

## **1. Objetivo**

Presentar el funcionamiento, instalación y ejecución del Centro Multimedia Ruko's Multimedia.

## **2. Introducción**

En el presente trabajo se implementará un sistema embebido por medio de un centro de entretenimiento para la reproducción de películas, videos, música y fotografías, entendiendo que un centro multimedia tiene el poder de reproducir cualquier película o serie, escuchar música, ver fotos y mucho más; todo ello de la forma más sencilla posible.

Existen diferentes dispositivos en el cual se pueden programar para que funcionen como un centro multimedia en este caso se utilizara la microcomputadora Raspberry Pi 4 en donde se le instalara el sistema operativo Raspbian Lite (32 bits) para presentar diversas plataformas de streaming, así como al agregarle un dispositivo de USB pueda reproducir música, videos y una galería de fotos, todo esto programado en el lenguaje Python.

## **3. Antecedentes**

Raspberry Pi es un pequeño computador que corre un sistema operativo linux capaz de permitirle a las personas de todas las edades explorar la computación y aprender a programar lenguajes como Scratch y Python. Es capaz de hacer la mayoría de las tareas típicas de un computador de escritorio, desde navegar en internet, reproducir videos en alta resolución, manipular documentos de ofimática, hasta reproducir juegos.

Raspberry Pi OS es el sistema operativo antes conocido como Raspbian. Su nombre deja muy poco a la imaginación en lo que respecta a su procedencia y los dispositivos a los que va destinado, aunque también cuenta con una compilación dirigida a máquinas x86 de 32-bit.

## **4. Materiales.**

- Microprocesador Raspberry Pi 3 o superior
- Fuente de alimentación micro sub de 5[V] y 2[A]
- Tarjeta microSD 16GB o más
- USB de 4GB o más
- Monitor de entrada HDMI
- Sistema Operativo Raspberry Pi OS
- Equipo de cómputo de escritorio o laptop

## **5. Instrucciones**

### **5.1 Como empezar**

Primero se debe hacer las configuraciones pertinentes con el sistema operativo necesario, siga los siguientes pasos:

#### **5.1.1 Configuración de tarjeta SD y carga de imagen**

Inserte la tarjeta microSD en el adaptador que viene en su caja de compra y este a su vez en su equipo de cómputo de escritorio o laptop, una vez insertada previamente debería tener instalado en este equipo el Install Raspberry Pi OS using Raspberry Pi Imager si no fuera el caso por favor

diríjase a la siguiente página web <https://www.raspberrypi.com/software/> ahí podrá descargarlo con fidelidad y confianza.

Si ya lo tiene, empiece por formatear su microSD en el icono de Operating System CHOOSE OS seleccione Erase (format card as FAT32), posteriormente pase al icono de Storage CHOOSE STORAGE seleccione el dispositivo a formatear y por último el icono de WRITE se breve el formateo.

Una vez realizado lo anterior, proceda a ocupar los mismos iconos, pero ahora en el primer icono CHOOSE OS seleccione Raspberry Pi OS (other) el cual le desplegará otra ventana y seleccione el Sistema operativo que desee, en este caso se ocupó Raspbian Lite (32-bit), seleccione el dispositivo formateado y después WRITE así obtendrá la imagen de su microSD para poder trabajar.

### **5.1.2. Conexión de componentes**

Una vez teniendo la tarjeta SD configurada, proceda a conectar a la tarjeta Raspberry Pi el cable HDMI, los periféricos (mouse y teclado), tarjeta SD en su respectiva ranura. Conecte el otro extremo del cable HDMI al monitor y la fuente de alimentación a la corriente y la Raspberry Pi. Debería de visualizar después de unos instantes configuraciones del sistema, elija los más convenientes (idioma del sistema, zona horaria, etc....)

### **5.1.3. Instalación y actualización de paquetes**

Para su fácil instalación se tiene el archivo “install.sh” donde se encuentra los comandos de instalación necesarios para que funcione el programa, a grandes rasgos se instalan:

- Módulos necesarios de Python para el programa.
- Lightdm y entorno de escritorio
- Chromium
- VLC para la Raspberry (No Python)
- Widevine (Para el DRM y acceder el contenido de streaming)

### **5.1.4. Programa**

Ya una vez instalado lo necesario es importante copiar el programa para su funcionamiento, la manera en que se descargue a la Raspberry es trivial lo importante es definir donde se ubicará el archivo “RukoStreaming.py” y su carpeta “src”, nosotros recomendamos en /home/<usuario>/

### **5.1.5. Configuración**

Para su fácil configuración se tiene el archivo “config.txt” donde detalla los pasos a seguir para la configuración “manual” de la Raspberry para que ejecute y muestre solo el programa al prenderse, a grandes rasgos las configuraciones son:

- Habilitar login sin contraseña.
- Deshabilitar mensaje de usb insertada.
- Deshabilitar la barra de tareas.
- Configuración para correr el programa al prender la Raspberry.
- Reiniciar Raspberry.

