



Desarrollo de un Reproductor Multimedia con React y Tailwind CSS

Objetivo:

El objetivo de este ejercicio es crear un reproductor multimedia reutilizable utilizando React y estilizarlo con Tailwind CSS. Este reproductor debe ser capaz de manejar tanto archivos de audio como de video, ofreciendo controles básicos para reproducir, pausar, detener el contenido multimedia, y ajustar el volumen.

Configuración Inicial:

- Crea un nuevo proyecto React utilizando **create-react-app**.
- Instala Tailwind CSS en tu proyecto siguiendo la documentación oficial.

Creación del Componente MediaPlayer:

- Desarrolla un componente React llamado **MediaPlayer**.
- Este componente debe aceptar props que indiquen el tipo de medio ("**audio**" o "**video**") y la fuente del archivo (**src**).

Implementación de Controles Multimedia:

- Añade controles para reproducir, pausar, y detener el medio. Estos controles no deben usar los controles predeterminados del navegador, sino ser implementados como botones personalizados.
- Implementa un control deslizante para ajustar el volumen del reproductor multimedia.

Integración con Tailwind CSS:

- Utiliza Tailwind CSS para estilizar tu componente **MediaPlayer**. Asegúrate de que el reproductor sea visualmente atractivo y responsive.
- Personaliza los botones y el control deslizante para que se integren armoniosamente con el diseño de tu reproductor.

Pruebas y Validación:

- Prueba tu reproductor con diferentes archivos de audio y video para asegurarte de que funciona correctamente.
- Verifica que el reproductor sea responsive y que mantenga su funcionalidad y estética en diferentes tamaños de pantalla.

Entregables:

- Código fuente del proyecto que incluya el componente **MediaPlayer** y cualquier otro componente auxiliar utilizado.
- Instrucciones sobre cómo ejecutar tu proyecto, incluyendo la instalación de dependencias y el comando para iniciar la aplicación.
- (Opcional) Un breve documento o archivo README que describa tu enfoque para diseñar y construir el reproductor multimedia.



Criterios de Evaluación:

- **Funcionalidad:** El reproductor debe ser capaz de reproducir, pausar, detener medios y ajustar el volumen.
- **Reutilización y Estructura del Código:** El componente debe ser reutilizable y estar bien estructurado.
- **Estilo y Responsiveness:** El reproductor debe estar estéticamente bien diseñado y ser responsive.
- **Implementación de Tailwind CSS:** Uso efectivo de Tailwind CSS para estilizar el componente.

Creación del Componente MediaPlayer

Primero, vamos a crear el archivo `MediaPlayer.js` en tu carpeta de componentes. *Este componente aceptará **props** para `type` (**audio o video**) y `src` (la fuente del archivo multimedia).*

```
// MediaPlayer.js
import React, { useRef, useState } from 'react';

const MediaPlayer = ({ type, src }) => {
  const mediaRef = useRef(null); // Ref para acceder al elemento de audio o video
  const [volume, setVolume] = useState(1); // Estado para manejar el volumen

  // Función para reproducir el medio
  const play = () => mediaRef.current.play();

  // Función para pausar el medio
  const pause = () => mediaRef.current.pause();

  // Función para detener el medio (pausar y reiniciar tiempo)
  const stop = () => {
    mediaRef.current.pause();
    mediaRef.current.currentTime = 0;
  };

  // Función para cambiar el volumen
  const changeVolume = (e) => {
    const newVolume = e.target.value;
    setVolume(newVolume);
    mediaRef.current.volume = newVolume;
  };

  // Renderizado condicional basado en el tipo de medio
  return (
```



```

<div className="p-4">
  {type === 'video' ? (
    <video ref={mediaRef} className="w-full max-w-xl mx-auto"
controls={false}>
      <source src={src} type="video/mp4" />
      Your browser does not support the video tag.
    </video>
  ) : (
    <audio ref={mediaRef} controls={false}>
      <source src={src} type="audio/mp3" />
      Your browser does not support the audio element.
    </audio>
  )}

  {/* Controles para reproducir, pausar, y detener el medio */}
  <div className="flex justify-center space-x-2 my-2">
    <button onClick={play} className="px-4 py-2 bg-blue-500 text-
white rounded">Play</button>
    <button onClick={pause} className="px-4 py-2 bg-yellow-500 text-
white rounded">Pause</button>
    <button onClick={stop} className="px-4 py-2 bg-red-500 text-white
rounded">Stop</button>
  </div>

  {/* Control deslizante para el volumen */}
  <div className="flex justify-center items-center space-x-2">
    <label htmlFor="volume" className="text-white">Volume:</label>
    <input id="volume" type="range" min="0" max="1" step="0.1"
value={volume} onChange={changeVolume} className="w-full max-w-xs" />
  </div>
</div>
);
};

export default MediaPlayer;

```



Uso del Componente MediaPlayer

A continuación, puedes usar el componente **MediaPlayer** en tu aplicación. Por ejemplo, en **App.js**, importa y utiliza el componente para reproducir un archivo de audio y otro de video.

```
// App.js
import React from 'react';
import MediaPlayer from '../components/MediaPlayer'; // Asegúrate de
ajustar la ruta de importación según tu estructura de carpetas

function App() {
  return (
    <div className="App">
      <h2 className="text-center text-2xl font-bold my-4">Reproductor
Multimedia con React y Tailwind</h2>
      <MediaPlayer type="video" src="/path/to/your-video.mp4" />
      <MediaPlayer type="audio" src="/path/to/your-audio.mp3" />
    </div>
  );
}

export default App;
```

Al seleccionar archivos de audio y video para utilizar en tu reproductor multimedia, es crucial respetar las licencias bajo las cuales se distribuyen estos archivos. Muchos creadores permiten el uso de su trabajo bajo licencias Creative Commons, las cuales pueden requerir atribución, limitar el uso a contextos no comerciales, prohibir obras derivadas o permitir el uso con pocas restricciones, dependiendo de la licencia específica aplicada.

Consideraciones:

- **Busca archivos con licencias apropiadas:** Utiliza recursos que estén explícitamente marcados para reutilización. Sitios como [Free Music Archive](#) para audio y [Pixabay](#) para videos ofrecen contenido que a menudo se puede usar libremente, pero siempre verifica la licencia específica de cada archivo.
- **Respetar los términos de la licencia:** Si un archivo requiere atribución, asegúrate de incluir un crédito adecuado en tu proyecto. Si la licencia prohíbe el uso comercial, no utilices el archivo en proyectos que se beneficien económicamente.
- **Incluye atribuciones cuando sea necesario:** Cuando uses contenido bajo una licencia que requiera atribución, añade una sección de créditos en tu aplicación o sitio web que liste adecuadamente las fuentes y los autores de los archivos multimedia.



Ejemplo de Atribución:

Si tu aplicación utiliza un archivo de audio o video que requiere atribución, podrías incluir algo como lo siguiente en una sección de créditos de tu aplicación o sitio web:

`"Sunset Journey" by John Doe, utilizado bajo CC BY-SA. Disponible en:
[URL de origen del archivo]`