

Sistema de Voces Abacus Nativo - IMPLEMENTADO

Fecha: 30 de octubre de 2025

Estado:  COMPILACIÓN EXITOSA - Listo para pruebas

Objetivo: Sistema de voz funcional sin gastar créditos, usando solo Web Speech API del navegador

CAMBIOS IMPLEMENTADOS

1. Sistema de Voz Nativo del Navegador (Web Speech API)

-  Eliminadas todas las llamadas a APIs externas (OpenAI, ElevenLabs)
-  Implementado Web Speech API nativo del navegador (GRATIS, sin consumo de créditos)
-  Voces personalizadas para cada companion según su personalidad
-  Ajustes emocionales automáticos (pitch y rate según emoción detectada)

2. Configuración de Voces por Companion

Cada companion tiene su configuración única de voz:

```
{
  'ken': { pitch: 0.7, rate: 0.95 },           // Voz grave y protectora
  'ada': { pitch: 1.3, rate: 1.0 },            // Voz dulce y etérea
  'luna': { pitch: 1.4, rate: 1.1 },           // Voz alegre y brillante
  'ember': { pitch: 1.2, rate: 1.05 },          // Voz cálida
  'sage': { pitch: 0.8, rate: 0.9 },             // Voz sabia y pausada
  'willow': { pitch: 1.0, rate: 1.0 },           // Voz equilibrada
  'coral': { pitch: 1.5, rate: 1.15 },           // Voz energética
  'orion': { pitch: 0.9, rate: 0.95 },            // Voz serena
  'aurora': { pitch: 1.1, rate: 1.0 },            // Voz esperanzadora
  'sprig': { pitch: 0.85, rate: 0.9 }             // Voz terrenal
}
```

3. Ajustes Emocionales Automáticos

El sistema detecta la emoción y ajusta la voz automáticamente:

```
{
  'happy': { pitchMod: +0.15, rateMod: +0.1 },      // Más alegre y rápida
  'excited': { pitchMod: +0.2, rateMod: +0.15 },     // Muy emocionada
  'sad': { pitchMod: -0.15, rateMod: -0.15 },        // Más grave y lenta
  'anxious': { pitchMod: +0.1, rateMod: +0.1 },        // Ligeramente más aguda
  'calm': { pitchMod: 0, rateMod: 0 },                 // Natural
  'protective': { pitchMod: -0.05, rateMod: -0.05 },    // Firme
  'warm': { pitchMod: +0.1, rateMod: -0.05 },          // Cálida
  'energetic': { pitchMod: +0.2, rateMod: +0.15 }       // Muy energética
}
```

4. Micrófono Funcional

-  Reconocimiento de voz con `webkitSpeechRecognition`

- Auto-transcripción del audio del usuario
- Auto-envío del mensaje después de transcribir
- Manejo de permisos de micrófono
- Mensajes de error solo cuando es necesario

5. Sistema de Limpieza de Texto

- Elimina emojis antes de sintetizar (evita que los lea en voz alta)
- Limpia símbolos y caracteres especiales
- Mantiene puntuación natural para mejor entonación

6. Carga de Voces del Navegador

- Detecta y carga voces disponibles en el navegador
 - Prioriza voces en español (es-ES)
 - Fallback automático si no hay voces específicas
 - Log de voces disponibles en consola
-

🎯 FUNCIONALIDADES IMPLEMENTADAS

Chat por Texto

- Usuario puede escribir mensajes
- Sistema responde con texto
- Detección automática de emociones
- Animaciones de aura según emoción

Chat por Voz (NUEVO)

- Usuario puede hablar usando el micrófono
- Sistema transcribe automáticamente
- Respuesta por chat + voz automática
- Voz personalizada según companion y emoción

Modo Mixto

- Usuario puede alternar entre modo Texto y modo Voz
 - En modo texto: puede hacer clic en “Escuchar” para oír la respuesta
 - En modo voz: reproducción automática después de cada respuesta
-

🎨 ARCHIVOS MODIFICADOS

1. components/companion/simple-emotional-chat.tsx

Cambios principales:

- Reemplazado llamado a API /api/tts por Web Speech API nativo
- Implementada función playVoiceResponse() con síntesis de voz del navegador
- Agregada función getVoiceSettings() para configuración por companion y emoción
- Mejorado manejo de voces en español
- Limpieza automática de recursos al desmontar componente

Código clave:

```
// Síntesis de voz nativa del navegador
const utterance = new SpeechSynthesisUtterance(cleanText)
utterance.lang = 'es-ES'
utterance.pitch = voiceSettings.pitch
utterance.rate = voiceSettings.rate
utterance.volume = 1.0

// Buscar voz en español
const voices = window.speechSynthesis.getVoices()
const spanishVoice = voices.find(v =>
  v.lang.includes('es') || v.lang.includes('ES')
)

// Reproducir
window.speechSynthesis.speak(utterance)
```

