

# GUÍA COMPLETA DE CONFIGURACIÓN SUPABASE PARA VIRA

Esta guía te llevará paso a paso para configurar VIRA con Supabase, eliminando completamente la dependencia de Prisma.

## PASO 1: CREAR PROYECTO EN SUPABASE

### 1.1 Crear Cuenta y Proyecto

1. Ve a [supabase.com](https://supabase.com) (<https://supabase.com>)
2. Regístrate o inicia sesión
3. Crea un nuevo proyecto:
  - **Name:** `VIRA Production` (o el nombre que prefieras)
  - **Database Password:** Genera una contraseña segura y guárdala
  - **Region:** Elige la región más cercana a Chile (São Paulo o North Virginia)

### 1.2 Obtener Variables de Entorno

Una vez creado el proyecto, ve a **Settings > API** y copia:

```
# === BASE DE DATOS SUPABASE ===  
NEXT_PUBLIC_SUPABASE_URL=https://tu-proyecto-id.supabase.co  
NEXT_PUBLIC_SUPABASE_ANON_KEY=tu_anon_key_aquí  
SUPABASE_SERVICE_ROLE_KEY=tu_service_role_key_aquí
```

## PASO 2: CREAR EL SCHEMA DE BASE DE DATOS

### 2.1 Ejecutar el Schema SQL

1. En tu proyecto Supabase, ve a **SQL Editor**
2. Crea una nueva query
3. Copia y pega TODO el contenido del archivo `database/supabase_schema.sql`
4. Ejecuta la query (puede tomar 1-2 minutos)

### 2.2 Verificar Tablas Creadas

Ve a **Database > Tables** y verifica que se crearon:

- ☒ `users` - Usuarios del sistema
- ☒ `accounts` - Cuentas OAuth
- ☒ `sessions` - Sesiones activas
- ☒ `newscast_templates` - Plantillas de noticieros
- ☒ `news_reports` - Reportes generados
- ☒ `news_sources` - Fuentes de noticias
- ☒ `scraped_news` - Noticias extraídas
- ☒ `audio_library` - Biblioteca de audio
- ☒ `ad_campaigns` - Campañas publicitarias
- ☒ `cloned_voices` - Voces clonadas
- ☒ `token_usage` - Uso de tokens/costos

- ☒ `daily_metrics` - Métricas diarias
- ☒ Y más...



## PASO 3: CONFIGURAR AUTENTICACIÓN

### 3.1 Habilitar Proveedores OAuth

Ve a **Authentication > Providers**:

#### Google OAuth:

1. Habilita Google
2. Configura:
  - **Client ID**: Tu Google Client ID
  - **Client Secret**: Tu Google Client Secret
  - **Authorized redirect URLs**: `https://tu-dominio.com/api/auth/callback/google`

#### Configuración adicional (opcional):

- GitHub, Facebook, Twitter, etc.

### 3.2 Configurar Políticas de Seguridad (RLS)

Las políticas ya están incluidas en el schema. Verifica en **Database > Tables** que RLS esté habilitado en:

- ☒ `users`
- ☒ `newscast_templates`
- ☒ `news_reports`
- ☒ `news_sources`
- ☒ Y otras tablas de usuario

### 3.3 Configurar NextAuth

El archivo `lib/auth.ts` ya está configurado para usar Supabase. Solo necesitas agregar tus OAuth credentials a `.env`:

```
# === NEXTAUTH OAUTH ===
GOOGLE_CLIENT_ID=tu_google_client_id
GOOGLE_CLIENT_SECRET=tu_google_client_secret

# También puedes agregar:
GITHUB_CLIENT_ID=tu_github_client_id
GITHUB_CLIENT_SECRET=tu_github_client_secret
```



