

Proyecto: Centro Multimedia

Fundamentos de Sistemas Embebidos

Jesús Vázquez Romero

28 de mayo de 2025

1 Objetivo

Poner en práctica los conocimientos adquiridos durante el laboratorio y la teoría de Fundamentos de Sistemas Embebidos, logrando implementar un centro multimedia para la reproducción de películas, videos, música e imágenes.

2 Introducción

En la actualidad, el contenido multimedia ha aumentado considerablemente, tanto para educación como, sobre todo, para entretenimiento, lo que ha incrementado el uso de dispositivos dedicados. Los sistemas embebidos, por su bajo costo y eficiencia energética, son ideales para el desarrollo de dispositivos con funciones específicas. La Raspberry Pi es una herramienta fundamental para la creación de estos sistemas.

El presente proyecto se enfoca en desarrollar un Centro Multimedia utilizando una Raspberry Pi, basada en Raspbian Lite, un sistema operativo ligero que proporciona el rendimiento adecuado con recursos limitados. Se desarrolla una interfaz gráfica con Pygame, que permite al usuario navegar por el centro multimedia de forma amigable, visualizando contenido almacenado en USB (imágenes, videos, música), contenido de servicios de streaming y también configurando la red Wi-Fi, ofreciendo un servicio fluido. Este sistema prioriza la facilidad de uso para el usuario y el consumo mínimo de recursos.

3 Antecedentes Teóricos

Sistema Embebido. Es un sistema diseñado para realizar funciones específicas dentro de un sistema mecánico o electrónico más grande. Estos sistemas están optimizados para cumplir con tareas específicas, lo que permite que sean más pequeños, económicos y eficientes en energía.¹

Raspberry Pi. Es una computadora de bajo costo, de tamaño pequeño, aproximadamente del tamaño de una tarjeta bancaria, siendo útil para desarrollo y educación. Permite trabajar con hardware externo.²

Raspbian Lite. Es un sistema operativo basado en Debian, optimizado para su uso en una Raspberry Pi que contiene un procesador ARM. Permite la gestión de paquetes APT y soporta hardware. La versión Lite permite construir proyectos de forma más controlada, instalando solo los componentes necesarios para nuestros proyectos, siendo más ligero y eficiente.³

¹**Sistema Embebido.** Embedded: The Benefits of Embedded Systems | Lenovo US. (2023, 28 mayo). <https://bit.ly/4mAkily>

²**Raspberry Pi.** Halfacree, G. (2018). THE OFFICIAL Raspberry Pi Beginner's Guide How to use your new computer. Phil King. <https://n9.c1/lm5h0>

³**Raspbian Lite.** Raspberry Pi. "Raspberry Pi Documentation - Getting Started." www.raspberrypi.com/documentation/computers/getting-started.html.

Centro Multimedia. Permite a los usuarios acceder y reproducir contenido multimedia, como videos, música y fotos, desde almacenamiento local o internet. Logra llevar entretenimiento o educación a un lugar y presentarlo de forma amigable en un monitor, televisión o proyector.⁴

4 Lista de Materiales

- 1 laptop que permita leer y escribir tarjetas microSD y con conexión a internet.
- 1 Raspberry Pi 4Gb.
- 1 memoria MicroSD mínimo 8Gb.
- Monitor con salida VGA o HDMI.
- Adaptador de VGA a microHDMI o HDMI a microHDMI.
- Una fuente de alimentación de 5V@2A con adaptador USB-C.
- Bocinas en caso de que el monitor no tenga salida de audio.
- 1 Teclado USB.
- 1 Mouse USB.

5 Funcionamiento de Raspberry Pi

Funciona como cualquier computadora personal, pero a una escala reducida, enfocándose en la eficiencia y el bajo costo. Para el Centro Multimedia se utilizó una Raspberry Pi 4 de 4Gb, que contiene:

Sistema en Chip (SoC). Es el cerebro de la Raspberry Pi; en este chip se integran los componentes que en una computadora normal están por separado. Contiene:

- CPU de 4 núcleos.
- GPU que se encarga de procesar gráficos y video.
- Memoria RAM que permite al sistema almacenar temporalmente datos y programas en uso. La Raspberry Pi 4 contiene 4 GB de RAM.
- Integra controladores para USB, Ethernet, Wi-Fi, Bluetooth, etc.

Tarjeta MicroSD. El almacenamiento se realiza mediante una Tarjeta MicroSD (donde se instala el sistema operativo), que guardará todos los archivos y programas. La velocidad y el tamaño de la MicroSD son importantes para un mejor rendimiento.

Sistema Operativo. El sistema operativo es fundamental para ejecutar nuestros proyectos y poder interactuar con nuestra Raspberry Pi.

Entradas y Salidas. La Raspberry Pi contiene lo siguiente:

- Puertos USB.
- Puertos MicroHDMI.

