10-12-2024

DAVID ALEGRE LÓPEZ DE LA FRANCA

ESCUELA SUPERIOR DE INFORMÁTICA, UCLM

Memoria Laboratorio

Reproductor de vídeo

Contenido

[1. Descripción General 3](#_Toc184753156)

[2. Requisitos de la Aplicación 3](#_Toc184753157)

[3. Decisiones y Justificación de Diseño e Implementación 4](#_Toc184753158)

[4. Conclusiones 5](#_Toc184753159)

[5. Bibliografía 5](#_Toc184753160)

[6. Manual de Usuario (Guía de Inicio) 6](#_Toc184753161)

# 1. Descripción General

El proyecto consiste en el desarrollo de una aplicación de escritorio capaz de reproducir contenido multimedia (audio y video), mostrando metadatos asociados a los archivos y permitiendo al usuario ajustar ciertas propiedades visuales del video (brillo, contraste, saturación), además de ofrecer funciones básicas de control (play, pause, stop, avance y retroceso). Está implementado en Python, haciendo uso de librerías enfocadas en la reproducción multimedia (VLC), el análisis de metadatos (mutagen, pymediainfo), y una interfaz gráfica (tkinter).

El objetivo principal es proporcionar una interfaz sencilla, con una experiencia enriquecida por la visualización de metadatos y controles de efectos de video, todo en un entorno amigable. A nivel visual, la aplicación ofrece dos temáticas de color (azul y naranja) seleccionables durante la ejecución, así como un modo pantalla completa y un "mini player".

# 2. Requisitos de la Aplicación

Requisitos funcionales:

* Reproducir archivos de audio (mp3, entre otros) y video (mp4, avi, mkv, etc.).
* Mostrar metadatos del archivo reproducido:
  + Para audio: título, artista, álbum, género y duración.
  + Para video: formato, resolución, duración, codec de audio y número de canales.
* Controles de reproducción:
  + Play/Pause
  + Stop
  + Avance y retroceso en intervalos de 5 segundos
* Ajustes de video:
  + Brillo
  + Contraste
  + Saturación
* Cambiar entre fondo azul y naranja
* Modo pantalla completa y modo mini player.
* Barra de progreso para saltar a un punto específico en la reproducción.
* Carga de archivos mediante un cuadro de diálogo.

Requisitos no funcionales:

* Interfaz intuitiva, con diseño responsive.
* Uso de bibliotecas Python estándar y librerías especializadas (python-vlc, mutagen, pymediainfo, Pillow).
* Portabilidad: El ejecutable final se genera con PyInstaller para facilitar su distribución en Windows.
* Rendimiento aceptable para reproducir archivos multimedia sin demoras significativas.

Dependencias externas:

* VLC instalado en el sistema (o librerías libvlc incluidas).
* MediaInfo instalado en el sistema para análisis de metadatos de video.
* Python y pip para el entorno de desarrollo, no requeridos para el usuario final si se distribuye un ejecutable.

# 3. Decisiones y Justificación de Diseño e Implementación

Elección del lenguaje y librerías:

* Se eligió Python por su rapidez de desarrollo y la amplia disponibilidad de librerías para multimedia y GUI.
* tkinter se utilizó para la interfaz gráfica por ser parte de la biblioteca estándar de Python y fácil de integrar.
* python-vlc fue elegido para la reproducción multimedia, ya que ofrece control avanzado sobre audio y video.
* mutagen para analizar metadatos de audio (ID3 en MP3), y pymediainfo para extraer metadatos de video, dando información valiosa al usuario.
* Pillow para el manejo y redimensionamiento de íconos e imágenes.

Arquitectura:

* Una clase MediaPlayer encapsula la lógica principal:
  + Configura la ventana principal (root) con tkinter.
  + Crea un panel para el video, frames para metadatos, efectos, barra de progreso, y botones.
  + El diseño es modular, separando análisis de metadatos en funciones analyze\_mp3 y analyze\_video.