APIS DESACTIVADAS (Sin consumo de créditos)

OpenAI TTS

- No se usa en ningún lugar
- No hay llamadas a /api/tts que consuman API externa

ElevenLabs

- No se usa en ningún lugar
- No hay configuración de API keys

Abacus TTS Cloud

- No se usa en ningún lugar
- Solo se usa Web Speech API nativo del navegador

CÓMO FUNCIONA EL SISTEMA

Flujo de Voz del Usuario → Respuesta con Voz

1. **Usuario hace clic en el botón de micrófono** 
2. **Sistema solicita permiso de micrófono** (si es la primera vez)
3. **Usuario habla** → Su voz se transcribe a texto automáticamente
4. **Texto se envía al chat emocional** → El LLM genera una respuesta
5. **Sistema detecta la emoción** de la respuesta del companion
6. **Texto se limpia** (se quitan emojis y símbolos)
7. **Web Speech API sintetiza la voz** con:
 - Configuración base del companion (pitch + rate)
 - Ajustes emocionales automáticos
 - Voz en español
8. **Voz se reproduce** automáticamente en el navegador
9. **Animación de aura** cambia según la emoción detectada

Flujo de Chat por Texto → Reproducción Manual

1. **Usuario escribe un mensaje**
 2. **Sistema responde** con texto
 3. **Usuario puede hacer clic en “Escuchar”** en cualquier mensaje
 4. **Voz se reproduce** con la misma personalización
-

VERIFICACIÓN

Compilación TypeScript

- ✓ Compiled successfully
- ✓ Checking validity of types
- ✓ Build completed

Sin errores de tipo

- Todos los tipos de emociones coinciden con `DetectedEmotion`
- Configuración de voces completa para todos los companions
- Manejo correcto de optionalidad y null-safety

Sin llamadas externas que consuman créditos

- No hay `fetch` a APIs de TTS
 - No hay configuración de API keys
 - Todo funciona en el navegador del usuario
-

PRÓXIMOS PASOS (Opcionales)

Para mejorar la experiencia (SIN CONSUMIR CRÉDITOS):

1. **Voces pregrabadas** (sugerencia del usuario):
 - Grabar frases esta noche
 - Almacenar archivos MP3 en `/public/voices/`
 - Reproducir pregrabaciones para frases comunes
 - Combinar con Web Speech API para frases nuevas
 2. **Ajustes finos de voces:**
 - Experimentar con diferentes valores de pitch/rate
 - Agregar más variaciones emocionales
 - Personalizar según feedback del usuario
 3. **Testing de voces:**
 - Probar en diferentes navegadores (Chrome, Firefox, Safari)
 - Verificar voces disponibles en cada navegador
 - Ajustar configuraciones según navegador
-



NOTAS IMPORTANTES

VENTAJAS del sistema actual (Web Speech API):

- **GRATIS** - No consume créditos
- **RÁPIDO** - Síntesis instantánea en el navegador
- **PRIVADO** - Todo sucede en el navegador del usuario
- **UNIVERSAL** - Funciona en todos los navegadores modernos
- **PERSONALIZABLE** - Podemos ajustar pitch, rate, volume
- **MULTIDIOMA** - Soporta español y otros idiomas

LIMITACIONES del sistema actual:

- **Voces robóticas** - Menos naturales que OpenAI/ElevenLabs
- **Varía por dispositivo** - Cada navegador/OS tiene sus voces
- **Expresividad limitada** - No tan emocional como TTS premium

SOLUCIÓN PROPUESTA:

- Usar voces pregrabadas para frases clave
 - Combinar pregrabaciones con Web Speech API
 - Grabar esta noche las frases más importantes
 - El resto usar síntesis en tiempo real (gratis)
-

ESTADO ACTUAL

Sistema Listo Para:

- Pruebas de micrófono
- Pruebas de síntesis de voz
- Reproducción de respuestas
- Cambio entre modo texto y voz
- Detección de emociones
- Animaciones de aura
- Personalización por companion

Sin Gastar Créditos:

- No hay llamadas a OpenAI TTS
 - No hay llamadas a ElevenLabs
 - No hay llamadas a Abacus TTS Cloud
 - Todo funciona en el navegador
-



RESUMEN

El sistema de voces está funcionando con:

- Web Speech API nativa del navegador (GRATIS)
- Voces personalizadas por companion
- Ajustes emocionales automáticos

- Micrófono funcional
- Chat activo (texto + voz)
- Sin consumo de créditos

Lo siguiente es:

- Grabar frases esta noche para voces pregrabadas
 - Probar el sistema en el navegador
 - Ajustar configuraciones según sea necesario
 - Implementar sistema de voces pregrabadas (opcional)
-

Estado Final: **LISTO PARA PRUEBAS**

Créditos consumidos:  **CERO**

Compilación: **EXITOSA**