



# Integración de Google Calendar y Meet - Sistema 1 a 1

---

## ✓ Implementación Completada

---

Se ha implementado exitosamente la integración de **Google Calendar y Google Meet** en el sistema de prácticas 1 a 1 de SpeaklyPlan.

---

## 🎯 Funcionalidades Implementadas

---

### 1. Conexión de Cuenta Google

- Cada usuario puede conectar su propia cuenta de Google
- Autenticación OAuth2 segura
- Almacenamiento encriptado de tokens
- Renovación automática de tokens cuando expiran

### 2. Creación Automática de Eventos

- Al programar una práctica 1 a 1, se crea automáticamente:
- ✓ Evento en Google Calendar
- ✓ Link de Google Meet incluido
- ✓ Invitación al compañero de práctica
- ✓ Recordatorios automáticos (1 día antes, 1 hora antes, 10 minutos antes)

### 3. Gestión de Eventos

- Los eventos incluyen:
  - Título descriptivo con el tema de práctica
  - Descripción con consejos útiles
  - Duración configurable (por defecto 30 minutos)
  - Zona horaria (America/Bogota)
- 



## Configuración de Google Cloud Console

---

Para que la integración funcione, necesitas configurar un proyecto en Google Cloud Console:

### Paso 1: Crear Proyecto en Google Cloud Console

1. Ve a [Google Cloud Console](https://console.cloud.google.com/) (<https://console.cloud.google.com/>)
2. Crea un nuevo proyecto o selecciona uno existente
3. Habilita las siguientes APIs:
  - **Google Calendar API**
  - **Google People API** (opcional, para información de perfil)

## Paso 2: Configurar OAuth 2.0

1. Ve a **APIs & Services > Credentials**
2. Haz clic en **Create Credentials > OAuth client ID**
3. Selecciona **Web application**
4. Configura los siguientes campos:

### Authorized JavaScript origins:

```
http://localhost:3000
https://speaklyplan.abacusai.app
```

### Authorized redirect URIs:

```
http://localhost:3000/api/google/callback
https://speaklyplan.abacusai.app/api/google/callback
```

1. Guarda y descarga las credenciales

## Paso 3: Configurar Pantalla de Consentimiento OAuth

1. Ve a **APIs & Services > OAuth consent screen**
2. Configura:
  - **App name:** SpeaklyPlan
  - **User support email:** Tu email
  - **Developer contact information:** Tu email

3. Agrega los siguientes **scopes**:
  - `https://www.googleapis.com/auth/calendar`
  - `https://www.googleapis.com/auth/calendar.events`

## Paso 4: Agregar Usuarios de Prueba (Modo Desarrollo)

Si tu app está en modo "Testing":

1. Ve a **OAuth consent screen**
2. En **Test users**, agrega los emails de usuarios que podrán conectar Google Calendar
3. Cada usuario debe ser agregado manualmente

**Nota:** Para uso en producción, deberás solicitar verificación de Google.



## Variables de Entorno

Actualiza el archivo `.env` con tus credenciales de Google:

```
# Google Calendar Integration
GOOGLE_CLIENT_ID=tu-client-id-aqui.apps.googleusercontent.com
GOOGLE_CLIENT_SECRET=tu-client-secret-aqui
GOOGLE_REDIRECT_URI=https://speaklyplan.abacusai.app/api/google/callback
```



### IMPORTANTE:

- Reemplaza `YOUR_GOOGLE_CLIENT_ID` y `YOUR_GOOGLE_CLIENT_SECRET` con tus credenciales reales

- Para desarrollo local usa: `http://localhost:3000/api/google/callback`
- Para producción usa: `https://speaklyplan.abacusai.app/api/google/callback`



## Cambios en la Base de Datos

Se agregaron los siguientes campos al modelo `User` :

```
model User {
  // ... otros campos ...

  // Google Calendar Integration
  googleAccessToken    String? @db.Text
  googleRefreshToken   String? @db.Text
  googleTokenExpiresAt DateTime?
  googleCalendarId     String?
  googleConnected      Boolean @default(false)
}
```

La migración ya fue aplicada exitosamente a la base de datos.



## Interfaz de Usuario

### En la Página de Prácticas 1 a 1 ( `/practice` )

Si el usuario **NO** tiene Google Calendar conectado:

- Se muestra un banner informativo azul
- Botón “Conectar Google Calendar”
- Al hacer clic, redirige a la página de autorización de Google

Si el usuario **YA** tiene Google Calendar conectado:

- El banner no se muestra
- Indicador de “Conectado” visible
- Las prácticas se crean automáticamente con Google Meet

## Flujo de Usuario

### 1. Conectar Google Calendar

```

Usuario → Página Prácticas → Click "Conectar Google Calendar"
↓
Redirige a Google OAuth
↓
Usuario autoriza la aplicación
↓
Google redirige de vuelta con código
↓
App intercambia código por tokens
↓
Tokens se guardan en la base de datos
↓
Usuario redirigido a /practice?connected=true
↓
Mensaje de éxito: "¡Google Calendar conectado!"

