

Estado font Al-eon al 11 Enero 2026

🔎 AUDITORÍA TÉCNICA COMPLETA - AL-EON CHAT

Proyecto: Consola de Chat con AL-E Core

Fecha: 11 de enero de 2026

Auditor: GitHub Copilot (Análisis Automatizado)

Solicitante: Patricia Garibay

Entorno: Producción (Netlify + Supabase + Backend Core)

📊 RESUMEN EJECUTIVO

Métricas del Proyecto

- **Archivos de código:** 139 archivos (.js/.jsx)
- **Líneas de código:** ~33,808 líneas
- **Stack tecnológico:** React 18 + Vite + Supabase + Tailwind CSS
- **Estado actual de compilación:**  **SIN ERRORES**
- **Deployment:** Netlify (<https://al-eon.com>)
- **Backend:** <https://api.al-eon.com>

Estado General

Categoría	Estado	Completitud
Infraestructura	 Funcional	95%
Autenticación	 Parcial	85%
Chat Core	 Funcional	90%
Email	 Con bugs	70%
Proyectos	 RLS crítico	60%
Reuniones	 Funcional	85%
Calendario	 RLS crítico	65%
Voz	 Parcial	75%

1 INFRAESTRUCTURA Y CONFIGURACIÓN

Stack Tecnológico (100% funcional)

```
```json
{
 "Frontend": "React 18.2.0 + Vite 4.4.5",
 "UI": "Tailwind CSS 3.4.17 + Lucide Icons + Framer Motion",
 "State": "Zustand 5.0.9",
```

```
"Router": "React Router DOM 6.30.2",
"Backend": "Supabase 2.89.0 + AL-E Core API",
"Build": "Vite (ESM)",
"Deploy": "Netlify (CI/CD automático)"
}
```

```

****  Verificado:****

- No hay errores de compilación
- Dependencias actualizadas
- Build exitoso
- Deploy automático funcionando

****  Observaciones:****

- No hay tests automatizados
- No hay documentación de API interna
- ` `.env` tiene credenciales hardcodeadas (riesgo en repo local)

MÓDULO: AUTENTICACIÓN Y USUARIOS

Estado:  ****FUNCIONAL CON LIMITACIONES** (85%)**

 Funciona:

1. ****Signup (Registro)****

- Archivo: `src/pages/SignupPage.jsx`
-  Formulario completo (email, password, nombre, apellido)
-  Validación de campos
-  Integración con Supabase Auth
-  Creación automática de perfil en `user_profiles`
-  RLS policies correctas

2. ****Login (Inicio de sesión)****

- Archivo: `src/pages/LoginPage.jsx`
-  Email + Password
-  Manejo de sesión con JWT
-  Redirección a `/chat`

3. ****Logout****

-  Implementado en múltiples componentes
-  Limpieza de sesión Supabase

4. ****Perfil de usuario****

- Archivo: `src/pages/ProfilePage.jsx`
- Edición de nombre, apellido
- Avatar upload (Supabase Storage)
- RLS policies funcionales

NO Funciona / Falta:

1. **Cambio de contraseña**

- Archivo: `src/pages/SecurityPage.jsx` línea 23
- Comentario: `// TODO: Implementar cambio de contraseña con Supabase`
- **Estado:** NO IMPLEMENTADO

2. **Recuperar contraseña**

- Archivo: `src/pages/ForgotPasswordPage.jsx`
- UI existe pero funcionalidad backend no verificada

3. **Verificación de email**

- No hay flujo de confirmación de email
- **Riesgo:** Usuarios pueden registrarse con emails falsos

4. **2FA / MFA**

- No implementado

Tecnología:

- **Supabase Auth:** JWT-based
- **Session storage:** localStorage + Supabase client
- **RLS:** Habilitado y funcional en `user_profiles`

MÓDULO: CHAT Y CONVERSACIONES

Estado: **FUNCIONAL** (90%)

Funciona:

1. **Chat principal**

- Archivo: `src/features/chat/pages/ChatPage.jsx`
- Interfaz de mensajes
- Envío/recepción en tiempo real
- Historial persistente
- Markdown rendering
- Syntax highlighting para código

2. **Gestión de conversaciones**

- Servicio: `src/services/sessionsService.js`

- Crear nueva conversación
- Listar conversaciones
- Eliminar conversación
- Renombrar conversación

