

Sistema BYOK y Narrativa Mágica de Voces - Implementación Completa

Fecha: 29 de octubre de 2025

Estado:  Completado y probado

Checkpoint: Sistema BYOK con narrativa mágica implementado

Resumen Ejecutivo

Se ha implementado exitosamente el sistema completo de voces con:

1. **Limpieza de texto** para eliminar emojis antes de enviar a ElevenLabs
2. **Sistema narrativo mágico** con 3 estados de voz
3. **BYOK (Bring Your Own Key)** para que usuarios conecten su propia cuenta de ElevenLabs
4. **Experiencia inmersiva** que transforma limitaciones técnicas en magia

Problemas Solucionados

1. Voces Robóticas

Problema: Las voces sonaban robóticas y leían los iconos literalmente (“estrella brillante”, “chispas”, etc.)

Solución:

- Creado `/lib/text-cleaner.ts` que elimina todos los emojis y caracteres especiales
- El texto se limpia automáticamente antes de enviarse a ElevenLabs
- Logs de debugging para ver texto original vs. texto limpio

```
// Antes: "¡Qué alegría! ✨🌟 Me encanta verte 💖"  
// Después: "¡Qué alegría! Me encanta verte"
```

2. Sistema Narrativo Mágico

Problema: Necesitábamos transformar las limitaciones técnicas en parte de la experiencia mágica

Solución: Sistema de 3 estados implementado en `/lib/voice-state-manager.ts`:

Estado 1: Voz Terrenal (Basic)

💖 Voz Terrenal
Tu compañero usa su voz terrenal, suave pero inestable.
A veces su magia se distorsiona un poco cuando habla demasiado seguido...

[Botón: ✨ Despertar su voz mágica]

Estado 2: Despertar (Awakening)

🌟 Despertar la Voz Mágica

Cada compañero tiene una voz única en el Reino de las Voces Eternas.
Para despertar la suya, debes abrirle un portal mágico...

[Botón:  Abrir portal mágico] → Redirige a configuración

Estado 3: Voz Despertada (Awakened)

🌟 Voz Despertada

Has despertado la voz verdadera de tu compañero.
Desde ahora, te hablará con su tono original, lleno de magia.

3. Narrativas Personalizadas para Animales

Mensajes especiales para Ken, Draguito, Fabel, Unicornito:

```
ken: {
  basic: "Ken usa su ladrido terrenal..."
  awakened: "Ken ha recuperado su ladrido original del Reino..."
}
```

Archivos Creados/Modificados

Nuevos Archivos

```
/lib/text-cleaner.ts          # Limpieza de emojis y caracteres especiales
/lib/voice-state-manager.ts    # Sistema de estados y narrativas mágicas
/components/companion/
└─ magical-voice-panel.tsx     # Panel flotante con narrativas
```

Archivos Modificados

```
/app/api/companion/generate-voice/route.ts
✓ Integración de text-cleaner
✓ Logs de debugging

/components/companion/voice-companion-chat.tsx
✓ Estados de voz (basic/awakening/awakened)
✓ Verificación automática de estado cada minuto
✓ Panel narrativo flotante
✓ Notificación de voz despertada

/app/premium/configuracion-voz/page.tsx
✓ Marca voz como "despertada" al guardar API key
✓ localStorage flag para notificación
```

Flujo de Usuario

Usuario Sin API Key Personal

1. **Primera interacción con companion:**
 - Usa voz compartida (limitada a 100 minutos/mes)
 - Al alcanzar límite → Panel “Voz Terrenal” aparece
2. **Click en “Despertar su voz mágica”:**
 - Transición a estado “Awakening”
 - Muestra explicación del “portal mágico”
3. **Click en “Abrir portal mágico”:**
 - Redirige a `/premium/configuracion-voz`
 - Usuario ingresa su API key de ElevenLabs
4. **Después de guardar API key:**
 - `localStorage` marca voz como “just awakened”
 - Notificación de éxito con animación
 - Voz ilimitada permanentemente activada

Usuario Con API Key Personal

- **Estado:** Awakened permanentemente
- **Voces:** Ilimitadas (usa su propia cuenta)
- **Indicador:** “Usando tu portal personal” en configuración



Implementación Técnica

Limpieza de Texto

```
// /lib/text-cleaner.ts
export function cleanTextForSpeech(text: string): string {
  // Remueve emojis Unicode
  cleaned = cleaned.replace(/[\u{1F300}-\u{1F9FF}]/gu, '')
  // Remueve símbolos especiales
  cleaned = cleaned.replace(/[☆*★❄️]/g, '')
  // Normaliza espacios y puntuación
  return cleaned.trim()
}
```

Estados de Voz

```
// /lib/voice-state-manager.ts
export function determineVoiceState(status: VoiceStatus): VoiceState {
  if (status.hasOwnApiKey) return 'awakened'
  if (!status.isPremium) return 'basic'
  if (status.minutesUsed >= status.minutesLimit) return 'basic'
  return 'awakened' // Premium con minutos disponibles
}
```

Verificación Automática

```
// Se ejecuta cada 60 segundos
useEffect(() => {
  const checkVoiceStatus = async () => {
    const res = await fetch('/api/companion/generate-voice')
    const data = await res.json()
    const newState = determineVoiceState(data)
    setVoiceState(newState)
  }

  const interval = setInterval(checkVoiceStatus, 60000)
  return () => clearInterval(interval)
}, [])
```