## PASO 4: ACTUALIZAR VARIABLES DE ENTORNO

### 4.1 Archivo .env Completo

Actualiza tu archivo `.env` con TODAS estas variables:

```
# === BASE DE DATOS SUPABASE ===
NEXT_PUBLIC_SUPABASE_URL=https://tu-proyecto-id.supabase.co
NEXT_PUBLIC_SUPABASE_ANON_KEY=tu_anon_key_aquí
SUPABASE_SERVICE_ROLE_KEY=tu_service_role_key_aquí

# === NEXTAUTH CONFIGURATION ===
NEXTAUTH_SECRET="k3l0kVTrYesCYXQRFMGstZBXWYAdAkHb"
NEXTAUTH_URL=https://tu-dominio.com

# === OAUTH PROVIDERS ===
GOOGLE_CLIENT_ID=tu_google_client_id
GOOGLE_CLIENT_SECRET=tu_google_client_secret

# === IA Y PROCESAMIENTO ===
ABACUSAI_API_KEY="77af0fb805d34069b609ef8baea62041"

# === SÍNTESIS DE VOZ ===
ELEVENLABS_API_KEY=your_elevenlabs_key_here
AZURE_SPEECH_KEY=your_azure_speech_key_here
AWS_ACCESS_KEY_ID=your_aws_access_key_here
AWS_SECRET_ACCESS_KEY=your_aws_secret_key_here

# === ALMACENAMIENTO ===
AWS_BUCKET_NAME=vira-audio-files
AWS_FOLDER_PREFIX=production/

# === SCRAPING & NEWS ===
OPENWEATHER_API_KEY=your_openweather_key_here
EMOL_RSS_URL=https://www.emol.com/rss/rss.asp
LATERCERA_RSS_URL=https://www.latercera.com/feed/
BIOBIO_RSS_URL=https://www.biobiochile.cl/especial/rss/index.xml

# === SOCIAL MEDIA INTEGRATIONS ===
TWITTER_BEARER_TOKEN=your_twitter_bearer_token_here
FACEBOOK_APP_ID=your_facebook_app_id_here
INSTAGRAM_APP_ID=your_instagram_app_id_here

# === NOTIFICATION SERVICES ===
SENDGRID_API_KEY=your_sendgrid_key_here
SENDGRID_FROM_EMAIL=notificaciones@tu-dominio.com

# === PRODUCTION SETTINGS ===
NODE_ENV=production
```



## PASO 5: REMOVER PRISMA COMPLETAMENTE

### 5.1 Eliminar Archivos de Prisma

```
rm -rf prisma/
rm -f yarn.lock # Para regenerar dependencias
```

### 5.2 Actualizar package.json

Elimina estas dependencias de `package.json` :

```
{
  "devDependencies": {
    "prisma": "ELIMINAR",
  },
  "dependencies": {
    "@prisma/client": "ELIMINAR",
    "@next-auth/prisma-adapter": "ELIMINAR"
  }
}
```

## 5.3 Instalar/Verificar Dependencias Supabase

```
yarn add @supabase/supabase-js @supabase/auth-helpers-nextjs @supabase/auth-helpers-shared
yarn install
```



## PASO 6: ACTUALIZAR NEXTAUTH

### 6.1 Actualizar [...nextauth].ts

Reemplaza el contenido de `app/api/auth/[...nextauth]/route.ts`:

```
import NextAuth from 'next-auth'
import { authOptions } from '@lib/auth'

const handler = NextAuth(authOptions)
export { handler as GET, handler as POST }
```

### 6.2 Verificar Configuración de Auth





El archivo `lib/auth.ts` ya está configurado correctamente para usar Supabase.



## PASO 7: USAR LAS NUEVAS APIS

### 7.1 APIs Disponibles

Las siguientes APIs ya están implementadas con Supabase:

-  `/api/templates` - Gestionar plantillas
-  `/api/reports` - Gestionar reportes
-  `/api/news-sources` - Gestionar fuentes
-  `/api/generate-newscast` - Generar noticieros (actualizada)

### 7.2 Funciones de Supabase Disponibles

En `lib/supabase.ts` tienes acceso a:

```
// Usuarios
getUser(userId)
updateUser(userId, updates)

// Plantillas
createNewscastTemplate(template)
getUserNewscastTemplates(userId)
deleteNewscastTemplate(templateId, userId)

// Reportes
createNewsReport(report)
updateNewsReport(reportId, updates)
getUserNewsReports(userId, limit)

// Fuentes de noticias
createNewsSource(source)
getUserNewsSources(userId)
getNewsSourcesByRegion(region)

// Y muchas más...
```



## PASO 8: DEPLOYMENT EN PRODUCCIÓN

### 8.1 Variables de Entorno en Vercel/Netlify

Configura TODAS las variables de entorno en tu plataforma de deployment.

