# 🤖 Mega Chat: Chatbot Multimodal con MiniMax + Gemini

Sistema completo de chat multimodal embebible con doble integración LLM, funcionalidades admin avanzadas, RAG, y capacidades multimodales.

## 🚀 Características Principales

### Dual LLM Integration

* **MiniMax** + **Gemini** como proveedores configurables
* Selector de modelo en tiempo real (usuario elige)
* Modo **Auto** (mejor modelo según el tipo de consulta)
* Modo **Hybrid** (comparación simultánea)

### Funcionalidades Multimodales

* **Chat de Texto**: Streaming SSE en tiempo real
* **Audio**: Grabación/transcripción con MiniMax STT + Gemini Audio
* **Imágenes**: Análisis con MiniMax Vision + Gemini Vision
* **Archivos**: Soporte PDF/TXT/MD con análisis RAG

### Admin Features

* **Takeover**: Control total de conversaciones
* **Mejorador de Redacción**: Estilos (formal, conciso, amable, vendedor, neutro)
* **Dashboard Comparativo**: Métricas MiniMax vs Gemini
* **RAG Search**: Búsqueda semántica de documentos

### Tecnologías

* **Backend**: Supabase (Auth, DB, Storage, Realtime, Edge Functions)
* **Frontend**: Vanilla JS/TS mobile-first
* **Database**: PostgreSQL + pgvector
* **Security**: RLS, rate limiting, validaciones

## 📁 Estructura del Proyecto

/  
├── apps/  
│ ├── web/ # Frontend embebible  
│ │ ├── src/  
│ │ ├── dist/  
│ │ └── package.json  
│ └── web-admin/ # Consola administración  
│ ├── src/  
│ ├── dist/  
│ └── package.json  
├── supabase/  
│ ├── migrations/ # SQL migraciones  
│ ├── functions/ # Edge Functions  
│ │ ├── chat/  
│ │ ├── transcribe/  
│ │ ├── vision/  
│ │ ├── enhance/  
│ │ ├── takeover/  
│ │ ├── upload/  
│ │ └── rag-search/  
│ └── seed/ # Datos iniciales  
├── docs/  
└── .env.example

## ⚡ Setup Rápido

### 1. Instalación de Dependencias

# Instalar Supabase CLI  
npm install -g @supabase/cli  
  
# Clonar el proyecto  
git clone <tu-repo>  
cd mega-chat  
  
# Configurar variables de entorno  
cp .env.example .env  
# Editar .env con tus credenciales

### 2. Setup Supabase

# Inicializar Supabase  
supabase init  
  
# Vincular a tu proyecto  
supabase link --project-ref TU\_PROJECT\_ID  
  
# Aplicar migraciones  
supabase db reset  
  
# Desplegar edge functions  
supabase functions deploy

### 3. Build Frontends

# Frontend embebible  
cd apps/web  
npm install  
npm run build  
  
# Consola admin  
cd ../web-admin  
npm install  
npm run build

### 4. Configuración Inicial

1. **Crear bucket de Storage**:

* INSERT INTO storage.buckets (id, name, public) VALUES ('uploads', 'uploads', true);

1. **Crear usuario admin**:

* UPDATE profiles SET role = 'admin' WHERE email = 'tu-email@ejemplo.com';

1. **Activar pgvector** (para RAG):

* CREATE EXTENSION IF NOT EXISTS vector;

## 🎯 Uso

### Frontend Embebible

<!DOCTYPE html>  
<html>  
<head>  
 <title>Mi Sitio Web</title>  
</head>  
<body>  
 <!-- Tu contenido -->  
   
 <!-- Widget Chat -->  
 <script src="/apps/web/dist/embed.js"></script>  
 <script>  
 MegaChat.init({  
 supabaseUrl: 'TU\_SUPABASE\_URL',  
 supabaseKey: 'TU\_ANON\_KEY',  
 defaultModel: 'auto', // 'minimax', 'gemini', 'auto', 'hybrid'  
 theme: 'light', // 'light', 'dark'  
 language: 'es-ES'  
 });  
 </script>  
</body>  
</html>

### Consola Admin

Acceder a /apps/web-admin/dist/index.html con credenciales de admin.

