Mega App de Chatbot Multimodal

Una aplicación completa de chatbot con capacidades multimodales, integración RAG (Retrieval-Augmented Generation), y soporte para múltiples LLMs (Gemini y MiniMax).

🚀 Características Principales

- **Chatbot Multimodal**: Soporte para texto, audio, imágenes y archivos
- **Dual LLM**: Integración con Google Gemini y MiniMax API
- **Sistema RAG**: Búsqueda semántica en documentos utilizando embeddings vectoriales
- **Administración en Tiempo Real**: Consola de administración con capacidad de tomar control de conversaciones
- **Widget Embebible**: Widget de chat que se puede integrar en cualquier sitio web
- **Transcripción de Audio**: Conversión de voz a texto
- **Análisis de Imágenes**: Procesamiento y análisis de imágenes con IA
- **Gestión de Documentos**: Subida y gestión de documentos para el sistema RAG
- **Autenticación Segura**: Sistema completo de autenticación con Supabase Auth
- **Almacenamiento de Archivos**: Gestión de archivos con Supabase Storage

📁 Estructura del Proyecto

```
.
├── supabase/
│   ├── migrations/              # Migraciones de base de datos
│   │   ├── 0000_initial_schema.sql
│   │   ├── 0001_rls_policies.sql
│   │   ├── 0002_storage_and_helpers.sql
│   │   ├── 0003_admin_views.sql
│   │   ├── 0004_events_and_takeovers.sql
│   │   └── 0005_documents_and_rag.sql
│   ├── seed.sql                 # Datos iniciales
│   └── functions/               # Edge Functions (Deno)
│       ├── chat/
│       ├── transcribe/
│       ├── enhance/
│       ├── takeover/
│       ├── upload/
│       ├── vision/
│       ├── rag-search/
│       ├── upload-document/
│       └── manage-documents/
├── apps/
│   ├── web/                     # Widget embebible
│   │   ├── index.html
│   │   ├── style.css
│   │   └── main.js
│   └── web-admin/               # Consola de administración
│       ├── index.html
│       ├── style.css
│       └── main.js
├── docs/                        # Documentación y archivos RAG
├── .env.example                 # Variables de entorno de ejemplo
└── README.md                    # Este archivo
```

🛠️ Requisitos Previos

Antes de comenzar, asegúrate de tener instalado:

1. Supabase CLI

```
# Instalar Supabase CLI
npm install -g supabase

# O usando Homebrew (macOS)
brew install supabase/tap/supabase

# O usando Scoop (Windows)
scoop bucket add supabase https://github.com/supabase/scoop-
bucket.git
scoop install supabase
```

2. Deno (para Edge Functions)

```
# Instalar Deno
curl -fsSL https://deno.land/install.sh | sh

# O usando Homebrew (macOS)
brew install deno

# O usando Chocolatey (Windows)
choco install deno
```

3. Node.js (para servir los frontends localmente)

```
# Versión recomendada: Node.js 18 o superior
# Descargar desde: https://nodejs.org/
```

4. Cuentas y API Keys

- Cuenta de Supabase: https://supabase.com
- Google AI Studio (Gemini): https://makersuite.google.com
- MiniMax API: https://api.minimax.chat

⚙️ Configuración

1. Crear Proyecto en Supabase

1. Ve a Supabase Dashboard
2. Crea un nuevo proyecto
3. Espera a que el proyecto esté listo (puede tomar unos minutos)
4. Ve a **Settings** > **API** y copia:
   - Project URL
   - Anon public key
   - Service role key

2. Configurar Variables de Entorno

1. Copia el archivo de ejemplo:

```
cp .env.example .env
```

1. Edita el archivo `.env` con tus credenciales:

```
# Configuración de Supabase
SUPABASE_URL=https://tu-proyecto.supabase.co
SUPABASE_ANON_KEY=tu_anon_key_aqui
SUPABASE_SERVICE_ROLE_KEY=tu_service_role_key_aqui

# API Keys para LLMs
GEMINI_API_KEY=tu_gemini_key_aqui
MINIMAX_API_KEY=tu_minimax_key_aqui
```