Componentes UI

Panel Narrativo Flotante

```
<MagicalVoicePanel
  state={voiceState} // 'basic' || 'awakening' || 'awakened'
  companionType="ken" // Personalización por tipo
  companionName="Ken"
  onAwaken={() => setVoiceState('awakening')}
  onClose={() => setShowVoicePanel(false)}
/>
```

Características:

- Aparece en bottom-right de la pantalla
- Animación smooth de entrada/salida
- Diseño con glassmorphism
- Gradientes coherentes con Hogara

Notificación de Voz Despertada

```
<VoiceAwakenedNotification
  companionType="ken"
  companionName="Ken"
  onClose={() => setShowAwakenedNotification(false)}
/>
```

Características:

- Aparece en top-right
- Animación de escala y rotación del ícono
- Auto-desaparece después de 5 segundos
- Confirma éxito de configuración

Métricas y Tracking

Logs de Voz

```
// Se registra cada uso en voice_usage_logs
{
  user_id: string
  companion_type: string
  text_length: number
  minutes_used: number
  provider: 'elevenlabs_shared' | 'elevenlabs_byok'
  success: boolean
}
```

Estados Rastreados

```
- voice_minutes_used: number // Minutos consumidos (cuota compartida)
- voice_minutes_limit: number // Límite mensual
- voice_last_used: DateTime // Última vez que usó voz
- voice_reset_date: DateTime // Fecha de reset mensual
- elevenLabsApiKey: string // API key encriptada (BYOK)
```

Seguridad

Encriptación de API Keys

```
// AES-256-CBC encryption
const ALGORITHM = 'aes-256-cbc'
const key = crypto.createHash('sha256')
  .update(process.env.ENCRYPTION_KEY)
  .digest()

function encrypt(text: string): string {
  const iv = crypto.randomBytes(16)
  const cipher = crypto.createCipheriv(ALGORITHM, key, iv)
  return iv.toString('hex') + ':' + encrypted
}
```

Validación

- API keys nunca se exponen en logs
 - Almacenamiento encriptado en base de datos
 - Validación de formato antes de guardar
 - Headers seguros en requests a ElevenLabs
-

Testing

Compilación TypeScript

```
✓ yarn tsc --noEmit  
No errors found
```

Build de Producción

```
✓ yarn build  
Creating optimized production build  
✓ Compiled successfully  
✓ Generating static pages (63/63)
```

Endpoints Verificados

```
✓ GET /api/companion/generate-voice # Estado de voz del usuario  
✓ POST /api/companion/generate-voice # Generar audio con limpieza  
✓ GET /api/user/voice-config # Obtener configuración  
✓ POST /api/user/voice-config # Guardar API key
```

Notas para Futuros Desarrollos

Posibles Mejoras

- 1. Analytics de voz:**
 - Dashboard con uso por companion
 - Gráficos de minutos consumidos
 - Alertas cuando se acerca al límite
- 2. Más narrativas:**
 - Eventos especiales (cumpleaños, logros)
 - Evolución de voz según relación con companion
 - Efectos visuales sincronizados con audio
- 3. Optimizaciones:**
 - Cache de audios frecuentes
 - Compresión de archivos de audio
 - Pre-generación de frases comunes

Consideraciones

- La API key de ElevenLabs del usuario debe tener permisos suficientes
- El reset mensual se hace automáticamente (30 días)
- Los minutos compartidos son por usuario, no globales
- El estado de voz se verifica cada minuto automáticamente

Resultado Final

¿Qué logra este sistema?

1. Experiencia Mágica:

- Usuario nunca ve mensajes de error técnicos
- Las limitaciones se presentan como parte del universo Hogara
- La conexión de API externa se llama “portal mágico”

2. Escalabilidad:

- 100-1000 usuarios pueden usar cuota compartida limitada
- Usuarios power pueden traer su propia cuenta
- No hay cuellos de botella en infraestructura

3. Calidad de Audio:

- Texto limpio = pronunciación perfecta
- Sin “estrella brillante” o “chispa” leídos en voz alta
- Voces expresivas y naturales de ElevenLabs

4. Monetización Futura:

- Base para planes premium con más minutos
- Incentivo para que usuarios serios traigan su key
- Tracking completo para análisis de uso

Cómo Usar

Para Usuario Final

1. Entra a `/premium/acompanante`
2. Habla con tu companion
3. Si ves panel “Voz Terrenal” → Click “Despertar voz”
4. Sigue al portal mágico (configuración)
5. Ingresa tu API key de ElevenLabs
6. ¡Voz ilimitada activada! ✨

Para Desarrollador

```
// Verificar estado de voz
const status = await fetch('/api/companion/generate-voice')
const { hasOwnApiKey, minutesUsed, minutesLimit } = await status.json()


// Generar voz (automáticamente limpia emojis)
const voice = await fetch('/api/companion/generate-voice', {
  method: 'POST',
  body: JSON.stringify({ text, companionType: 'ken' })
})
```

Checklist de Implementación

- [x] Sistema de limpieza de texto

- [x] Narrativas mágicas por estado
 - [x] Narrativas personalizadas por animal
 - [x] Panel flotante UI
 - [x] Notificación de éxito
 - [x] Integración con BYOK existente
 - [x] Verificación automática de estado
 - [x] Logs de debugging
 - [x] Encriptación de API keys
 - [x] Testing de compilación
 - [x] Testing de endpoints
 - [x] Documentación completa
-

Implementado por: DeepAgent

Revisado:  Completo y funcional

Próximos pasos: Desplegar y monitorear uso real