### **5.1.6. Encienda su centro multimedia Ruko's Multimedia**

1. Conectar pantalla, mouse, teclado y cable ethernet (este se recomienda para tener mejor velocidad de internet).
2. Conectar a la fuente de poder para encender la Raspberry, esperar que encienda y cargue el programa.
3. Para acceder a servicio de video en Streaming, presione algún icono (Netflix, HBO, Blim) y será redireccionado al sitio web, ingrese sus credenciales y disfrute.
4. Para acceder a servicio de música en Streaming, presione algún icono (Spotify, Deezer, Youtube Music) y será redireccionado al sitio web, ingrese sus credenciales y disfrute.
5. Para reproducir algún contenido del medio usb, inserte la usb con el contenido a reproducir, es importante mencionar que solo acepta archivos .mp3 .mp4 y .jpg.

### **6. Manual de seguridad**

IMPORTANTE: POR FAVOR GUARDE ESTA INFORMACIÓN PARA REFERENCIAS FUTURAS.

#### **ADVERTENCIAS**

- Este producto solo debe conectarse a una fuente de alimentación externa de 5 VCC y una corriente mínima de 2Amp. Todas las fuentes de alimentación utilizadas con la Raspberry Pi deben cumplir con las normas y reglamentos aplicables correspondientes en el país en que se van a usar.
- Este producto debe ser operado en un entorno con buena ventilación, si se va utilizar en una caja esta no deberá ser cubierta.
- Este producto se debe poner en una superficie estable, plana y no conductiva y no deberá estar en contacto con componentes conductivos.

#### **INSTRUCCIONES PARA UN USO SEGURO**

Para evitar el mal funcionamiento o daño de su Centro multimedia Ruko's siga las siguientes recomendaciones:

- No lo exponga al agua ni a la humedad ni la coloque en una superficie conductora mientras está en funcionamiento.
- No la exponga a ninguna fuente de calor, es decir; la Raspberry Pi está diseñada para el funcionamiento fiable a temperatura normal ambiental normal.
- Tenga cuidado al manipularla para evitar daños mecánicos o eléctricos en la placa del circuito impreso y los conectores.
- Evite manipular la placa del circuito impreso si esta encendida. Solo toque los bordes para minimizar el riesgo de daños por descargas electrostáticas.
- La Raspberry Pi no está diseñada para ser alimentada mediante un puerto USB desde otro equipo conectado, e intentar conectarla de esa manera puede causar su mal funcionamiento.

## 7. Desarrollo

El código fue desarrollado en el lenguaje de programación Python el cual se podrá observar en el repositorio de GitHub en la siguiente dirección <https://github.com/Y4smin/Ruko-s-Multimedia>, donde como primer acercamiento tenemos que se debe de montar en una microSD el sistema operativo a utilizar, para posteriormente cargar el archivo denominado RukoStreaming.py que también puede encontrarlo en el repositorio antes mencionado, en el video.

Al leer el código de “RukoStreaming.py” podemos encontrar que el flujo del programa consta de:

1. **Importación de módulos:** El código comienza importando los módulos necesarios para el funcionamiento del programa, como Tkinter, filedialog, webbrowser, vlc, os, Image, ImageTk, customtkinter, pyudev, time y subprocess.
2. **Variables y configuraciones iniciales:** Se definen algunas variables y configuraciones iniciales, como las variables "current\_dir" y "script\_dir" para obtener el directorio actual y el directorio del script, respectivamente, para que a la hora de ejecución el programa pueda localizar donde están sus recursos gráficos.
3. **device\_event\_handler:** Esta función es un controlador de eventos para los eventos de conexión y desconexión de dispositivos USB. Dependiendo del evento se va a mostrar o ocultar un título en la interfaz gráfica. En el caso de que se detecte una USB se llama a la función usbMedia que es la que maneja la lógica de la USB.
4. **buscar\_ruta\_usb:** Esta función busca la ruta de un dispositivo USB conectado en la Raspberry Pi. Utiliza la biblioteca pyudev para detectar dispositivos USB y devuelve la ruta del directorio del dispositivo si se encuentra uno. (Se dejó el nombre fijo ya que el intento de que maneje cualquier usb no funciona de último momento).
5. **detectar\_archivos\_mp4\_mp3\_jpg:** Esta función recibe una ruta como argumento y busca archivos con extensiones .mp4, .mp3 y .jpg en esa ruta. Devuelve tres listas con los nombres de los archivos encontrados para cada extensión.
6. **reproducir\_jpg:** Esta función reproduce un archivo de imagen .jpg utilizando la biblioteca vlc. Crea una instancia de vlc, carga el archivo de imagen y lo reproduce.
7. **reproducir\_mp3:** Esta función reproduce un archivo de audio .mp3 utilizando la biblioteca vlc. Crea una instancia de vlc, carga el archivo de audio y lo reproduce.
8. **reproducir\_mp4:** Esta función reproduce un archivo de video .mp4 utilizando la biblioteca vlc. Crea una instancia de vlc, carga el archivo de video y lo reproduce.

