo

Implementa monitoreo de pagos:

```
// Log todos los eventos importantes
console.log('💰 Pago creado:', preferenceId);
console.log('✅ Pago aprobado:', paymentId);
console.log('❌ Pago rechazado:', paymentId, reason);
```

Pasar de Test a Producción

Checklist

Antes de pasar a producción:

- ☐ Cuenta certificada
- ☐ Credenciales de producción obtenidas
- ☐ Webhooks configurados en URL de producción

- [] Testing completo realizado en sandbox
- [] Variables de entorno actualizadas
- [] `VITE_MERCADOPAGO_SANDBOX=false`
- [] Monitoreo y alertas configurados
- [] Documentación de procesos lista

Cambio Gradual

Considera un rollout gradual:

1. **Semana 1:** 10% de usuarios en producción
2. **Semana 2:** 50% de usuarios
3. **Semana 3:** 100% de usuarios

```
const useProduction = Math.random() < 0.1; // 10%
const token = useProduction
  ? PRODUCTION_TOKEN
  : TEST_TOKEN;
```

Recursos Adicionales

- [Documentación Oficial Mercado Pago](https://www.mercadopago.cl/developers/es/docs) (<https://www.mercadopago.cl/developers/es/docs>)
- [Checkout Pro](https://www.mercadopago.cl/developers/es/docs/checkout-pro/landing) (<https://www.mercadopago.cl/developers/es/docs/checkout-pro/landing>)
- [Webhooks](https://www.mercadopago.cl/developers/es/docs/your-integrations/notifications/web-hooks) (<https://www.mercadopago.cl/developers/es/docs/your-integrations/notifications/web-hooks>)
- [Tarjetas de Prueba](https://www.mercadopago.cl/developers/es/docs/checkout-api/integration-test/test-cards) (<https://www.mercadopago.cl/developers/es/docs/checkout-api/integration-test/test-cards>)

Soporte

¿Necesitas ayuda?

- Revisa los logs de tu servidor
- Consulta la [documentación oficial](https://www.mercadopago.cl/developers) (<https://www.mercadopago.cl/developers>)
- Contacta al soporte de Mercado Pago
- Revisa el código de ejemplo en este repositorio

Versión: 1.0.0

Última actualización: Octubre 2025