

# Guía Rápida: Integración Bankinter Open Banking



## Resumen

Esta integración permite a INMOVA conectarse con Bankinter a través de Open Banking PSD2 (Redsys) para:

- Verificación automática de pagos de alquiler
- Sincronización de transacciones bancarias
- Iniciación de pagos SEPA
- Conciliación automática de pagos
- Verificación de ingresos de inquilinos



## Configuración Rápida

### 1. Variables de Entorno

Añadir al archivo `.env`:

```
# Bankinter / Redsys PSD2 Configuration
REDSYS_API_URL=https://apis-i.redsys.es:20443/psd2/xs2a/api-entrada-xs2a/services
REDSYS_OAUTH_URL=https://apis-i.redsys.es:20443/psd2/xs2a/api-oauth-xs2a
REDSYS_BANKINTER_CODE=bankinter

# Credenciales OAuth (obtener de Redsys)
REDSYS_CLIENT_ID=your_client_id
REDSYS_CLIENT_SECRET=your_client_secret

# Certificados eIDAS (rutas absolutas)
REDSYS_CERTIFICATE_PATH=/path/to/qwac_cert.pem
REDSYS_CERTIFICATE_KEY_PATH=/path/to/qwac_key.pem
REDSYS_SEAL_CERTIFICATE_PATH=/path/to/qseal_cert.pem
REDSYS_SEAL_KEY_PATH=/path/to/qseal_key.pem

# URL de la aplicación (para callbacks)
NEXT_PUBLIC_APP_URL=https://tu-dominio.com
```

### 2. Certificados eIDAS

Para **producción**, necesitas:

1. **Certificado QWAC** (Qualified Website Authentication Certificate)
2. **Certificado QSealC** (Qualified Seal Certificate)

Para **sandbox/pruebas**, puedes usar certificados de prueba proporcionados por Redsys.

### 3. Verificar Configuración

El servicio se inicializa automáticamente y verifica la configuración. Si falta alguna variable o certificado, operará en **MODO DEMO**.

## Uso de los Endpoints

### 1. Conectar Cuenta Bancaria de Bankinter

```
// Frontend - Iniciar conexión
const response = await fetch('/api/open-banking/bankinter/connect', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' }
});

const data = await response.json();
// { consentId: 'xxx', authUrl: 'https://...', ... }

// Redirigir usuario a la URL de autenticación de Bankinter
window.location.href = data.authUrl;
```

#### Flujo:

1. Usuario hace clic en “Conectar Bankinter”
2. Sistema crea consentimiento AIS en Redsys
3. Usuario es redirigido a Bankinter para autenticar con Bankinter Móvil
4. Después de autenticar, Bankinter redirige a /api/open-banking/bankinter/callback
5. Sistema valida el consentimiento y marca la conexión como activa

### 2. Sincronizar Transacciones

```
// Sincronizar transacciones de los últimos 90 días
const response = await fetch('/api/open-banking/bankinter/sync', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({
    connectionId: 'conn_xxx',
    diasAtras: 90
  })
});

const data = await response.json();
// { success: true, total: 150, transacciones: [...], ... }
```

#### Qué hace:

- Obtiene todas las cuentas del consentimiento
- Descarga transacciones de cada cuenta
- Las guarda en `BankTransaction` automáticamente
- Evita duplicados

### 3. Verificar Ingresos de Inquilino

```
// El inquilino debe tener una conexión Bankinter activa primero
const response = await fetch('/api/open-banking/verify-income', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({
    tenantId: 'tenant_xxx'
  })
});

const data = await response.json();
/*
{
  success: true,
  informe: {
    tenantId: 'tenant_xxx',
    ingresosPromedio: 2500,
    ingresosMinimos: 2000,
    ingresosMaximos: 3000,
    estabilidad: 85,
    fuentesIngresos: ['EMPRESA ABC', 'FREELANCE'],
    recomendacion: 'Buena capacidad de pago...',
    proveedor: 'bankinter'
  }
}
*/
```