3. **Proyectos en conversaciones**

- Asignar conversación a proyecto
- Filtrar por proyecto

4. **Modo de voz (Voice Mode)**

- Archivo: `src/hooks/useVoiceMode.js`
- Grabar audio desde micrófono
- Enviar a STT (Groq Whisper)
- Recibir respuesta TTS (Edge-TTS)
- Modo manos libres

! Limitaciones:

1. **Voice Mode:**

- Archivo: `useVoiceMode.js` línea 100-150
- ! Recientemente fixed: `mediaRecorder.start(1000)` para captura de chunks
- ! Requiere permisos de micrófono (no siempre funciona en todos los navegadores)
- ! Sólo funciona con conexión a internet (no offline)

2. **Historial:**

- Archivo: `src/pages/HistoryPage.jsx` línea 7
- ! Comentario: `// TODO: Integrar con backend para cargar conversaciones reales del usuario`
- **Estado:** UI dummy, datos reales no cargan

🔧 Tecnología:

- **Backend:** `https://api.al-eon.com/api/sessions`
- **STT:** Groq Whisper large-v3-turbo
- **TTS:** Edge-TTS (es-MX-DaliaNeural)
- **Storage:** Supabase `user_conversations` table
- **WebSocket:** NO (polling HTTP)

4 MÓDULO: EMAIL

Estado: ! **FUNCIONAL CON BUGS CRÍTICOS** (70%)

Funciona:

1. **Conexión de cuentas**

- Archivo: `src/features/email/components/EmailConfigWizard.jsx`
- Gmail OAuth (parcial)
- Outlook OAuth (parcial)
- IMAP/SMTP manual

2. **Lectura de correos**

- Archivo: `src/pages/EmailModulePage.jsx`
- Lista de mensajes
- Detalle de mensaje
- Carpetas (Inbox, Sent, Drafts, Spam, Trash)

3. **Envío de correos**

- Archivo: `src/features/email/components/EmailComposer.jsx`
- Nuevo mensaje
- Responder
- Responder a todos
- Reenviar
- Adjuntos

Bugs Conocidos:

1. **OAuth timeout**

- Documentado en: `FIX-OAUTH-Y-TIMEOUT-IMPLEMENTADO.md`
-  Tokens expiran después de 1 hora
-  No hay refresh automático
- **Estado:** PENDIENTE DE FIX EN BACKEND

2. **Composer z-index**

- Archivo: `EmailComposer.jsx` línea 296
- FIXED: `zIndex: 9999` (commit 4634d47)
- **Estado:** RESUELTO 10/ene/2026

3. **Folder filtering**

- Servicio: `src/services/emailService.js` línea 53
-  Comentario: `// ELIMINAR filtro is_active para debugging`
- **Estado:** Workaround activo

4. **Reply no funciona siempre**

- Documentado en: `FIX-EMAIL-REPLY-COMPOSE.md`
-  Timing issue con state de React
- Mitigado con `setTimeout(50ms)`

Limitaciones:

- **Sin búsqueda de emails****

- ****Sin filtros avanzados****
- ****Sin soporte para HTML rico (solo plain text en composer)****
- ****Sin notificaciones push de nuevos correos****
- ****Sincronización manual (no automática en background)****

Tecnología:

- ****Backend:**** `https://api.al-eon.com/api/email/*`
- ****OAuth:**** Google + Microsoft (parcial)
- ****IMAP/SMTP:**** Compatible con cualquier proveedor
- ****Storage:**** Supabase `email_accounts`, `email_messages`, `email_folders`

MÓDULO: PROYECTOS

Estado:  **CRÍTICO - RLS BLOQUEANDO FUNCIONALIDAD** (60%)**

Funciona (para owner):

1. ****CRUD básico****

- Servicio: `src/services/projectsService.js`
-  Crear proyecto
-  Listar proyectos propios
-  Editar proyecto
-  Eliminar proyecto

2. ****Documentos****

- Componente: `src/features/projects/components/ProjectDocumentsModal.jsx`
-  Subir documentos (PDF, DOCX, TXT, etc.)
-  Ver lista de documentos
-  Descargar documentos
-  Eliminar documentos
-  Storage en Supabase: `user-files/{userId}/projects/{projectId}/`

NO Funciona:

1. ****Proyectos compartidos****

- ****PROBLEMA CRÍTICO:**** Usuario 1 (owner) ve proyectos, Usuario 2 (miembro) NO ve nada
 - ****Causa:**** RLS policies bloqueando acceso
 - Documentado en: `FIX-PROJECTS-RLS-DEFINITIVO.sql`
 - ****SQL Fix creado pero NO EJECUTADO****
 - ****Estado:****  PENDIENTE DE APLICAR EN SUPABASE