3. Obtener API Keys

Google Gemini API Key:

1. Ve a Google AI Studio
2. Inicia sesión con tu cuenta de Google
3. Haz clic en "Create API Key"
4. Copia la clave generada

MiniMax API Key:

1. Ve a MiniMax
2. Regístrate o inicia sesión
3. Ve a la sección de API Keys
4. Genera una nueva clave
5. Copia la clave generada

🗄️ Configuración de Base de Datos

1. Inicializar Supabase Localmente

```
# Inicializar configuración de Supabase
supabase init

# Vincular con tu proyecto (usa la Project ID desde el dashboard)
supabase link --project-ref tu-project-id
```

2. Aplicar Migraciones

```
# Aplicar todas las migraciones
supabase db push

# O aplicar una migración específica
supabase migration up --file 0000_initial_schema.sql
```

3. Aplicar Datos Iniciales (Opcional)

```
# Ejecutar el script de seed
supabase db reset --linked
```

4. Verificar la Base de Datos

```
# Abrir el dashboard de la base de datos
supabase dashboard
```

🚀 Despliegue de Edge Functions

1. Desplegar Todas las Funciones

```
# Desplegar todas las Edge Functions
supabase functions deploy
```

2. Desplegar Funciones Individuales

```
# Desplegar función específica
supabase functions deploy chat
supabase functions deploy transcribe
supabase functions deploy enhance
supabase functions deploy takeover
supabase functions deploy upload
supabase functions deploy vision
supabase functions deploy rag-search
supabase functions deploy upload-document
supabase functions deploy manage-documents
```

3. Configurar Variables de Entorno para Edge Functions

```
# Configurar secrets para las Edge Functions
supabase secrets set GEMINI_API_KEY=tu_gemini_key_aqui
supabase secrets set MINIMAX_API_KEY=tu_minimax_key_aqui
supabase secrets set SUPABASE_URL=https://tu-proyecto.supabase.co
supabase secrets set
SUPABASE_SERVICE_ROLE_KEY=tu_service_role_key_aqui
```

4. Verificar Despliegue

```
# Listar funciones desplegadas
supabase functions list

# Ver logs en tiempo real
supabase functions logs --follow
```

💻 Ejecutar Frontends Localmente

Widget Embebible (`apps/web`)

1. Navegar al directorio:

```
cd apps/web
```

1. Servir localmente:

```
# Usando Python (si está instalado)
python -m http.server 3000

# O usando Node.js
npx serve -p 3000

# O usando cualquier servidor web estático
```

1. Abrir en el navegador: `http://localhost:3000`

Consola de Administración (`apps/web-admin`)

1. Navegar al directorio:

```
cd apps/web-admin
```

1. Servir localmente:

```
# Usando Python
python -m http.server 3001

# O usando Node.js
npx serve -p 3001
```

1. Abrir en el navegador: `http://localhost:3001`

🔧 Configuración Adicional

Configurar CORS en Supabase

1. Ve a **Settings** > **API** en tu proyecto de Supabase
2. En la sección "CORS origins", añade:
   - `http://localhost:3000` (para el widget)
   - `http://localhost:3001` (para la consola admin)
   - Tu dominio de producción cuando lo tengas

Configurar Storage Buckets

Las migraciones ya crean el bucket 'uploads', pero puedes verificar:

1. Ve a **Storage** en tu dashboard de Supabase
2. Verifica que existe el bucket 'uploads'
3. Configura las políticas de acceso si es necesario