9. **reproducir\_todo:** Esta función reproduce un archivo multimedia (ya sea imagen, audio o video) utilizando la biblioteca vlc. Crea una instancia de vlc, carga el archivo multimedia y lo reproduce.
10. **usbMedia:** Esta función principal se encarga de detectar y reproducir archivos multimedia (videos, audios e imágenes) desde un dispositivo USB conectado. Utiliza las funciones anteriores para detectar los archivos y reproducirlos según su tipo. Aquí me es importante mencionar que esta función está tratando la excepción de no encontrar los archivos de la USB, en caso de que se desconecte, esto para que no se rompa el programa y lance instancias de vlc sin imágenes.
11. **AbrirPagina:** Esta función abre una página web en el navegador utilizando la biblioteca webbrowser. Recibe una URL como argumento y utiliza el comando os.system para abrir la URL en un navegador específico (en este caso, Chromium en modo kiosko).
12. **Configuración de la interfaz gráfica:** A continuación, se crea la ventana principal y se configura su apariencia. Se definen varios marcos (frames) para organizar los elementos de la interfaz en secciones.
13. **Creación de widgets y botones:** Se crean varios widgets y botones dentro de los marcos de la interfaz gráfica. Estos botones están

En general el programa está corriendo la interfaz gráfica con los botones correspondientes para los servicios de streaming, y sin importar el servicio que escoja el programa ejecuta chromium en modo kiosko mostrando el sitio web del servicio elegido. Mientras que a la par está escuchando activamente si hay una USB conectada y si encuentra una muestra el label de "USB MEDIA" para después internamente detectar que tipo de archivos tiene la usb y según el tipo los reproduce de forma diferente.

## 8. Cuestionario

- 1) ¿Cuál es el proceso para agregar o eliminar servicios de streaming en el centro multimedia?  
¿Requiere conocimientos técnicos avanzados o se puede hacer de manera sencilla?
- 2) ¿Qué medidas se han tomado para facilitar la detección y reproducción automática de contenido multimedia almacenado en dispositivos USB conectados?
- 3) ¿Se está considerando la posibilidad de agregar funciones adicionales, como la capacidad de reproducir archivos de video o música locales almacenados en la Raspberry Pi?

## 9. Mejoras

- Si bien es un prototipo en el cual se necesita profundizar en el desarrollo e implementación para que salga al mercado se puede observar que se le puede agregar un control a larga distancia para poder controlar las operaciones de una manera más remota.
- También se podría estar actualizando el software necesario para tener una mayor calidad en el desarrollo de cada uno de los iconos.

- En el video realizado por fallas de hardware pues se pierde un poco la calidad de la implementación, por lo que se sugirió tener uno de repuestos.
- La organización del cableado es una medida de seguridad que permitiría resolver los cables ordenados y ocultos para evitar enredos y darle un aspecto más limpio al centro multimedia. Utiliza canaletas de cableado o sistemas de gestión de cables para mantener todo organizado.
- Sin contar las funcionalidades que se intentaron, pero no se lograron como son la detección de cualquier tipo de USB y eventos para controlar la app desde la GUI y no desde la consola.

## 10. Conclusión

Durante este proyecto de desarrollo de un centro multimedia enfocado a ser un sistema embebido, he adquirido conocimientos fundamentales sobre la importancia de optimizar los recursos de hardware y software en dispositivos con capacidades limitadas. He explorado diferentes estrategias de diseño y desarrollo para lograr un funcionamiento eficiente y fluido en un entorno embebido. Además, he aprendido más acerca de las configuraciones básicas de un sistema operativo (como es Linux) en especial de la cuestión gráfica, a pesar de no haberlo podido realizar con nodm. En resumen, este proyecto nos ha permitido comprender los desafíos y oportunidades que existen al diseñar y desarrollar sistemas embebidos para aplicaciones multimedia, y nos ha brindado una base sólida para futuros proyectos en este campo.

## 11. Links del proyecto

Github del proyecto: <https://github.com/Y4smin/Ruko-s-Multimedia>

Video Demo Ruko's Multimedia: <https://www.youtube.com/watch?v=QpEQHRHVqXA>

## 12. Bibliografía

Raspberry Pi Documentation - Raspberry Pi hardware. (s. f.).

<https://www.raspberrypi.com/documentation/computers/raspberrypi.html>

Graphical User Interfaces with Tk. (s/f). Python documentation. Recuperado el 17 de enero de 2023, de <https://docs.python.org/3/library/tk.html>

pildorasinformaticas [@pildorasinformaticas]. (2018, enero 31). Curso Python. Interfaces graficas I. Vídeo 42. Youtube. <https://www.youtube.com/watch?v=hTUJC8HsC2I&list=PLqlQ2-9ypflQQEepQJvGQ6RJ8llnzk6Kj>

About FFmpeg. (s. f.). <https://ffmpeg.org/about.html>

VideoLAN. (s. f.). *Official download of VLC media player, the best Open Source player* VideoLAN. <https://www.videolan.org/vlc/>

pyudev. (2023, 21 abril). PyPI. <https://pypi.org/project/pyudev/>

Ltd, R. P. (s. f.). Raspberry Pi OS – Raspberry Pi. Raspberry Pi. <https://www.raspberrypi.com/software/>