2. ****Invitar miembros****

- Servicio: `src/services/projectCollaboration.js`

-  Código existe pero policies RLS bloquean funcionalidad
- **Estado:** NO VERIFICADO

3. **Permisos granulares**

-  No hay roles (viewer, editor, admin)
-  Todos los miembros son "colaboradores genéricos"

Base de Datos:

```sql

Tables:

- user\_projects (owner\_user\_id, name, description)
- project\_members (project\_id, user\_id, accepted\_at)

Problema RLS:

- Policy actual: USING (user\_id = auth.uid()) -- solo owner
  - Policy necesaria: USING (user\_id = auth.uid() OR EXISTS en project\_members)
- ```

#### **\*\* ACCIÓN REQUERIDA:**

```bash

Ejecutar en Supabase SQL Editor:

FIX-PROJECTS-RLS-DEFINITIVO.sql (líneas 1-120)

```

---

## ## MÓDULO: REUNIONES

### Estado:  **FUNCIONAL** (85%)

#### ##### Funciona:

##### 1. **Crear reunión**

- Página: `src/pages/MeetingsPage.jsx`
-  Modo live (grabar en vivo)
-  Modo upload (subir audio)
-  Metadatos (título, participantes)

##### 2. **Grabar audio**

-  MediaRecorder API
-  Subir a Supabase Storage: `meeting-recordings/`

##### 3. **Transcripción**

-  Groq Whisper (vía backend)
-  Ver transcripción completa

#### 4. \*\*Minutas\*\*

- Generación con GPT-4
- Secciones: resumen, acuerdos, pendientes, decisiones, riesgos
- Envío automático por email/telegram

#### 5. \*\*RLS policies\*\*

- `owner\_user\_id = auth.uid()`

#### ##### ! Limitaciones:

- \*\*Sin edición de minutos\*\*
- \*\*Sin compartir reuniones con otros usuarios\*\*
- \*\*Sin integraciones con Google Meet / Zoom\*\*

#### ##### 🔧 Tecnología:

- \*\*Backend:\*\* `https://api.al-eon.com/api/meetings/\*`
- \*\*Storage:\*\* Supabase `meeting-recordings` bucket
- \*\*Table:\*\* `meetings` (con campos `mode`, `status`, `participants`)

---

## ## 7 MÓDULO: CALENDARIO / AGENDA

### ### Estado: ✗ \*\*CRÍTICO - RLS BLOQUEANDO EVENTOS\*\* (65%)

#### ##### ✅ Funciona (parcial):

##### 1. \*\*CRUD eventos\*\*

- Página: `src/pages/CalendarPage.jsx`
- Servicio: `src/services/calendarService.js`
- Crear evento
- Ver eventos propios
- Editar evento
- Eliminar evento

##### 2. \*\*UI\*\*

- Vista de semana
- Vista de lista
- Modal de creación

#### ##### ✗ NO Funciona:

##### 1. \*\*Eventos NO visibles para algunos usuarios\*\*

- \*\*PROBLEMA CRÍTICO:\*\* Usuario con ID `56bc3448...` ve sus eventos, Usuario con ID `aeafa6b7...` NO ve sus propios eventos
- \*\*Causa:\*\* Policy `calendar\_events\_owner\_policy` con `cmd = ALL` está

conflictiva

- Documentado en: `FIX-CALENDAR-RLS-URGENTE.sql`
- **\*\*SQL Fix creado pero NO EJECUTADO\*\***
- **\*\*Estado:\*\***  PENDIENTE DE APLICAR EN SUPABASE

## 2. **\*\*No hay soporte para eventos compartidos\*\***

- Tabla `calendar\_events` NO tiene columna `participants`
-  No se pueden invitar usuarios a eventos

### ##### Base de Datos:

```
```sql
```

Table: calendar_events

Problema: Policies duplicadas/conflictivas

Fix:

```
DROP POLICY "calendar_events_owner_policy" ON calendar_events;
```

```
CREATE POLICY "Users can view own events" ...
```

```
```
```

### \*\* ACCIÓN REQUERIDA:\*\*

```
```bash
```

Ejecutar en Supabase SQL Editor:

```
FIX-CALENDAR-RLS-URGENTE.sql (líneas 20-67)
```

```
```
```

---

## ## MÓDULO: TAREAS

### ### Estado: \*\*FUNCIONALIDAD MÍNIMA\*\* (50%)

#### ##### Funciona:

##### 1. **\*\*Crear tarea desde email\*\***

- Botón en `EmailMessageDetail.jsx`
-  Modal de creación

##### 2. **\*\*Crear tarea desde chat\*\***

-  Botón "Crear tarea"

