

## 1º CUATRIMESTRE

### ✦ ACTIVIDAD 10: PLATAFORMA STREAMING DE MÚSICA Y CANCIONES - PWD 7º 2º

(Presencial/Virtual Asincrónico - Actividades de Contingencia y Continuidad Pedagógica) 2025

📖 Materia: Diseño e Implementación de Sitios Web Dinámicos

📍 Escuela: EEST N.º 1 - “Eduardo Ader” Vicente López

📅 Curso: 7º Año 2º Grupo A

📅 Día y horario: Miércoles de 17:35 a 21:45 Hs

🔊 Modalidad: Presencial/Virtual Asincrónico

#### Introducción.

En la actualidad, las plataformas de streaming de música son una parte esencial de nuestras vidas cotidianas, ofrecen acceso instantáneo a millones de canciones, listas de reproducción personalizadas y recomendaciones basadas en gustos individuales. Spotify, una de las plataformas más populares, combina una interfaz intuitiva con potentes funcionalidades técnicas, como autenticación de usuarios, gestión de bibliotecas musicales y reproducción multimedia fluida.

En este trabajo, deberán desarrollar una plataforma de música inspirada o no en Spotify, podría ser también al estilo YouTube Music, Deezer, entre otras, aplicando los conocimientos adquiridos en programación frontend y backend al día de la fecha.

El objetivo principal es crear una sitio/aplicación/sistema web dinámico que incluya:

- ✓ Diseño atractivo con animaciones y efectos visuales.
- ✓ Autenticación de usuarios (registro e inicio de sesión).
- ✓ Gestión de canciones (alta, baja y modificación).
- ✓ Almacenamiento de datos (MySQL para el backend e IndexedDB para el frontend).
- ✓ Diseño responsivo (adaptable a móviles, tablets y PC).
- ✓ Reproducción de audio con controles básicos (play, pause, volumen).

Además, se utilizarán librerías y frameworks modernos, como p5.js (para animaciones interactivas) y p5.sound.js (para el manejo de audio), entre otras.

#### ***Al finalizar, deberán:***

1. Entregar los archivos en la plataforma de la escuela.

2. Subir el código a GitHub con un repositorio bien documentado.
3. Desplegar la aplicación en Vercel (u otro servicio similar) para demostrar su funcionamiento en tiempo real.

### **Estructura del Proyecto.**

#### ***Interfaz Atractiva con Animaciones y Efectos.***

La plataforma debe contar con un diseño moderno y atractivo, incluyendo:

- ✓ Animaciones CSS/JS (transiciones suaves, hover effects, carga de elementos).
- ✓ Uso de p5.js para efectos visuales interactivos (ej: visualizador de ondas de sonido).
- ✓ Paleta de colores coherente (oscuro como Spotify o personalizado).
- ✓ Navegación intuitiva (barra lateral, menú de usuario, buscador).

#### ***Autenticación de Usuarios (Login/Registro).***

- ✓ Formularios de registro e inicio de sesión con validación.
- ✓ Encriptación de contraseñas (usando bcrypt o similar en el backend).
- ✓ Manejo de sesiones (JWT o cookies).
- ✓ Protección de rutas (solo usuarios registrados pueden acceder a ciertas funciones).

#### ***Gestión de Canciones (CRUD).***

- ✓ Alta de canciones: Formulario para subir archivos MP3 + metadatos (título, artista, álbum).
- ✓ Baja de canciones: Opción para eliminar canciones (solo administradores o usuarios propietarios).
- ✓ Modificación: Editar información de canciones (ej: corregir nombre de artista).

#### ***Almacenamiento de Datos.***

Backend (MySQL):

- ✓ Tablas: `usuarios`, `canciones`, `playlists`.
- ✓ Consultas SQL para gestionar datos.

Frontend (IndexedDB):

- ✓ Almacenamiento en caché para mejorar rendimiento (ej: guardar canciones escuchadas recientemente).

#### ***Diseño Responsivo.***

- ✓ Mobile-first approach o uso de media queries.
- ✓ Pruebas en diferentes dispositivos (Chrome DevTools, dispositivos reales).

- ✓ Flexbox/Grid CSS para layouts adaptables.

### ***Reproducción de Contenido Multimedia.***

Reproductor de audio con:

- ✓ Play/Pause.
- ✓ Barra de progreso.
- ✓ Control de volumen.

Uso de p5.sound.js para manipulación avanzada de audio (ecualizador, efectos).

Carga dinámica de canciones desde la base de datos.

### ***Tecnologías y Librerías Requeridas.***

Frontend:

- ✓ HTML5, CSS3, JavaScript (ES6+).
- ✓ p5.js + p5.sound.js (para animaciones y audio).
- ✓ Opcional: React.js o Vue.js para mejor estructura.

Backend:

- ✓ PHP(o Node.js + Express si prefieren).
- ✓ MySQL (o MongoDB como alternativa).

Despliegue:

- ✓ GitHub (control de versiones).
- ✓ Vercel / Netlify (frontend).
- ✓ Render / Railway (backend).

### **Formato de Entrega.**

- ✓ Repositorio GitHub bien documentado (README.md con explicación del proyecto, capturas, instrucciones de instalación).
- ✓ Demo en vivo (Vercel, Netlify, etc.).
- ✓ Código fuente entregado en la plataforma de la escuela.

### **Requisitos Mínimos.**

- ✓ Interfaz visual atractiva con animaciones.
- ✓ Login/registro funcional.
- ✓ CRUD de canciones.

- ✓ Base de datos MySQL + IndexedDB.
- ✓ Diseño responsivo.
- ✓ Reproductor funcional.

#### **Extras (Puntos Adicionales).**

- ★ Sistema de playlists.
- ★ Búsqueda inteligente de canciones.
- ★ Compartir canciones en redes sociales.
- ★ Modo oscuro/claro.

#### **Fechas y Evaluación.**

- ✓ Fecha de entrega: 28/05/2025
- ✓ Presentación en clase: Demostración en vivo + explicación técnica (siguiente clase).

#### **Criterios de evaluación:**

- ✓ Funcionalidad (50%).
- ✓ Diseño y usabilidad (30%).
- ✓ Documentación y despliegue (20%).