## 🔧 API Keys Necesarias

### MiniMax

1. Registrarse en [MiniMax](https://api.minimax.ai)
2. Obtener API key
3. Configurar en .env

### Gemini

1. Ir a [Google AI Studio](https://aistudio.google.com)
2. Crear API key
3. Configurar en .env

### Supabase

1. Crear proyecto en [Supabase](https://supabase.com)
2. Obtener URL y keys del dashboard
3. Configurar en .env

## 📊 Endpoints API

### Edge Functions

| Endpoint | Método | Descripción |
| --- | --- | --- |
| /functions/v1/chat | POST | Chat con streaming SSE |
| /functions/v1/transcribe | POST | Transcripción de audio |
| /functions/v1/vision | POST | Análisis de imágenes |
| /functions/v1/enhance | POST | Mejorar redacción (admin) |
| /functions/v1/takeover | POST | Control conversaciones (admin) |
| /functions/v1/upload | POST | Subir archivos |
| /functions/v1/rag-search | POST | Búsqueda semántica |

### Ejemplos de Uso

// Chat streaming  
const eventSource = new EventSource('/functions/v1/chat', {  
 body: JSON.stringify({  
 conversationId: 'uuid',  
 message: 'Hola',  
 model: 'auto'  
 })  
});  
  
// Transcripción  
const { data } = await supabase.functions.invoke('transcribe', {  
 body: { audioFile: base64Audio }  
});  
  
// Análisis imagen  
const { data } = await supabase.functions.invoke('vision', {  
 body: { imageUrl: signedUrl, model: 'gemini' }  
});

## 🛡️ Seguridad

### Row Level Security (RLS)

* Todas las tablas protegidas con RLS
* Usuarios solo ven sus datos
* Admins acceso completo

### Rate Limiting

* 100 requests/15 minutos por usuario
* Validaciones de tamaño de archivo
* Moderación básica de contenido

### Separación de Claves

* SERVICE\_ROLE\_KEY solo en edge functions
* ANON\_KEY en frontend
* Nunca exponer claves sensibles al cliente

## 🚀 Despliegue en Producción

### Vercel (Frontend)

cd apps/web  
npm run build  
vercel --prod  
  
cd ../web-admin  
npm run build  
vercel --prod

### Supabase (Backend)

# Edge functions  
supabase functions deploy  
  
# Configurar dominios en Supabase dashboard  
# Actualizar CORS\_ALLOWED\_ORIGINS

## 📝 Personalización

### Estilos

Editar apps/web/src/styles/theme.css para personalizar: - Colores - Tipografía - Espaciado - Animaciones

### Modelos LLM

Configurar en supabase/functions/\_shared/models.ts: - Endpoints - Parámetros - Fallbacks

### Moderación

Ajustar reglas en supabase/functions/\_shared/moderation.ts: - Palabras prohibidas - Filtros de contenido - Límites de uso

## 🐛 Troubleshooting

### Problemas Comunes

1. **Error de CORS**

* Verificar CORS\_ALLOWED\_ORIGINS en .env

1. **Edge Functions no responden**

* supabase functions logs chat

1. **Audio no se graba**

* Verificar permisos de micrófono en navegador

1. **Imágenes no se suben**

* Verificar configuración Storage bucket 'uploads'

### Logs y Debugging

# Ver logs edge functions  
supabase functions logs --follow  
  
# Ver logs específicos  
supabase functions logs chat --follow  
  
# Check database  
supabase db logs --follow

## 📈 Métricas y Observabilidad

### Dashboard Admin

* Conversaciones activas
* Tokens consumidos
* Errores por función
* Comparativa modelos
* Latencias promedio

### Eventos Tracked

* message\_sent
* file\_uploaded
* transcription\_completed
* vision\_analysis
* admin\_takeover
* model\_switched

## 🤝 Contribución

1. Fork el repositorio
2. Crear rama feature (git checkout -b feature/nueva-funcionalidad)
3. Commit cambios (git commit -am 'Agregar nueva funcionalidad')
4. Push a la rama (git push origin feature/nueva-funcionalidad)
5. Crear Pull Request

## 📄 Licencia

MIT License - ver LICENSE para detalles.

## 🆘 Soporte

* **Issues**: GitHub Issues
* **Documentación**: /docs
* **Email**: soporte@ejemplo.com

**MiniMax Agent** - Creado para impulsar conversaciones inteligentes 🚀