⁴**Centro Multimedia.** Centro Multimedia. (s.f.). Los Diccionarios y las Enciclopedias Sobre el Académico. <https://es-academic.com/dic.nsf/eswiki/245848>

- Puerto Ethernet.
- GPIO.

Para saber más de la Raspberry Pi, se puede consultar la documentación: <https://www.raspberrypi.com/documentation/computers/getting-started.html>.

6 Configuración de Raspberry Pi

Obtener el Sistema Operativo. En la página oficial de Raspberry Pi se puede encontrar la imagen del sistema operativo: <https://www.raspberrypi.com/software/operating-systems/>. Es importante descargar Raspberry Pi OS Lite.

Preparar Tarjeta MicroSD. Una vez descargada la imagen, se necesita usar una herramienta para grabarla en la MicroSD. Se recomienda Balena Etcher, que tiene soporte para Windows, Linux y macOS <https://etcher.balena.io/>. Una vez instalado Balena Etcher, se debe seleccionar la imagen descargada y la MicroSD a grabar. Este proceso puede tardar unos minutos.

Primer Arranque. Completada la grabación, se inserta la MicroSD en la Raspberry Pi y se conecta a la corriente. La Raspberry Pi lee el sistema operativo desde la tarjeta y arranca. Nos pedirá ingresar la contraseña de root, nombre de usuario y su contraseña. Después se reiniciará y finalmente podremos ocupar nuestra Raspberry Pi con el sistema operativo instalado.

Conexión Wi-Fi y GPU. Para poder descargar nuestros requerimientos para el Centro Multimedia, es necesario conectarse a internet.

- Usamos el comando `sudo raspi-config`.
- Una vez dentro de la interfaz, seleccionamos la opción 1 – System Option.
- Después seleccionamos Wireless LAN.
- Ahora se ingresa el Wi-Fi de nuestra preferencia, introducimos el SSID y la contraseña.
- Para evitar problemas al ejecutar nuestro Centro Multimedia y reproducir nuestro contenido, modificaremos la GPU. Regresamos al menú principal de `raspi-config`, ahora seleccionamos 4 Performance Options.
- Después se selecciona P2 GPU Memory y asignamos 256 MB. Salimos.
- Para comprobar si se consiguió conectar a internet, podemos hacer un ping a Google con `ping 8.8.8.8`.

Inicio Limpio. Para evitar que aparezca tanto texto al iniciar la Raspberry Pi, se debe modificar el archivo `cmdline.txt`. Se ejecuta el siguiente comando: `sudo nano /boot/cmdline.txt` y se modifica lo que se encuentre por:

```
console=serial0,115200 console=tty1 root=PARTUUID=3c5dd69d-02 rootfstype=ext4
fsck.repair=yes rootwait quiet splash plymouth.ignore-system-theme logo.nologo
```

Actualizamos con `sudo update-initramfs -u` para que los cambios surtan efecto en el arranque.

Arranque en modo Kiosko. Es importante para este proyecto que el Centro Multimedia cargue después de que se inicie el sistema operativo, por lo que se realiza lo siguiente:

```
sudo nano /etc/rc.local
```

Una vez dentro del archivo `rc.local`, se debe agregar el comando `python3 /home/jesus/proyecto1.py` a ejecutar para que, al arrancar, se inicie. Es importante añadirlo antes de `exit 0`. Al final se deben dar los permisos para que pueda ejecutarse: `sudo chmod +x /etc/rc.local`.

7 Instalación de Requerimientos para Centro Multimedia

Para la instalación se desarrolló un código Shell, para ahorrar tiempo y comandos por ejecutar por el usuario. Para poder obtener el código, es necesario instalar Git. Primero hacemos una actualización del sistema con `sudo apt update` y después instalamos Git: `sudo apt install git -y`.

Ahora clonamos el repositorio: `git clone https://github.com/JesusSpliner/Centro_Multimedia.` Nos movemos al repositorio: `cd Centro_Multimedia` Damos los permisos para la ejecución del script: `chmod +x instalador.sh` Ejecutamos el script: `sudo ./instalador.sh`

7.1 ¿Qué contiene el Script?

Actualización del Sistema.

- `apt update`. Actualiza la lista de paquetes disponibles a las últimas versiones.⁵
- `apt upgrade`. Actualiza todos los paquetes ya instalados en el sistema a sus versiones recientes.⁵

Instalación de Componentes Gráficos y de Entorno Mínimo.

- `xserver-xorg`. Es el servidor X Window System, importante para el entorno gráfico.⁶
- `xinit`. Permite iniciar manualmente una sesión del servidor X.⁶
- `openbox`. Un gestor de ventanas ligero y eficiente.⁷
- `x11-xserver-utils`. Utilidades adicionales para el servidor X.⁶
- `lxterminal`. Emulador de terminal ligero.⁶

Software de Reproducción Multimedia.

