

Configuración de Integración CRM

Índice

- 1. [Visión General](#)
 - 2. [Salesforce CRM](#)
 - 3. [HubSpot CRM](#)
 - 4. [Zoho CRM](#)
 - 5. [Mapeo de Campos](#)
 - 6. [Sincronización](#)
 - 7. [Troubleshooting](#)
-







Visión General

La plataforma soporta integración con tres sistemas CRM principales:

CRM	Tipo de Autenticación	Complejidad	Documentación
Salesforce	OAuth 2.0	Media	Docs (https://developer.salesforce.com/)
HubSpot	Private App Token	Baja	Docs (https://developers.hubspot.com/)
Zoho	OAuth 2.0	Media	Docs (https://www.zoho.com/crm/developer/)

Funcionalidades Comunes

Todas las integraciones CRM proporcionan:

-  Sincronización de contactos/deudores
 -  Importación de deudas
 -  Actualización de estados
 -  Registro de actividades
 -  Creación de acuerdos de pago
 -  Sincronización completa e incremental
-

Salesforce CRM

Requisitos Previos

- Cuenta de Salesforce (Developer, Enterprise, o Professional)
- Permisos de administrador para crear Connected Apps

Paso 1: Crear Connected App

1. Inicia sesión en Salesforce
2. Ve a **Setup** (Configuración)
3. En Quick Find, busca **“App Manager”**
4. Haz clic en **“New Connected App”**

Paso 2: Configurar OAuth

1. Completa la información básica:
 - **Connected App Name:** Plataforma de Incentivos
 - **API Name:** Se genera automáticamente
 - **Contact Email:** Tu email
2. Habilita **“Enable OAuth Settings”**:
 - **Callback URL:** `https://tudominio.com/auth/salesforce/callback`
 - **Selected OAuth Scopes:**
 - `Full access (full)`
 - `Perform requests on your behalf at any time (refresh_token, offline_access)`
 - `Access and manage your data (api)`
3. Haz clic en **“Save”**

Paso 3: Obtener Credenciales

1. Una vez guardada, espera 2-10 minutos para que se propague
2. Haz clic en **“Manage Consumer Details”**
3. Copia el **Consumer Key** y **Consumer Secret**

Paso 4: Generar Access Token

Existen varias formas de obtener el Access Token:

Opción A: Usando Postman (Recomendado)

1. Descarga [Postman](https://www.postman.com/) (<https://www.postman.com/>)
2. Crea una nueva request POST:
`https://login.salesforce.com/services/oauth2/token`
3. En Body (x-www-form-urlencoded), agrega:
`grant_type=password`
`client_id=tu_consumer_key`
`client_secret=tu_consumer_secret`
`username=tu_salesforce_username`
`password=tu_salesforce_password_y_security_token`
4. El security token se obtiene en Salesforce > Settings > Reset Security Token
5. La respuesta incluirá el `access_token` e `instance_url`

Opción B: Usando cURL

```
curl -X POST https://login.salesforce.com/services/oauth2/token \
-d "grant_type=password" \
-d "client_id=TU_CONSUMER_KEY" \
-d "client_secret=TU_CONSUMER_SECRET" \
-d "username=TU_EMAIL" \
-d "password=TU_PASSWORD_Y_SECURITY_TOKEN"
```

Paso 5: Configurar Campos Personalizados (Importante)

La integración usa campos personalizados en Salesforce. Créalos:

1. Ve a **Object Manager > Contact**
2. Crea estos campos personalizados:
 - RUT__c (Text, 12 caracteres)
 - Total_Debt__c (Currency)
 - Platform_User_ID__c (Text, 50 caracteres)
3. Ve a **Object Manager > Crea objeto personalizado "Debt__c"**:
 - Amount__c (Currency)
 - Status__c (Picklist: Pending, Paid, Overdue)
 - Due_Date__c (Date)
 - Original_Creditor__c (Text, 100)
 - Contact__c (Lookup to Contact)

Paso 6: Configurar en la Plataforma

```
# .env
VITE_SALESFORCE_ACCESS_TOKEN=tu_access_token_aqui
VITE_SALESFORCE_INSTANCE_URL=https://yourinstance.salesforce.com
```

Paso 7: Probar Integración

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

function TestSalesforce() {
  const { getDebtors, activeCRM } = useCRM();

  const test = async () => {
    console.log('CRM activo:', activeCRM); // "salesforce"
    const debtors = await getDebtors({ limit: 10 });
    console.log('Deudores:', debtors);
  };

