<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Amplificador de Voz</title>

<style>

\* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: -apple-system, BlinkMacSystemFont, 'Segoe UI', Roboto, sans-serif;

background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);

min-height: 100vh;

display: flex;

flex-direction: column;

align-items: center;

justify-content: center;

padding: 20px;

color: white;

}

.container {

background: rgba(255, 255, 255, 0.1);

backdrop-filter: blur(10px);

border-radius: 20px;

padding: 30px;

text-align: center;

box-shadow: 0 8px 32px rgba(0, 0, 0, 0.1);

border: 1px solid rgba(255, 255, 255, 0.2);

max-width: 400px;

width: 100%;

}

h1 {

font-size: 2.5em;

margin-bottom: 10px;

font-weight: 300;

}

.subtitle {

font-size: 1.1em;

opacity: 0.8;

margin-bottom: 30px;

}

.status {

font-size: 1.2em;

margin-bottom: 30px;

padding: 15px;

border-radius: 10px;

background: rgba(255, 255, 255, 0.1);

border: 1px solid rgba(255, 255, 255, 0.2);

}

.status.active {

background: rgba(76, 175, 80, 0.3);

border-color: rgba(76, 175, 80, 0.5);

}

.status.error {

background: rgba(244, 67, 54, 0.3);

border-color: rgba(244, 67, 54, 0.5);

}

.controls {

display: flex;

flex-direction: column;

gap: 20px;

margin-bottom: 30px;

}

.btn {

padding: 15px 30px;

font-size: 1.2em;

border: none;

border-radius: 50px;

cursor: pointer;

transition: all 0.3s ease;

background: rgba(255, 255, 255, 0.2);

color: white;

border: 2px solid rgba(255, 255, 255, 0.3);

}

.btn:hover {

background: rgba(255, 255, 255, 0.3);

transform: translateY(-2px);

box-shadow: 0 5px 15px rgba(0, 0, 0, 0.2);

}

.btn:active {

transform: translateY(0);

}

.btn.primary {

background: linear-gradient(45deg, #4CAF50, #45a049);

border-color: #4CAF50;

}

.btn.danger {

background: linear-gradient(45deg, #f44336, #d32f2f);

border-color: #f44336;

}

.volume-control {

margin-top: 20px;

}

.volume-control label {

display: block;

margin-bottom: 10px;

font-size: 1.1em;

}

.volume-slider {

width: 100%;

height: 8px;

border-radius: 5px;

background: rgba(255, 255, 255, 0.3);

outline: none;

-webkit-appearance: none;

}

.volume-slider::-webkit-slider-thumb {

-webkit-appearance: none;

appearance: none;

width: 20px;

height: 20px;

border-radius: 50%;

background: white;

cursor: pointer;

box-shadow: 0 2px 6px rgba(0, 0, 0, 0.3);

}

.volume-slider::-moz-range-thumb {

width: 20px;

height: 20px;

border-radius: 50%;

background: white;

cursor: pointer;

border: none;

box-shadow: 0 2px 6px rgba(0, 0, 0, 0.3);

}

.info {

font-size: 0.9em;

opacity: 0.8;

line-height: 1.5;

background: rgba(255, 255, 255, 0.1);

padding: 15px;

border-radius: 10px;

margin-top: 20px;

}

.microphone-icon {

font-size: 4em;

margin-bottom: 20px;

opacity: 0.7;

}

.pulse {

animation: pulse 2s infinite;

}

@keyframes pulse {

0% { transform: scale(1); opacity: 0.7; }

50% { transform: scale(1.1); opacity: 1; }

100% { transform: scale(1); opacity: 0.7; }

}

</style>

</head>

<body>

<div class="container">

<div class="microphone-icon" id="micIcon">🎤</div>

<h1>Amplificador de Voz</h1>

<p class="subtitle">Para clases sin quedar afónica</p>

<div class="status" id="status">

Presiona "Iniciar" para comenzar

</div>

<div class="controls">

<button class="btn primary" id="startBtn" onclick="startAmplifier()">

🎤 Iniciar Amplificador

</button>

<button class="btn danger" id="stopBtn" onclick="stopAmplifier()" style="display: none;">

⏹️ Detener

</button>

</div>

<div class="volume-control">

<label for="volumeSlider">🔊 Volumen de salida:</label>

<input type="range" id="volumeSlider" class="volume-slider" min="0" max="100" value="80">

<div style="margin-top: 10px; font-size: 1.1em;" id="volumeValue">80%</div>

</div>

<div class="info">

<strong>💡 Instrucciones:</strong><br>

• Conecta auriculares para evitar retroalimentación<br>

• Conecta un parlante externo si es necesario<br>

• Ajusta el volumen según tu necesidad<br>

• Mantén el teléfono cerca para mejor captura

</div>

</div>

<script>

let audioContext;

let microphone;

let gainNode;

let isAmplifying = false;

const statusEl = document.getElementById('status');

const startBtn = document.getElementById('startBtn');

const stopBtn = document.getElementById('stopBtn');

const volumeSlider = document.getElementById('volumeSlider');

const volumeValue = document.getElementById('volumeValue');

const micIcon = document.getElementById('micIcon');

// Actualizar display del volumen

volumeSlider.addEventListener('input', function() {

const volume = this.value;

volumeValue.textContent = volume + '%';

if (gainNode) {

gainNode.gain.value = volume / 100;

}

});

async function startAmplifier() {

try {

// Solicitar permisos de micrófono

const stream = await navigator.mediaDevices.getUserMedia({

audio: {

echoCancellation: false,

noiseSuppression: false,

autoGainControl: false

}

});

// Crear contexto de audio

audioContext = new (window.AudioContext || window.webkitAudioContext)();

microphone = audioContext.createMediaStreamSource(stream);

gainNode = audioContext.createGain();

// Configurar volumen inicial

gainNode.gain.value = volumeSlider.value / 100;

// Conectar micrófono -> ganancia -> salida

microphone.connect(gainNode);

gainNode.connect(audioContext.destination);

// Actualizar UI

isAmplifying = true;

statusEl.textContent = '🔴 AMPLIFICANDO - Hablando en vivo';

statusEl.className = 'status active';

startBtn.style.display = 'none';

stopBtn.style.display = 'block';

micIcon.className = 'pulse';

micIcon.textContent = '🔴';

} catch (error) {

console.error('Error al acceder al micrófono:', error);

statusEl.textContent = '❌ Error: No se pudo acceder al micrófono';

statusEl.className = 'status error';

}

}

function stopAmplifier() {

if (audioContext) {

audioContext.close();

audioContext = null;

}

isAmplifying = false;

statusEl.textContent = 'Amplificador detenido';

statusEl.className = 'status';

startBtn.style.display = 'block';

stopBtn.style.display = 'none';

micIcon.className = '';

micIcon.textContent = '🎤';

}

// Evitar que la pantalla se apague

let wakeLock = null;

async function requestWakeLock() {

try {

if ('wakeLock' in navigator) {

wakeLock = await navigator.wakeLock.request('screen');

}

} catch (err) {

console.log('Wake Lock no disponible:', err);

}

}

// Solicitar wake lock cuando se inicie

document.addEventListener('DOMContentLoaded', function() {

requestWakeLock();

});

// Manejar cuando la aplicación pierde el foco

document.addEventListener('visibilitychange', function() {

if (wakeLock !== null && document.visibilityState === 'visible') {

requestWakeLock();

}

});

</script>

</body>

</html>