**Qué hace:**

- Analiza transacciones de ingresos de los últimos 3 meses
- Calcula promedio, mínimo, máximo
- Evalúa estabilidad de ingresos (coeficiente de variación)
- Identifica fuentes principales de ingresos
- Genera recomendación automática

### 4. Conciliar Pagos Automáticamente

```
// Conciliar pagos del último mes
const response = await fetch('/api/open-banking/bankinter/reconcile', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({
    mesesAtras: 1
  })
});

const data = await response.json();
// { success: true, conciliados: 45, total: 50, ... }
```

**Qué hace:**

- Busca pagos pendientes en el sistema
- Busca transacciones bancarias que coincidan con:
  - Monto similar ( $\pm 1\%$  tolerancia)
  - Fecha cercana a la de vencimiento ( $\pm 5$  días antes,  $+10$  después)
  - Nombre del inquilino o concepto

- Marca automáticamente los pagos como “pagado”
- Actualiza las transacciones como “conciliado”

## 5. Iniciar un Pago SEPA

```
// Iniciar pago a proveedor
const response = await fetch('/api/open-banking/bankinter/payment', {
  method: 'POST',
  headers: { 'Content-Type': 'application/json' },
  body: JSON.stringify({
    debtorIban: 'ES1234567890123456789012', // Cuenta origen
    creditorIban: 'ES9876543210987654321098', // Cuenta destino
    creditorName: 'Nombre del Proveedor',
    amount: 1500.00,
    currency: 'EUR',
    concept: 'Pago de factura #123'
  })
});

const data = await response.json();
// { success: true, paymentId: 'pay_xxx', authUrl: 'https://...', ... }

// Redirigir usuario a autenticar el pago
window.location.href = data.authUrl;
```

### Flujo:

1. Sistema crea orden de pago en Redsys
2. Usuario es redirigido a Bankinter para autorizar con Bankinter Móvil
3. Una vez autorizado, Bankinter ejecuta el pago
4. Sistema puede consultar el estado del pago

## 6. Consultar Estado de Consentimiento

```
const response = await fetch(
  '/api/open-banking/bankinter/status?consentId=consent_xxx'
);

const data = await response.json();
/*
{
  success: true,
  consentId: 'consent_xxx',
  status: 'valid',
  validUntil: '2025-03-01T00:00:00Z',
  connection: {
    id: 'conn_xxx',
    nombreBanco: 'Bankinter',
    estado: 'conectado',
    ultimaSync: '2025-01-15T10:30:00Z'
  }
}
*/
```

## Automatizaciones

### Sincronización Automática Diaria

Puedes crear un cron job o tarea programada para sincronizar transacciones diariamente:

```
// scripts/sync-bankinter-daily.ts
import { prisma } from '@/lib/db';
import { bankinterService } from '@/lib/bankinter-integration-service';

async function syncAllConnections() {
  const connections = await prisma.bankConnection.findMany({
    where: {
      proveedor: 'bankinter_redsys',
      estado: 'conectado'
    }
  });

  for (const connection of connections) {
    try {
      await bankinterService.sincronizarTransaccionesBankinter(
        connection.id,
        1 // Último día
      );
      console.log(`✅ Sincronizado: ${connection.id}`);
    } catch (error) {
      console.error(`❌ Error: ${connection.id}`, error);
    }
  }
}

syncAllConnections();
```

## Conciliación Automática Nocturna

```
// scripts/reconcile-payments-nightly.ts
import { prisma } from '@/lib/db';
import { bankinterService } from '@/lib/bankinter-integration-service';

async function reconcileAllCompanies() {
  const companies = await prisma.company.findMany({
    where: {
      bankConnections: {
        some: {
          proveedor: 'bankinter_redsys',
          estado: 'conectado'
        }
      }
    }
  });
}

for (const company of companies) {
  try {
    const result = await bankinterService.conciliarPagosBankinter(
      company.id,
      1 // Último mes
    );
    console.log(`✓ ${company.nombre}: ${result.conciliados}/${result.total}`);
  } catch (error) {
    console.error(`✗ ${company.nombre}:`, error);
  }
}
}

reconcileAllCompanies();
```



