

# Configuración de Google Sheets - WMC Soluciones Metálicas

---

Esta guía te llevará paso a paso para configurar la sincronización con Google Sheets.

## Tabla de Contenidos

1. Prerrequisitos
  2. Configuración en Google Cloud Console
  3. Configuración del Spreadsheet
  4. Configuración de la Aplicación
  5. Pruebas
  6. Solución de Problemas
- 

## Prerrequisitos

- Una cuenta de Google
  - Acceso a [Google Cloud Console](#)
  - El proyecto WMC descargado en tu computadora
- 

## Configuración en Google Cloud Console

### Paso 1: Crear un Proyecto

1. Ve a [Google Cloud Console](#)
2. Haz clic en el selector de proyectos (arriba a la izquierda)
3. Clic en **"Nuevo Proyecto"**
4. Nombre del proyecto:
5. Clic en **"Crear"**

### Paso 2: Habilitar Google Sheets API

1. En el menú lateral, ve a **"APIs y servicios"** → **"Biblioteca"**
2. Busca:

3. Haz clic en el resultado y luego en **"Habilitar"**

### Paso 3: Crear Credenciales (API Key)

1. Ve a **"APIs y servicios" → "Credenciales"**
2. Clic en **"Crear credenciales" → "Clave de API"**
3. Copia la API Key que aparece (la necesitarás después)
4. Clic en **"Restringir clave"** para mayor seguridad:

#### Restricciones de aplicación:

- Selecciona: **"Referentes HTTP (sitios web)"**
- Agrega estos referentes:

```
http://localhost:*  
http://127.0.0.1:*  
https://tu-dominio.com/*
```

#### Restricciones de API:

- Selecciona: **"Restringir clave"**
- Marca: **"Google Sheets API"**

5. Clic en **"Guardar"**

### Paso 4: Crear OAuth Client ID

1. Primero, configura la **Pantalla de consentimiento OAuth**:

- Ve a **"APIs y servicios" → "Pantalla de consentimiento de OAuth"**
- Tipo de usuario: **"Externo"**
- Clic en **"Crear"**

#### 2. Información de la aplicación:

- Nombre de la aplicación:
- Correo de asistencia: tu correo
- Dominio de la aplicación: (opcional por ahora)
- Correo del desarrollador: tu correo
- Clic en **"Guardar y continuar"**

### 3. Permisos (Scopes):

- Clic en **"Añadir o quitar permisos"**
- Busca: `Google Sheets API`
- Selecciona: `https://www.googleapis.com/auth/spreadsheets`
- Clic en **"Actualizar"** y luego **"Guardar y continuar"**

### 4. Usuarios de prueba (mientras estés en Testing):

- Clic en **"Agregar usuarios"**
- Agrega tu correo y el de cualquier persona que vaya a probar
- Clic en **"Guardar y continuar"**

### 5. Resumen:

- Revisa y clic en **"Volver al panel"**

### 6. Ahora crea el OAuth Client ID:

- Ve a **"APIs y servicios" → "Credenciales"**
- Clic en **"Crear credenciales" → "ID de cliente de OAuth 2.0"**
- Tipo de aplicación: **"Aplicación web"**
- Nombre: `WMC Web Client`

### Orígenes de JavaScript autorizados:

```
http://localhost:3000
http://localhost:8080
http://localhost:5500
http://127.0.0.1:3000
http://127.0.0.1:8080
http://127.0.0.1:5500
```

(Agrega tu dominio de producción cuando lo tengas)

### URLs de redireccionamiento autorizados:

- Puedes dejar esto vacío para aplicaciones JavaScript del lado del cliente

### 7. Clic en **"Crear"**

### 8. ¡IMPORTANTE! Copia el **Client ID** que aparece (empieza con algo como `123456-abc...apps.googleusercontent.com`)

## Configuración del Spreadsheet

### Paso 1: Crear el Spreadsheet

1. Ve a [Google Sheets](#)
2. Crea un nuevo spreadsheet
3. Nómbralo: `WMC - Base de Datos Cotizaciones`

### Paso 2: Crear las Hojas (Pestañas)

Crea 5 hojas con estos nombres **exactos**:

1. **Clients**
2. **Materials**
3. **Labor**
4. **Products**
5. **Quotes**

### Paso 3: Configurar Headers (Fila 1 de cada hoja)

#### Hoja: Clients

```
A1: ID
B1: Nombre
C1: NIT
D1: Contacto
E1: Teléfono
F1: Email
G1: Ciudad
H1: LastModified
```

#### Hoja: Materials

```
A1: ID
B1: Código
C1: Descripción
D1: Tipo
E1: Unidad
```

```
F1: Precio  
G1: LastModified
```

### Hoja: Labor

```
A1: ID  
B1: Código  
C1: Descripción  
D1: Tipo  
E1: Unidad  
F1: Costo  
G1: LastModified
```

### Hoja: Products

```
A1: ID  
B1: Código  
C1: Nombre  
D1: Tipo  
E1: Materials (JSON)  
F1: Labor (JSON)  
G1: Precio Unitario  
H1: LastModified
```

### Hoja: Quotes

```
A1: ID  
B1: Número  
C1: ClientID  
D1: Proyecto  
E1: Fecha  
F1: Items (JSON)  
G1: Descuento General  
H1: Totals (JSON)  
I1: Observaciones
```

```
J1: Estado  
K1: LastModified
```

## Paso 4: Obtener el Spreadsheet ID

1. Mira la URL de tu spreadsheet, se ve así:

```
https://docs.google.com/spreadsheets/d/ABC123xyz456/edit#gid=0
```

2. El **Spreadsheet ID** es la parte entre `/d/` y `/edit`:

```
ABC123xyz456
```

3. **Cópialo**, lo necesitarás en el siguiente paso

## Paso 5: Compartir el Spreadsheet

1. Clic en **"Compartir"** (arriba a la derecha)
2. Asegúrate de agregar tu correo con permisos de **Editor**
3. También puedes compartirlo con otros usuarios que usarán el sistema

---

## Configuración de la Aplicación

### Paso 1: Crear el archivo de configuración

1. En la carpeta del proyecto, ve a: `dashboard/js/`
2. Copia el archivo `google-config.example.js` y renómbralo a `google-config.js`:

```
cp google-config.example.js google-config.js
```

3. Abre `google-config.js` y edita los valores:

```
export const GOOGLE_CONFIG = {  
  // Pega tu Client ID de OAuth aquí  
  CLIENT_ID: '123456-abc...apps.googleusercontent.com',  
}
```

```
// Pega tu API Key aquí
API_KEY: 'AIzaSy...',

// Pega tu Spreadsheet ID aquí
SPREADSHEET_ID: 'ABC123xyz456',

// No cambies estos valores
DISCOVERY_DOC: 'https://sheets.googleapis.com/$discovery/rest?version=v4',
SCOPES: 'https://www.googleapis.com/auth/spreadsheets',

SHEET_NAMES: {
  clients: 'Clients',
  materials: 'Materials',
  labor: 'Labor',
  products: 'Products',
  quotes: 'Quotes'
}
};
```

4. **¡IMPORTANTE!** Nunca subas este archivo a GitHub. Ya está en `.gitignore`.

## Paso 2: Agregar los scripts al HTML

1. Abre `dashboard/index.html`

2. Antes de la línea `<script src="js/app.js"></script>`, agrega:

```
<!-- Google Sheets Integration (type="module" es importante) -->
<script type="module" src="js/google-integration.js"></script>
```

3. Guarda el archivo

## Pruebas

### Paso 1: Iniciar un servidor local

**¡MUY IMPORTANTE!** No puedes abrir el archivo HTML directamente con `file://`. Debes usar un servidor local.

## Opciones:

### 1. Python (si lo tienes instalado):

```
# Python 3  
cd dashboard  
python3 -m http.server 8080
```

Luego abre: `http://localhost:8080`

### 2. Node.js (con `http-server`):

```
npm install -g http-server  
cd dashboard  
http-server -p 8080
```

Luego abre: `http://localhost:8080`

### 3. VS Code (con Live Server extension):

- Instala la extensión "Live Server"
- Click derecho en `index.html`
- "Open with Live Server"

## Paso 2: Primera sincronización

1. Abre la aplicación en el navegador
2. Si todo está configurado correctamente, verás una opción para conectar Google Sheets en el sidebar
3. Haz clic en el botón de sincronización o configuración
4. Se abrirá un modal - clic en **"Iniciar sesión con Google"**
5. Selecciona tu cuenta de Google
6. **Verás una advertencia** que dice "Google no ha verificado esta aplicación":
  - Esto es normal cuando estás en modo de prueba
  - Clic en **"Avanzado"**
  - Clic en **"Ir a WMC Sistema de Cotizaciones (no seguro)"**



7. Acepta los permisos solicitados
8. Selecciona una opción:
  - **"Subir datos locales"** - Si ya tienes datos en localStorage
  - **"Descargar desde Google Sheets"** - Si ya tienes datos en el spreadsheet
9. Clic en **"Inicializar Sincronización"**
10. ¡Listo! Ahora tus datos se sincronizarán automáticamente cada 5 minutos y cuando vuelvas a estar online.