#### ##### NO Funciona:

- **\*\*Sin lista de tareas dedicada\*\***
- **\*\*Sin gestión de estados (pendiente, en progreso, completada)\*\***
- **\*\*Sin asignación de tareas\*\***
- **\*\*Sin fechas límite\*\***

- **\*\*Sin prioridades\*\***

**\*\*Estado:\*\*** MÓDULO INCOMPLETO

---

## ## BACKEND / CORE INTEGRATION

#### Estado:  **\*\*FUNCIONAL\*\* (90%)**

####  Endpoints Verificados:

```javascript

BASE_URL: <https://api.al-eon.com>

-  /api/sessions/* - Chat/Conversaciones
 -  /api/voice/stt - Speech to Text
 -  /api/voice/tts - Text to Speech
 -  /api/email/* - Email CRUD
 -  /api/calendar/events - Calendario
 -  /api/meetings/* - Reuniones
 -  /api/projects/* - Proyectos
- ```

 Problemas Conocidos:

1. ****OAuth tokens expiran****

- Timeout después de 1 hora
- No hay refresh automático
- ****Estado:**** BACKEND debe implementar

2. ****Timeouts en requests largos****

- Documentos grandes > 10MB
-  Timeout hardcoded a 60s
- ****Estado:**** MITIGADO con chunking

3. ****Rate limiting****

-  No implementado
- ****Riesgo:**** Abuse posible

 Autenticación:

- ****JWT via Supabase:**** `Authorization: Bearer {token}`
- ****Session:**** Manejada por Supabase client
- ****Refresh:**** Automático (Supabase)

10 BASE DE DATOS (SUPABASE)

Estado: ⚠ **FUNCIONAL CON PROBLEMAS RLS** (80%)

✓ Tablas Principales:

```sql

- ✓ user\_profiles (RLS OK)
- ✓ user\_settings (RLS OK)
- ✓ user\_conversations (RLS OK)
- ✗ user\_projects (RLS CRÍTICO)
- ✗ project\_members (RLS CRÍTICO)
- ✓ meetings (RLS OK)
- ✗ calendar\_events (RLS CRÍTICO)
- ✓ email\_accounts (RLS OK)
- ✓ email\_messages (RLS OK)
- ✓ email\_folders (RLS OK)

```

✗ Problemas RLS Críticos:

1. **user_projects + project_members**

- Recursión infinita en policies
- Usuarios invitados NO ven proyectos compartidos
- **Fix:** `FIX-PROJECTS-RLS-DEFINITIVO.sql`

2. **calendar_events**

- Policy `calendar_events_owner_policy` conflictiva
- Algunos usuarios NO ven sus propios eventos
- **Fix:** `FIX-CALENDAR-RLS-URGENTE.sql`

3. **meetings**

- Sin soporte para compartir con participantes
- Solo owner ve reunión
- **Fix:** `FIX-MEETINGS-RLS-DEFINITIVO.sql`

✓ Storage Buckets:

```sql

- ✓ user-files (RLS OK) - Documentos de proyectos
- ✓ meeting-recordings (RLS OK) - Audios de reuniones
- ✓ avatars (RLS OK) - Fotos de perfil

```

1 1 BUGS CRÍTICOS PENDIENTES

P0 (Bloqueantes en Producción)

● P0-1: Proyectos compartidos no visibles

- **Archivo:** Toda la funcionalidad de `projectsService.js`
- **Causa:** RLS policies incorrectas
- **Fix disponible:** `FIX-PROJECTS-RLS-DEFINITIVO.sql` ✓
- **Estado:** ✗ **NO APLICADO EN SUPABASE****
- **Impacto:** Colaboración multi-usuario completamente rota

● P0-2: Eventos de calendario no visibles

- **Archivo:** `calendarService.js`, `CalendarPage.jsx`
- **Causa:** Policy `calendar_events_owner_policy` conflictiva
- **Fix disponible:** `FIX-CALENDAR-RLS-URGENTE.sql` ✓
- **Estado:** ✗ **NO APLICADO EN SUPABASE****
- **Impacto:** Usuario `aeafa6b7...` NO ve su evento del 6/ene