## Seguridad

### Renovación de Consentimientos

Los consentimientos AIS expiran después de **90 días**. Implementa un sistema de notificaciones:

```
// Verificar consentimientos próximos a expirar
const proximosAEpirar = await prisma.bankConnection.findMany({
  where: {
    proveedor: 'bankinter_redsys',
    estado: 'conectado',
    consentValidUntil: {
      lte: new Date(Date.now() + 7 * 24 * 60 * 60 * 1000) // 7 días
    }
  }
});

// Enviar notificaciones a usuarios para renovar
```

## Protección de Certificados

**⚠ IMPORTANTE:** Los certificados eIDAS contienen información crítica. Asegúrate de:

1. Almacenarlos fuera del repositorio Git

2. Usar permisos restrictivos (600 o 400)
  3. Considerar usar Azure Key Vault, AWS Secrets Manager, o similar
  4. Renovarlos antes de su expiración
- 

## Troubleshooting

### Error: “Integración con Bankinter no configurada”

**Causa:** Faltan variables de entorno o certificados.

**Solución:**

1. Verificar que todas las variables `REDSYS_*` estén en `.env`
2. Verificar que los archivos de certificados existan
3. Revisar logs del servidor para detalles

### Error: “Consentimiento no válido”

**Causa:** El consentimiento ha expirado o fue revocado.

**Solución:**

1. Solicitar al usuario que vuelva a conectar su cuenta
2. Crear un nuevo consentimiento

## Transacciones no se sincronizan

**Posibles causas:**

1. Consentimiento expirado
2. Cuenta bancaria bloqueada o cerrada
3. Límite de frecuencia de API alcanzado

**Solución:**

1. Verificar estado del consentimiento con `/bankinter/status`
  2. Revisar logs de errores
  3. Respetar límites de frecuencia (4 peticiones/día por defecto)
- 

## Monitoreo

### Métricas Clave

1. **Conexiones activas:** Número de conexiones Bankinter activas
2. **Transacciones sincronizadas/día:** Volumen de transacciones obtenidas
3. **Tasa de conciliación:** % de pagos conciliados automáticamente
4. **Consentimientos próximos a expirar:** Alertas tempranas
5. **Errores de API:** Tasa de errores en peticiones a Redsys

## Queries Útiles

```
-- Conexiones activas de Bankinter
SELECT COUNT(*) FROM bank_connections
WHERE proveedor = 'bankinter_redsys' AND estado = 'conectado';

-- Transacciones sincronizadas hoy
SELECT COUNT(*) FROM bank_transactions
WHERE created_at >= CURRENT_DATE;

-- Pagos conciliados automáticamente este mes
SELECT COUNT(*) FROM payments
WHERE estado = 'pagado'
AND metodo_pago = 'transferencia_bancaria'
AND fecha_pago >= DATE_TRUNC('month', CURRENT_DATE);
```



## Recursos Adicionales

- **Documentación completa:** /INTEGRACION\_BANKINTER\_GUIA.md
- **Portal Redsys PSD2:** <https://market.apis-i.redsys.es/psd2/xs2a/nodos/bankinter>
- **Documentación PSD2:** <https://www.redsys.es/es/psd2>
- **Info Bankinter Open Banking:** <https://www.bankinter.com/blog/noticias-bankinter/bankinter-abre-apis-psd2>



## Checklist de Implementación

- [ ] Variables de entorno configuradas
- [ ] Certificados eIDAS obtenidos e instalados
- [ ] Registro en Redsys PSD2 Platform completado
- [ ] Flujo de conexión probado en sandbox
- [ ] Sincronización de transacciones funcionando
- [ ] Conciliación automática probada
- [ ] Sistema de notificaciones para renovación de consentimientos
- [ ] Monitoreo y alertas configurados
- [ ] Documentación de usuario creada
- [ ] Capacitación del equipo de soporte

**Última actualización:** Diciembre 2025

**Versión de la integración:** 1.0