### **Paso 3: Verificar sincronización**

1. Crea un nuevo cliente o cotización en la aplicación
2. Espera unos segundos
3. Revisa tu Google Spreadsheet - deberías ver los datos allí
4. Si editas algo en el spreadsheet, la app lo descargará en la próxima sincronización

---

## **Solución de Problemas**

### **Error: "redirect\_uri\_mismatch"**

**Problema:** El origen desde donde estás accediendo no está autorizado.

**Solución:**

1. Ve a Google Cloud Console → Credenciales
2. Edita tu OAuth Client ID
3. Agrega el origen exacto que estás usando (incluyendo puerto):
  - Ejemplo: `http://localhost:8080`
  - Ejemplo: `http://127.0.0.1:5500`

### **Error: "This app is blocked"**

**Problema:** La pantalla de consentimiento no está configurada correctamente.

**Solución:**

1. Ve a "Pantalla de consentimiento de OAuth"
2. Asegúrate de haber agregado el scope: `https://www.googleapis.com/auth/spreadsheets`

3. Agrega tu correo en "Usuarios de prueba"

### Error: "API key not valid"

**Problema:** La API Key está mal copiada o las restricciones son muy estrictas.

#### **Solución:**

1. Verifica que copiaste la API Key completa
2. En Cloud Console, edita la API Key
3. Temporalmente, quita todas las restricciones para probar
4. Si funciona, agrega las restricciones de nuevo una por una

### Los datos no se sincronizan

**Problema:** La sincronización automática no funciona.

#### **Solución:**

1. Abre la consola del navegador (F12)
2. Busca errores en rojo
3. Verifica que estés online (revisa el ícono de sincronización)
4. Haz clic en "Sincronizar ahora" manualmente
5. Si hay errores, cópialos y consúltalos

### Error: "Spreadsheet not found"

**Problema:** El Spreadsheet ID es incorrecto o no tienes acceso.

#### **Solución:**

1. Verifica el Spreadsheet ID en `google-config.js`
2. Asegúrate de que el spreadsheet esté compartido contigo (el usuario autenticado)
3. Revisa los permisos del spreadsheet

### Error CORS

**Problema:** Estás intentando abrir el archivo HTML directamente con `file://`

#### **Solución:**

- DEBES usar un servidor local (ver Paso 1 de Pruebas)

- Google APIs no funcionan con `file://` por seguridad
- 

## Notas Adicionales

### Modo Offline

- La aplicación funciona completamente offline
- Los cambios se guardan en localStorage
- Cuando vuelvas a estar online, se sincronizarán automáticamente

### Conflictos

- Si editas lo mismo en dos lugares, la versión más reciente gana
- Se usa el timestamp `LastModified` para decidir

### Seguridad

- **NUNCA** subas `google-config.js` a GitHub
- Las API Keys tienen restricciones de dominio
- OAuth tokens se manejan de forma segura en memoria

### Sincronización Automática

- Se sincroniza cada 5 minutos cuando estás online
- También sincroniza cuando vuelves a estar online
- Puedes sincronizar manualmente con el botón

### Verificación de la App (Producción)

Si quieres publicar la app para más de 100 usuarios:

1. Ve a "Pantalla de consentimiento de OAuth"
  2. Clic en **"Publicar aplicación"**
  3. Google te pedirá verificar la app
  4. Sigue las instrucciones para el proceso de verificación
  5. Puede tomar varios días
- 

## Soporte

Si tienes problemas:

1. Revisa la consola del navegador (F12 → Console)
  2. Copia cualquier error que veas
  3. Consulta la sección "Solución de Problemas"
  4. Verifica que todos los pasos de configuración estén completos
- 

**¡Listo!** Ahora tienes tus datos sincronizados con Google Sheets automáticamente.