```

### 2. Crear Práctica con Google Meet

```

Usuario → Programa práctica 1 a 1 → Selecciona fecha/hora/tema
↓
Sistema verifica si Google Calendar está conectado
↓
Si está conectado:
├─ Crea evento en Google Calendar
├─ Genera link de Google Meet
├─ Envía invitación al compañero
├─ Guarda evento en base de datos con calendarEventId
└─ Muestra mensaje: "Sesión programada con Google Meet"
↓
Si NO está conectado:
└─ Crea práctica sin Meet link

```

## Archivos Creados/Modificados

### Nuevos Archivos:

- `lib/google-calendar.ts` - Utilidades de Google Calendar
- `app/api/google/auth/route.ts` - Genera URL de autenticación
- `app/api/google/callback/route.ts` - Maneja callback de OAuth
- `app/api/google/disconnect/route.ts` - Desconecta Google Calendar
- `app/api/google/status/route.ts` - Verifica estado de conexión
- `app/dashboard/practicas/_components/google-calendar-connect.tsx` - Componente de conexión






### Archivos Modificados:

- `prisma/schema.prisma` - Agregados campos de Google Calendar al User
- `lib/services/google-calendar-service.ts` - Actualizado para usar nuevos campos
- `app/api/practice/sessions/route.ts` - Crea eventos automáticamente
- `app/practice/page.tsx` - Pasa estado de Google Calendar

- `components/practice/connect-calendar-button.tsx` - Actualizado con nuevas rutas

---

## Seguridad

-  Los tokens se almacenan encriptados en la base de datos
-  Los tokens se renuevan automáticamente cuando expiran
-  Los usuarios solo pueden acceder a su propio calendario
-  OAuth2 con scopes mínimos necesarios
-  HTTPS obligatorio en producción

---

## Pruebas

### Para probar la integración:

#### 1. Verifica las variables de entorno:

```
bash
cd /home/ubuntu/speaklyplan/nextjs_space
cat .env | grep GOOGLE
```

#### 2. Inicia el servidor:

```
bash
yarn dev
```

#### 3. Prueba el flujo:

- Ve a `/practice`
- Haz clic en “Conectar Google Calendar”
- Autoriza la aplicación en Google
- Verifica que se muestre el mensaje de éxito
- Programa una práctica 1 a 1
- Verifica que el link de Google Meet se genere automáticamente
- Revisa tu Google Calendar para ver el evento

---

## Solución de Problemas

### Error: “redirect\_uri\_mismatch”

**Causa:** La URI de redirección no coincide con la configurada en Google Cloud Console.

**Solución:**

1. Ve a Google Cloud Console > Credentials
2. Verifica que la URI esté exactamente como: `https://speaklyplan.abacusai.app/api/google/callback`
3. Sin espacios, sin barras adicionales al final

### Error: “access\_denied”

**Causa:** El usuario canceló la autorización o no está en la lista de usuarios de prueba.

**Solución:**

1. Agrega el email del usuario en “Test users” en Google Cloud Console
2. O solicita verificación de la app para uso público

**Error: “token expired”**

**Causa:** El token de acceso expiró.





**Solución:** El sistema debería renovarlo automáticamente. Si persiste:

1. El usuario puede desconectar y reconectar su cuenta
2. Verifica que `googleRefreshToken` esté guardado correctamente



## Métricas de Éxito

Con esta implementación, esperamos:

-  Mayor adopción del sistema de prácticas 1 a 1
-  Reducción de prácticas canceladas (gracias a los recordatorios)
-  Mejor experiencia de usuario
-  Menos fricción al programar sesiones



## Próximos Pasos

1. **Configurar credenciales de Google Cloud Console**
2. **Probar la integración en desarrollo**
3. **Agregar usuarios de prueba**
4. **Desplegar a producción**
5. **Solicitar verificación de Google (para uso público)**



## Soporte

Si tienes problemas con la integración:

1. Verifica los logs del servidor
2. Revisa la consola del navegador
3. Verifica que las credenciales de Google sean correctas
4. Asegúrate de que las APIs estén habilitadas en Google Cloud Console

---

**¡La integración está lista para usarse! 🚀**