● P0-3: OAuth tokens expiran sin refresh

- **Archivo:** `emailService.js`, backend OAuth
- **Causa:** Backend no refresca tokens automáticamente
- **Fix disponible:** ✗ NO
- **Estado:** ✗ **PENDIENTE DE BACKEND****
- **Impacto:** Usuarios deben reconectar Gmail/Outlook cada hora

P1 (Alta prioridad)

🟡 P1-1: Voice mode no captura audio consistentemente

- **Archivo:** `useVoiceMode.js` línea 100-150
- **Causa:** `mediaRecorder.start()` sin timeslice
- **Fix aplicado:** ✓ `mediaRecorder.start(1000)` (commit bc927df)
- **Estado:** ✓ **FIXED 10/ene/2026****
- **Requiere:** Testing en múltiples navegadores

🟡 P1-2: Email composer bloqueado por overlay

- **Archivo:** `EmailComposer.jsx` línea 296
- **Causa:** z-index 30 del sidebar tapaba composer
- **Fix aplicado:** ✓ `zIndex: 9999` (commit 4634d47)
- **Estado:** ✓ **FIXED 10/ene/2026****

🟡 P1-3: Sin cambio de contraseña

- **Archivo:** `SecurityPage.jsx` línea 23
- **Causa:** TODO pendiente

- **Fix disponible:**  NO
- **Estado:**  **NO IMPLEMENTADO**

P2 (Media prioridad)

P2-1: Historial de conversaciones dummy

- **Archivo:** `HistoryPage.jsx` línea 7
- **Causa:** TODO pendiente, datos reales no cargan
- **Fix disponible:**  NO
- **Estado:**  **NO IMPLEMENTADO**

P2-2: Sin tests automatizados

- **Causa:** No hay suite de tests
- **Fix disponible:**  NO
- **Estado:**  **NO IMPLEMENTADO**
- **Riesgo:** Regresiones no detectadas

COBERTURA DE FUNCIONALIDADES

Matriz de Completitud

Módulo	Diseñado	Implementado	Funcional	Testeado	
Auth (Signup/Login)					
Perfil de usuario					
Chat básico					
Voice mode					
Proyectos (CRUD)					
Proyectos (compartir)					
Documentos de proyecto					
Email (leer)					
Email (enviar)					
Email (OAuth)					
Reuniones (crear)					
Reuniones (transcribir)					
Reuniones (minutas)					
Calendario (CRUD)					
Tareas					
Notificaciones					
Búsqueda global					

Leyenda:

-  Completo / Funcional
-  Parcial / Con bugs
-  No implementado / Roto

RECOMENDACIONES TÉCNICAS

Críticas (Hacer AHORA)

1. ****Ejecutar fixes SQL en Supabase****

```
```bash
FIX-PROJECTS-RLS-DEFINITIVO.sql
FIX-CALENDAR-RLS-URGENTE.sql
FIX-MEETINGS-RLS-DEFINITIVO.sql
````
```

****Tiempo:**** 5 minutos

****Impacto:**** Desbloquea colaboración y calendarios

2. ****Implementar refresh de OAuth tokens****

- Backend debe detectar token expirado
- Refrescar automáticamente
- ****Tiempo:**** 2-3 horas backend

3. ****Agregar tests E2E básicos****

- Cypress o Playwright
- Flujos críticos: signup, login, chat, email
- ****Tiempo:**** 1 semana

Alta Prioridad (Esta semana)

4. ****Cambio de contraseña****

- Implementar en `SecurityPage.jsx`
- Usar Supabase Auth API
- ****Tiempo:**** 2 horas

5. ****Historial de conversaciones real****

- Conectar `HistoryPage.jsx` con backend
- ****Tiempo:**** 4 horas

6. ****Notificaciones push****

- Email nuevo
- Tarea asignada
- Mensaje de chat

- ****Tiempo:**** 1 semana

Media Prioridad (Próximo sprint)

7. ****Búsqueda global****

- Buscar en chats, emails, proyectos
- ****Tiempo:**** 1 semana

8. ****Rate limiting****

- Protección contra abuse
- ****Tiempo:**** 1 día backend

9. ****Documentación técnica****

- API interna
- Guías de desarrollo
- ****Tiempo:**** 3 días

RIESGOS DE SEGURIDAD

Críticos

1. ****Sin verificación de email****

- Usuarios pueden registrarse con emails falsos
- ****Riesgo:**** Spam, abuse

2. ****Credenciales en .env****

- Archivo ` `.env` tiene SUPABASE_ANON_KEY hardcoded
- ****Riesgo:**** Exposición si se commitea por error