Interfaz y usabilidad:

* Botones con íconos intuitivos.
* Barra de volumen y sliders de efectos de imagen fácilmente accesibles.
* Menús emergentes para cambiar entre miniplayer, pantalla completa, y fondo.
* Ajuste dinámico de fondo (azul/naranja) para preferencia del usuario.
* Se evita complejidad excesiva: el usuario se limita a las funciones esenciales de un reproductor.

Empaquetado:

* Uso de PyInstaller para generar un .exe autónomo, facilitando su distribución sin necesidad de instalar Python o las dependencias Python en la máquina del usuario.
* El usuario solo debe tener VLC y MediaInfo previamente instalados o gestionados a nivel del sistema.

# 4. Conclusiones

El proyecto logra un reproductor multimedia funcional, con una interfaz sencilla que a la vez ofrece varias características atractivas (ajustes de video, metadatos, modo fullscreen, fondo dinámico). La decisión de usar Python y sus librerías especializadas facilitó el desarrollo rápido y flexible.

Si bien la necesidad de VLC y MediaInfo instalados es una limitación, es posible empaquetarlos con instaladores externos usando Inno Setup, brindando una experiencia más integrada para el usuario final.

En futuras mejoras, se podría:

* Agregar soporte para listas de reproducción.
* Integrar más formatos de metadatos.
* Incluir traducciones o soporte multi-idioma.
* Añadir preajustes de ecualización o efectos de audio.

Además añadir que personalmente lo que más me costó fue implementar bien la barra de progreso, ya que al principio simplemente no hacía nada y cuando conseguí que la barra de progreso se actualizara a medida que el vídeo iba avanzando, cada vez que ésta se actualizaba causaba que el vídeo se quedara “pillado”.

Otro pequeño problema fue la integración de la pantalla flotante, que se veía en negro aunque el sonido del vídeo si que existía, aunque este fue más fácil de solucionar.

# 5. Bibliografía

* Python oficial: <https://www.python.org/>
* Documentación de tkinter: <https://docs.python.org/3/library/tkinter.html>
* PyInstaller: <https://www.pyinstaller.org/>
* python-vlc: https://wiki.videolan.org/PythonBinding/
* mutagen: <https://mutagen.readthedocs.io/>
* pymediainfo: <https://pymediainfo.readthedocs.io/>
* Pillow: <https://pillow.readthedocs.io/>
* MediaInfo: https://mediaarea.net/en/MediaInfo

# 6. Manual de Usuario (Guía de Inicio)

Instalación:

1. Instalar VLC en Windows desde https://www.videolan.org/vlc/.
2. Instalar MediaInfo desde https://mediaarea.net/en/MediaInfo.
3. Obtener el ejecutable main.exe (generado con PyInstaller) y colocarlo en una carpeta de su preferencia.

Primeros pasos:

1. Hacer doble clic en main.exe para iniciar el reproductor.
2. Desde la interfaz principal, haga clic en el ícono de "cargar archivo" (con forma de carpeta) para abrir el cuadro de diálogo y seleccionar un archivo multimedia.
3. Una vez cargado, la reproducción iniciará automáticamente. Se mostrarán los metadatos detectados en el panel superior.
4. Use los botones de la barra inferior para:
   * Play/Pause: iniciar o pausar la reproducción.
   * Stop: detener el video.
   * Avance o retroceso 5s: moverse rápidamente por el contenido.
   * Barra de progreso: arrastre el deslizador para saltar a una posición específica.
   * Slider de volumen: ajuste el volumen.
   * Slider de brillo, contraste, saturación: en la parte superior (efectos), ajuste la imagen durante la reproducción.
5. Para cambiar de fondo (azul/naranja), haga clic en el botón correspondiente (el ícono cambiará).
6. El modo pantalla completa se activa con el botón de pantalla completa. Presione el mismo botón o Esc para salir.
7. El modo mini player se activa con el botón correspondiente, abriendo una ventana más pequeña. Para volver, use el botón "Volver" en la ventana mini.

Con estos pasos, el usuario puede reproducir, ajustar y disfrutar de contenido multimedia con metadatos y controles simples.