### 8.2 Configurar Dominio en Supabase

1. Ve a **Authentication > URL Configuration**
2. Agrega tu dominio de producción a **Site URL**
3. Agrega callback URLs para OAuth: `https://tu-dominio.com/api/auth/callback/*`

### 8.3 Configurar CORS (si es necesario)

En **Settings > API**, configura CORS origins si tienes problemas de CORS.



## PASO 9: VERIFICACIÓN Y TESTING

### 9.1 Verificar Conexión

Prueba que la conexión funcione:

```
yarn dev
```

Ve a `http://localhost:3000` y:

1. ☒ Intenta hacer login
2. ☒ Crea una plantilla de noticiero
3. ☒ Genera un noticiero
4. ☒ Verifica que los datos se guarden en Supabase

### 9.2 Verificar en Dashboard de Supabase

Ve a **Database > Table Editor** y confirma que:

- Se crean usuarios al hacer login

- Se guardan plantillas al crearlas
- Se guardan reportes al generar noticieros



## PASO 10: MONITOREO Y MÉTRICAS

### 10.1 Dashboard de Supabase

Usa el dashboard para monitorear:

- **Database:** Usage y performance
- **Auth:** Usuarios activos y registros
- **API:** Requests y errors
- **Logs:** Debugging en tiempo real

### 10.2 Métricas de VIRA

Las vistas y funciones incluidas te permiten:

```
-- Ver estadísticas de usuarios
SELECT * FROM user_stats;

-- Ver noticias recientes
SELECT * FROM recent_news;

-- Ver métricas del sistema
SELECT * FROM system_metrics;
```



## FUNCIONALIDADES AVANZADAS

### 11.1 Búsqueda Semántica (Opcional)

Si instalas la extensión `vector`, puedes habilitar búsqueda semántica de noticias con embeddings.

### 11.2 Triggers y Funciones

El schema incluye triggers automáticos para:

- Actualizar `updated_at` timestamps
- Row Level Security
- Funciones útiles como `get_user_total_cost()`

### 11.3 Backups Automáticos

Supabase hace backups automáticos, pero puedes configurar backups adicionales en **Settings > Database**.



## SOLUCIÓN DE PROBLEMAS COMUNES

### Error: “Invalid API Key”

- Verifica que `NEXT_PUBLIC_SUPABASE_ANON_KEY` esté correcta
- Asegúrate de reiniciar el servidor después de cambiar `.env`

### Error de Autenticación

- Verifica que RLS esté habilitado y las políticas configuradas
- Confirma que las callback URLs estén correctas en OAuth providers

## Error de Permisos














- Verifica que `SUPABASE_SERVICE_ROLE_KEY` esté correcta
- Confirma que el usuario tenga acceso a las tablas necesarias

## Performance Issues

- Revisa los índices en las tablas más usadas
- Configura connection pooling si tienes muchos usuarios

## CHECKLIST FINAL

Antes de ir a producción, confirma:

- ☐  Proyecto Supabase creado y configurado
- ☐  Schema SQL ejecutado correctamente
- ☐  Variables de entorno configuradas
- ☐  OAuth providers configurados
- ☐  RLS policies habilitadas
- ☐  Prisma completamente removido
- ☐  APIs funcionando con Supabase
- ☐  Login/logout funcionando
- ☐  Creación y guardado de templates
- ☐  Generación de noticieros
- ☐  Deployment en producción
- ☐  Dominio configurado en Supabase
- ☐  Monitoring y logs funcionando

**¡VIRA CON SUPABASE ESTÁ LISTO PARA PRODUCCIÓN!** 🎉

## SOPORTE

Si tienes problemas:

1. Revisa los logs en Supabase Dashboard
2. Verifica las variables de entorno
3. Consulta la documentación de [Supabase](https://supabase.com/docs) (<https://supabase.com/docs>)
4. Revisa el código en `lib/supabase.ts` para debug

**VIRA ahora usa Supabase como base de datos principal, con todas las funcionalidades optimizadas y listas para escalar.**