3. ****Sin 2FA****

- Cuentas vulnerables a phishing
- ****Riesgo:**** Takeover de cuentas

Medios

4. ****Sin rate limiting en frontend****

- Posible DOS desde cliente malicioso
- ****Riesgo:**** Saturación del backend

5. ****RLS policies incorrectas****

- Ya documentado (proyectos, calendario)
- ****Riesgo:**** Data leakage entre usuarios

1 5 MÉTRICAS DE CALIDAD

Código

- **Archivos:** 139
- **Líneas:** 33,808
- **Componentes React:** ~80
- **Servicios:** ~15
- **Errores de compilación:** 0 
- **TODOs pendientes:** 5+
- **Console logs de debugging:** 30+

Performance

- **Lighthouse Score (estimado):**
 - Performance: ~85/100
 - Accessibility: ~90/100
 - Best Practices: ~80/100
 - SEO: ~95/100

Deuda Técnica

- **Tests:** 0% cobertura 
- **Documentación:** ~10% 
- **Code reviews:** Sin proceso formal 
- **CI/CD:** Netlify automático 

1 6 CONCLUSIONES

Fortalezas

1. **Stack moderno y escalable** (React + Vite + Supabase)
2. **UI pulida y consistente** (Tailwind + Lucide)
3. **Backend robusto** (AL-E Core API)
4. **Deploy automático** (Netlify CI/CD)
5. **Sin errores de compilación**

Debilidades Críticas

1. **RLS policies rotas** (proyectos, calendario) → **BLOQUEANTE**
2. **OAuth tokens expiran** → **BLOQUEANTE**
3. **Sin tests** → **RIESGO ALTO**
4. **Funcionalidades incompletas** (tareas, notificaciones)
5. **Bugs de UX** (voice mode, email composer) → **PARCIALMENTE FIXED**

Estado General del Proyecto

****75% completitud funcional****

- ****Production-ready:****  ****NO**** (por bugs P0)
- ****Beta-ready:****  ****Sí**** (con workarounds)
- ****MVP:****  ****Sí**** (funcionalidad básica funciona)

ACCIÓN INMEDIATA REQUERIDA

Para desarrollador:

```bash

# 1. Ejecutar en Supabase SQL Editor (5 min)

FIX-PROJECTS-RLS-DEFINITIVO.sql

FIX-CALENDAR-RLS-URGENTE.sql

# 2. Verificar en app (10 min)

- Usuario 1: Crear proyecto
- Usuario 2: Ver proyecto compartido
- Usuario con ID aeafa6b7...: Ver evento del 6/ene

# 3. Implementar en backend (2-3 horas)

- Refresh automático de OAuth tokens

# 4. Testing manual (1 hora)

- Voice mode en Chrome, Safari, Firefox
- Email composer en múltiples resoluciones

```

Para jefe/gerente:

1. ****Aprobar tiempo para ejecutar fixes SQL**** (5 min)
2. ****Priorizar implementación de OAuth refresh**** (3 horas backend)
3. ****Aprobar presupuesto para tests E2E**** (1 semana dev)

ANEXOS

Archivos de Fixes Disponibles

```

-  FIX-PROJECTS-RLS-DEFINITIVO.sql (listo)
-  FIX-CALENDAR-RLS-URGENTE.sql (listo)

- ✓ FIX-MEETINGS-RLS-DEFINITIVO.sql (listo)
  - ✓ DEBUG-RLS-POLICIES-NOW.sql (diagnóstico)
- ```

### ### Commits Recientes (últimas 24h)

```

- ✓ eb71f15 - FIX AGENDA RLS (11/ene)
 - ✓ bc927df - FIX VOZ chunk capture (10/ene)
 - ✓ 4634d47 - FIX EMAIL z-index (10/ene)
 - ✓ bc014b9 - FIX CRÍTICO P0 (10/ene)
- ```

Documentación Técnica Existente

```

- 📄 FRONTEND-README.md
  - 📄 BACKEND-RESPONSE-FORMAT.md
  - 📄 CORE-TASKS-MAIL-ENDPOINTS.md
  - 📄 VOICE-IMPLEMENTATION.md
  - 📄 OAUTH-SETUP-GUIDE.md
- ```

\*\*Fin del Reporte de Auditoría\*\*

Generado automáticamente por GitHub Copilot  
Fecha: 11 de enero de 2026  
Versión: 1.0