<!DOCTYPE html>

<html lang="es">

<head>

<meta charset="UTF-8">

<title> Fiesta de Colores </title>

<style>

body {

font-family: 'Arial', sans-serif; background-color: #f0f0f0; display: flex;

justify-content: center; align-items: center; min-height: 100vh; margin: 0;

text-align: center;

}

.container {

background-color: white; border-radius: 15px; padding: 30px;

box-shadow: 0 10px 20px rgba(0,0,0,0.1); max-width: 600px;

width: 100%;

}

#party-title {

font-size: 2.5em; margin-bottom: 20px;

transition: color 0.5s ease;

}

.color-palette { display: flex;

justify-content: center; flex-wrap: wrap;

gap: 10px;

margin-bottom: 20px;

}

.color-btn { width: 50px; height: 50px;

border-radius: 50%; border: none;

cursor: pointer; transition: transform 0.2s;

}

.color-btn:hover { transform: scale(1.1);

}

.volume-control { margin: 10px 0;

}

</style>

</head>

<body>

<div class="container">

<h1 id="party-title">Bienvenidos a la Fiesta de Colores</h1>

<div class="volume-control">

<label for="volume-slider">Volumen: </label>

<input type="range" id="volume-slider" min="0" max="1" step="0.1" value="0.5">

</div>

<div>

<input type="color" id="new-color-input">

<button onclick="addNewColor()">Añadir Color</button>

<button onclick="resetParty()">Reiniciar Fiesta</button>

<button onclick="finishParty()">Finalizar Fiesta</button>

</div>

<div class="color-palette" id="color-palette"></div>

<div id="info-display"></div>

<div id="timer-display" class="inactive-warning"></div>

<div id="final-results" class="final-results"></div>

</div>

<script>

const initialColors = ['#FF6B6B', '#4ECDC4', '#45B7D1', '#FFA07A', '#98D8AA'];

const colorSounds = { '#FF6B6B':'./flauta.mpeg', // Rojo '#4ECDC4':'./lectro.mpeg', // Verde '#45B7D1':'./suspen.mpeg', // Azul '#FFA07A':'./canserv.mpeg', // Naranja '#98D8AA':'./yatra.mpeg' // Verde claro

};

let colors = [...initialColors]; const colorVotes = {};

let inactivityTimer;

let colorAudioMap = {};

function preloadSounds() {

// Precargar sonidos para una reproducción más rápida Object.entries(colorSounds).forEach(([color, soundUrl]) => {

const audio = new Audio(soundUrl); audio.preload = 'auto'; colorAudioMap[color] = audio;

});

// Configurar control de volumen

const volumeSlider = document.getElementById('volume-slider'); volumeSlider.addEventListener('input', (e) => {

Object.values(colorAudioMap).forEach(audio => { audio.volume = e.target.value;

});

});

}

function generateRandomColor() {

const letters = '0123456789ABCDEF'; let color = '#';

for (let i = 0; i < 6; i++) {

color += letters[Math.floor(Math.random() \* 16)];

}

return color;

}

function playSoundForColor(color) {

const specificSound = colorSounds[color]; if (specificSound) {

const audio = colorAudioMap[color]; if (audio) {

audio.currentTime = 0; // Reiniciar reproducción audio.play().catch(error => console.log('Error playing sound:', error));

}

} else {

// Si no tiene sonido específico, usar un sonido genérico

const genericSound = new [Audio('https://www.soundjay.com/button/sounds/button-08.mp3');](http://www.soundjay.com/button/sounds/button-08.mp3%27)%3B) genericSound.volume = document.getElementById('volume-slider').value; genericSound.play().catch(error => console.log('Error playing generic sound:', error));

}

}

function initializeParty() {

preloadSounds(); // Precargar sonidos al iniciar

const palette = document.getElementById('color-palette'); palette.innerHTML = '';

colors.forEach(color => {

const colorBtn = document.createElement('button'); colorBtn.className = 'color-btn'; colorBtn.style.backgroundColor = color; colorBtn.onclick = () => {

selectColor(color); playSoundForColor(color);

};

palette.appendChild(colorBtn); colorVotes[color] = 0;

});

// Limpiar resultados finales document.getElementById('final-results').innerHTML = '';

}

function selectColor(color) {

// Cambiar título

const title = document.getElementById('party-title'); title.style.color = color;

// Contar votos colorVotes[color]++; updateMostPopularColor();

// Iniciar temporizador de 15 segundos startColorSelectionTimer();

}

function addNewColor() {

// Generar un color aleatorio automáticamente const newColor = generateRandomColor();

if (newColor && !colors.includes(newColor)) { colors.push(newColor); colorVotes[newColor] = 0;

// Asignar sonido genérico para nuevos colores

colorSounds[newColor] = ['https://www.soundjay.com/button/sounds/button-11.mp3';](http://www.soundjay.com/button/sounds/button-11.mp3%27%3B)

const genericAudio = new Audio(colorSounds[newColor]); genericAudio.preload = 'auto';

colorAudioMap[newColor] = genericAudio;

const palette = document.getElementById('color-palette'); const colorBtn = document.createElement('button'); colorBtn.className = 'color-btn'; colorBtn.style.backgroundColor = newColor; colorBtn.onclick = () => {

selectColor(newColor); playSoundForColor(newColor);

};

palette.appendChild(colorBtn);

// Actualizar el input de color con el nuevo color generado document.getElementById('new-color-input').value = newColor;

}

}

function startColorSelectionTimer() {

const timerDisplay = document.getElementById('timer-display'); clearTimeout(inactivityTimer);

timerDisplay.innerHTML = '';

inactivityTimer = setTimeout(() => {

timerDisplay.innerHTML = 'Tiempo de selección agotado. Reiniciando la fiesta...'; setTimeout(resetParty, 15000);

}, 12000);

}

function updateMostPopularColor() {

const infoDisplay = document.getElementById('info-display'); const mostPopularColor = Object.keys(colorVotes).reduce((a, b) =>

colorVotes[a] > colorVotes[b] ? a : b

);

infoDisplay.innerHTML = `Color :

<span style="color:${mostPopularColor}">

${mostPopularColor} (${colorVotes[mostPopularColor]} votos)

</span>`;

}

function resetParty() { clearTimeout(inactivityTimer);

const timerDisplay = document.getElementById('timer-display'); timerDisplay.innerHTML = '';

Object.keys(colorVotes).forEach(color => { colorVotes[color] = 0;

});

const title = document.getElementById('party-title'); title.style.color = 'black';

const infoDisplay = document.getElementById('info-display'); infoDisplay.innerHTML = '';

const finalResults = document.getElementById('final-results'); finalResults.innerHTML = '';

colors = [...initialColors]; initializeParty();

}

function finishParty() { clearTimeout(inactivityTimer);

const timerDisplay = document.getElementById('timer-display'); timerDisplay.innerHTML = '';

const mostPopularColor = Object.keys(colorVotes).reduce((a, b) => colorVotes[a] > colorVotes[b] ? a : b

);

const finalResults = document.getElementById('final-results'); finalResults.innerHTML = `

<h3> Color Ganador de la Fiesta </h3>

<div class="color-result" style="background-color:${mostPopularColor}; color:white;">

${mostPopularColor} (${colorVotes[mostPopularColor]} votos)

</div>

`;

setTimeout(resetParty, 5000);

}

document.addEventListener('DOMContentLoaded', () => { initializeParty();

});

</script>

</body>

</html>