Configurar Autenticación

1. Ve a **Authentication** > **Settings** en Supabase

2. Configura los providers que desees (Email, Google, etc.)
3. Actualiza las URLs de redirección según tus dominios

📚 Uso de la Aplicación

Widget Embebible

1. **Instalación**: Copia el código del widget en tu sitio web
2. **Configuración**: Actualiza la URL de Supabase en `main.js`
3. **Personalización**: Modifica los estilos en `style.css`

Consola de Administración

1. **Acceso**: Inicia sesión con una cuenta de administrador
2. **Gestión de Conversaciones**: Ve todas las conversaciones activas
3. **Tomar Control**: Asume el control de una conversación para responder directamente
4. **Gestión de Documentos**: Sube y gestiona documentos para el sistema RAG

Sistema RAG

1. **Subir Documentos**: Usa la consola de administración para subir PDFs
2. **Procesamiento Automático**: Los documentos se procesan y vectorizan automáticamente
3. **Búsqueda Semántica**: El chatbot usa estos documentos para responder preguntas

🧪 Pruebas

Probar Edge Functions

```
# Probar función de chat
curl -X POST 'https://tu-proyecto.supabase.co/functions/v1/chat' \
  -H 'Authorization: Bearer tu_anon_key' \
  -H 'Content-Type: application/json' \
  -d '{"message": "Hola", "conversation_id": "test"}'

# Probar función de transcripción
# (requiere archivo de audio en base64)

# Probar función de visión
# (requiere imagen en base64 o URL)
```

Probar Frontend

1. Abre el widget en `http://localhost:3000`
2. Inicia una conversación
3. Prueba funcionalidades:
   - Envío de mensajes de texto
   - Grabación de audio
   - Subida de imágenes
   - Subida de archivos

🐛 Solución de Problemas

Error: "Failed to fetch"

- Verifica que las URLs de Supabase sean correctas

- Asegúrate de que CORS esté configurado correctamente
- Verifica que las Edge Functions estén desplegadas

Error: "API Key not configured"

- Verifica que las API keys estén configuradas en Supabase secrets
- Usa `supabase secrets list` para verificar

Error: "Database connection failed"

- Verifica que las migraciones se hayan aplicado correctamente
- Usa `supabase db status` para verificar el estado

Error: "Permission denied"

- Verifica las políticas RLS en la base de datos
- Asegúrate de que el usuario esté autenticado correctamente

📊 Monitoreo

Logs de Edge Functions

```
# Ver logs en tiempo real
supabase functions logs --follow

# Ver logs de una función específica
supabase functions logs chat
```

Métricas de la Base de Datos

- Ve a **Reports** en tu dashboard de Supabase

- Monitorea el uso de API, conexiones de DB, y storage

🚀 Despliegue en Producción

1. Dominio y Hosting

- **Widget**: Despliega `apps/web` en cualquier hosting estático (Vercel, Netlify, etc.)
- **Admin Console**: Despliega `apps/web-admin` en un hosting estático
- **Edge Functions**: Ya están en Supabase, no requieren hosting adicional

2. Variables de Entorno de Producción

```
# Actualizar secrets en Supabase para producción
supabase secrets set WIDGET_BASE_URL=https://tu-widget-domain.com
supabase secrets set ADMIN_BASE_URL=https://tu-admin-domain.com
```

3. Configurar CORS para Producción

Añade tus dominios de producción en la configuración de CORS de Supabase.

4. SSL y Seguridad

- Asegúrate de que todos los dominios usen HTTPS
- Configura headers de seguridad apropiados
- Revisa las políticas RLS en producción

🤝 Contribución

Este es un proyecto auto-contenido. Para modificaciones:

1. Clona el repositorio

2. Haz tus cambios
3. Prueba localmente
4. Despliega los cambios

📄 Licencia

Este proyecto está disponible para uso personal y comercial.

🆘 Soporte

Para soporte técnico:

1. Revisa la documentación de Supabase
2. Consulta la documentación de Gemini API
3. Revisa la documentación de MiniMax API

¡Tu aplicación de chatbot multimodal está lista para usar! 🎉