  return <button onClick={test}>Probar Salesforce</button>;
}
```

HubSpot CRM

Requisitos Previos

- Cuenta de HubSpot (Free, Starter, Professional, o Enterprise)
- Permisos de administrador

Paso 1: Crear Private App

1. Inicia sesión en HubSpot
2. Ve a **Settings** (⚙️) > **Integrations** > **Private Apps**
3. Haz clic en **“Create a private app”**

Paso 2: Configurar Scopes

1. En la pestaña **“Scopes”**, selecciona:

Contacts:

- `crm.objects.contacts.read`
- `crm.objects.contacts.write`

Deals:

- `crm.objects.deals.read`
- `crm.objects.deals.write`

Notes/Activities:

- `crm.objects.notes.read`
- `crm.objects.notes.write`

Search:

- `crm.objects.contacts.search`
- `crm.objects.deals.search`

Paso 3: Obtener Access Token

1. Completa la información básica:
 - **Name:** Plataforma de Incentivos
 - **Description:** Integración para gestión de deudas
2. Haz clic en **“Create app”**
3. Copia el **Access Token** generado
4. ⚠️ **Importante:** Guárdalo de forma segura, no podrás verlo de nuevo

Paso 4: Configurar Propiedades Personalizadas

En HubSpot, las propiedades personalizadas se crean así:

1. Ve a **Settings** > **Properties**
2. Selecciona **“Contact properties”**
3. Haz clic en **“Create property”**

Crea estas propiedades:

- `rut` (Single-line text)
- `total_debt` (Number)
- `platform_user_id` (Single-line text)

1. Para **Deals**, crea:

- `installments` (Number)

- incentive_amount (Number)
- debt_type (Dropdown)

Paso 5: Configurar en la Plataforma

```
# .env
VITE_HUBSPOT_ACCESS_TOKEN=tu_access_token_aqui
```

Paso 6: Probar Integración

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

function TestHubSpot() {
  const { syncDebtor, changeCRM } = useCRM();

  const test = async () => {
    changeCRM('hubspot');

    const result = await syncDebtor({
      firstName: 'Juan',
      lastName: 'Pérez',
      email: 'juan@example.com',
      phone: '56912345678',
      rut: '12345678-9',
      totalDebt: 500000
    });

    console.log('Resultado:', result);
  };

  return <button onClick={test}>Probar HubSpot</button>;
}
```

Zoho CRM

Requisitos Previos

- Cuenta de Zoho CRM
- Permisos de administrador

Paso 1: Acceder a API Console

1. Ve a [Zoho API Console](https://api-console.zoho.com/) (https://api-console.zoho.com/)
2. Haz clic en “Add Client”
3. Selecciona “Server-based Applications”

Paso 2: Configurar Client

1. Completa la información:
 - **Client Name:** Plataforma de Incentivos
 - **Homepage URL:** https://tudominio.com
 - **Authorized Redirect URIs:** https://tudominio.com/auth/zoho/callback
2. Haz clic en “Create”

3. Copia **Client ID** y **Client Secret**

Paso 3: Generar Access Token

Obtener Authorization Code

1. Construye esta URL (reemplaza CLIENT_ID):

```
https://accounts.zoho.com/oauth/v2/auth?
scope=ZohoCRM.modules.ALL,ZohoCRM.settings.ALL&
client_id=TU_CLIENT_ID&
response_type=code&
access_type=offline&
redirect_uri=https://tudominio.com/auth/zoho/callback
```

1. Pégalas en el navegador
2. Autoriza la aplicación
3. Serás redirigido y verás el `code` en la URL
4. Copia ese código (válido por 60 segundos)

Intercambiar por Access Token

```
curl -X POST https://accounts.zoho.com/oauth/v2/token \
-d "code=TU_AUTHORIZATION_CODE" \
-d "client_id=TU_CLIENT_ID" \
-d "client_secret=TU_CLIENT_SECRET" \
-d "redirect_uri=https://tudominio.com/auth/zoho/callback" \
-d "grant_type=authorization_code"
```

La respuesta incluirá `access_token` y `refresh_token`.

Paso 4: Configurar Campos Personalizados

1. Ve a **Setup > Customization > Modules and Fields**
2. Selecciona **“Contacts”**
3. Agrega campos personalizados:
 - `RUT` (Single Line)
 - `Total_Debt` (Currency)
 - `Platform_User_ID` (Single Line)
4. Crea módulo personalizado **“Debts”**:
 - `Amount` (Currency)
 - `Status` (Pick List)
 - `Due_Date` (Date)
 - `Original_Creditor` (Single Line)
 - `Contact_Name` (Lookup - Contacts)

Paso 5: Configurar en la Plataforma

```
# .env
VITE_ZOHO_ACCESS_TOKEN=tu_access_token_aqui
VITE_ZOHO_API_DOMAIN=https://www.zohoapis.com
# Para otros data centers:
# EU: https://www.zohoapis.eu
# IN: https://www.zohoapis.in
# AU: https://www.zohoapis.com.au
```

Paso 6: Probar Integración

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

function TestZoho() {
  const { importDebts, changeCRM } = useCRM();

  const test = async () => {
    changeCRM('zoho');

    const result = await importDebts({ limit: 20 });
    console.log('Deudas importadas:', result);
  };

  return <button onClick={test}>Probar Zoho</button>;
}
```

Mapeo de Campos

Contactos/Deudores

Plataforma	Salesforce	HubSpot	Zoho
firstName	FirstName	firstname	First_Name
lastName	LastName	lastname	Last_Name
email	Email	email	Email
phone	Phone	phone	Phone
rut	RUT__c	rut	RUT
totalDebt	Total_Debt__c	total_debt	Total_Debt
platformUserId	Platform_User_ID__c	platform_user_id	Platform_User_ID

Deudas

Plataforma	Salesforce	HubSpot	Zoho
name	Name	dealname	Name
amount	Amount__c	amount	Amount
status	Status__c	dealstage	Status
dueDate	Due_Date__c	closedate	Due_Date
originalCreditor	Original_Creditor__c	original_creditor	Original_Creditor
contactId	Contact__c	associations.contacts	Contact_Name

Sincronización

Sincronización Completa

Importa TODOS los datos desde el CRM:

```
const { fullSync } = useCRM();

const result = await fullSync({
  debtorFilters: { hasDebt: true },
  debtFilters: { status: 'Pending' },
  includeHistory: true
});

console.log(`Importados: ${result.summary.debtors} deudores, ${result.summary.debts} deudas`);
```

Sincronización Incremental

Solo actualiza cambios recientes:

```
const { incrementalSync } = useCRM();

// Obtener cambios de las últimas 24 horas
const since = new Date(Date.now() - 24 * 60 * 60 * 1000);
const result = await incrementalSync(since);

console.log('Cambios sincronizados:', result.summary);
```

Sincronización Automática

Configura sincronización automática cada hora:


```
useEffect(() => {
  const interval = setInterval(async () => {
    const lastSync = localStorage.getItem('lastCRMSync');
    const since = lastSync ? new Date(lastSync) : new Date(Date.now() - 7 * 24 * 60 * 60 * 1000);

    await incrementalSync(since);
    localStorage.setItem('lastCRMSync', new Date().toISOString());
  }, 60 * 60 * 1000); // Cada hora

  return () => clearInterval(interval);
}, []);
```

Troubleshooting

Salesforce

Error: "INVALID_SESSION_ID"

- Tu access token expiró
- Genera un nuevo token siguiendo el Paso 4
- Los tokens expiran cada 90 días por defecto

Error: "ENTITY_IS_DELETED"

- Estás intentando actualizar un registro eliminado
- Verifica que el ID sea correcto

Error: "FIELD_CUSTOM_VALIDATION_EXCEPTION"

- Campos personalizados no existen
- Crea los campos siguiendo el Paso 5

HubSpot

Error: "401 Unauthorized"

- Access token inválido o expirado
- Los tokens de Private Apps no expiran, pero verifica que sea correcto

Error: "PROPERTY_DOESNT_EXIST"

- Propiedad personalizada no existe
- Crea las propiedades siguiendo el Paso 4

Error: "RATE_LIMIT_EXCEEDED"

- Has excedido el límite de API calls
- Free: 100 requests/10 segundos
- Paid: Límites mayores según tu plan

Zoho

Error: "INVALID_TOKEN"

- Access token expirado (válido 1 hora)
- Usa el refresh_token para obtener uno nuevo

Error: "MANDATORY_NOT_FOUND"

- Falta un campo obligatorio
- Verifica que todos los campos requeridos estén presentes

Error: "INVALID_DATA"

- Formato de datos incorrecto
- Verifica tipos de datos (currency, date, etc.)

Mejores Prácticas

1. Manejo de Tokens

```
// ✅ Implementa refresh de tokens automático
const refreshToken = async () => {
  // Lógica para refrescar token
  const newToken = await getNewToken();
  localStorage.setItem('crm_token', newToken);
};

// Verifica expiración antes de cada request
if (isTokenExpired()) {
  await refreshToken();
}
```

2. Batch Operations

```
// ✅ Procesa en lotes para mejor rendimiento
const debtors = [...]; // 1000 deudores
const batchSize = 100;

for (let i = 0; i < debtors.length; i += batchSize) {
  const batch = debtors.slice(i, i + batchSize);
  await syncDebtors(batch);
  await sleep(1000); // Pausa entre lotes
}
```

3. Error Handling

```
// ✅ Manejo robusto de errores
try {
  const result = await syncDebtor(data);
  if (!result.success) {
    // Log error pero continúa
    console.error('Error parcial:', result.error);
  }
} catch (error) {
  // Error crítico
  console.error('Error crítico:', error);
  // Notificar al equipo
}
```

4. Logging

```
// ✅ Log todas las operaciones
console.log('[CRM] Iniciando sincronización...');
console.log('[CRM] ✅ Sincronizados 150 deudores');
console.log('[CRM] ❌ Falló actualización de 3 registros');
```

Recursos Adicionales

- [Salesforce REST API Documentation](https://developer.salesforce.com/docs/atlas.en-us.api_rest.meta/api_rest/) (https://developer.salesforce.com/docs/atlas.en-us.api_rest.meta/api_rest/)
 - [HubSpot API Reference](https://developers.hubspot.com/docs/api/overview) (https://developers.hubspot.com/docs/api/overview)
 - [Zoho CRM API Documentation](https://www.zoho.com/crm/developer/docs/api/v3/) (https://www.zoho.com/crm/developer/docs/api/v3/)
-

Versión: 1.0.0

Última actualización: Octubre 2025