- `vlc`. Reproductor multimedia VLC. Permite manejar la reproducción de video y audio.⁸

Gestión de Red (Wi-Fi).

- `network-manager`. Permite gestionar las configuraciones de la red.⁹
- `network-manager-gnome`. Es importante porque brinda la herramienta `nmcli` que ayuda a conectarse a redes Wi-Fi.⁹

Gestión de Almacenamiento USB.

⁵ `apt update`. Canonical. (s.f.). Ubuntu Manpage:apt - command-line interface. <https://manpages.ubuntu.com/manpages/questing/en/man8/apt.8.html>

⁶ `xserver-xorg`. X.Org.(s.f.)www.x.org/wiki/Documentation/

⁷ `openbox`. Home - Openbox. (s.f.). Openbox. <https://openbox.org/>

⁸ `vlc`. VideoLAN. (s.f.). Official download of VLC media player, the best Open Source player - VideoLAN. <https://www.videolan.org/vlc/index.html>

⁹ `network-manager`. NetworkManager Reference Manual: NetworkManager Reference Manual. (s.f.). <https://networkmanager.dev/docs/api/latest/>

- **udisks2.** Gestiona los dispositivos de almacenamiento, permite el montaje y desmontaje de unidades USB.¹⁰
- **util-linux.** Contiene la herramienta `findmnt`, que ayuda a encontrar el punto de montaje de los dispositivos USB.¹¹

Navegador Web.

- **chromium-browser.** Es un navegador web basado en Chromium.¹²
- **Widevine DRM.** Permite la reproducción de contenido de los servicios de streaming.¹³

Herramientas de Python.

- **python3-pip.** Es el sistema de gestión de paquetes preferido para Python 3, permite instalar librerías de Python.¹⁴
- **python3-dev.** Contiene archivos de cabecera y librerías importantes para compilar extensiones de Python. Es importante porque algunas librerías contienen componentes de C.¹⁵
- **python3-pygame.** Es una herramienta ligera que en este caso nos permitirá crear la GUI y el manejo de multimedia en Python. Se instalan otras librerías base que utiliza Pygame para manejar formatos, audio, gráficos, entradas y renderizado, como `libgl1-mesa-glx`.¹⁶
- **python-vlc.** Permite la interacción de Python con VLC.⁸
- **pyudev.** Ayuda a detectar y gestionar los dispositivos conectados a través de udev.¹⁷

8 Creación del Centro Multimedia

Se describirá de forma general el código del Centro Multimedia Pi, detallando los módulos necesarios para lograr su propósito.

8.1 Menú Principal

Se crearon 8 botones: Netflix, Spotify, Prime Video, Apple Music, Disney+, USB, Salir y Wi-Fi. Se ha utilizado Pygame para crear la interfaz. Primero se definió el tamaño donde se dibuja el menú, y es importante que la interfaz esté en modo kiosko, por lo que se ocupa la función `pygame.FULLSCREEN`. También fue importante añadir los colores a usar tanto en el fondo, botones y texto, además de definir un estilo de texto y el tamaño.

Se añadieron los iconos, se cargan uno por uno, por ejemplo Netflix: `icons['netflix'] = pygame.image.load(os.path.join(ICON_PATH, 'netflix.png')).convert_alpha()` Va a la carpeta `icons`, encuentra el archivo `netflix.png`, carga el contenido, lo optimiza para dibujar en la pantalla y almacena la imagen con el nombre `netflix` para facilitar su uso. Esto se repite para todos los iconos y botones creados. Se asignan tamaños específicos, distinto para las aplicaciones que para el ícono

¹⁰ **udisks2.** Canonical. (s.f.-b). Ubuntu Manpage: `udisksctl` - The udisks command line tool. <https://manpages.ubuntu.com/manpages/xenial/man1/udisksctl.1.html>

¹¹ **util-linux.** Canonical. (s.f.-b). Ubuntu Manpage: `findmnt` - find a filesystem. <https://manpages.ubuntu.com/manpages/lunar/man8/findmnt.8.html>

¹² **Chromium.** Chromium. (s.f.). <https://www.chromium.org/Home/>

¹³ **Widevine.** Widevine DRM | Widevine. (s.f.). <https://www.widevine.com/solutions/widevine-drm>

¹⁴ **pip.** pip documentation v25.1.1. (s.f.). <https://pip.pypa.io/en/stable/>

¹⁵ **python3-dev.** Python 3.13 documentation. (s.f.). Python Documentation. <https://docs.python.org/3/>

¹⁶ **python3-pygame.** pygame news. (s.f.). <https://www.pygame.org/news>

¹⁷ **pyudev.** pyudev – pure Python libudev binding — pyudev 0.21.0 documentation. (s.f.). <https://pyudev.readthedocs.io/en/latest/>

de Wi-Fi, y para no alargar el código y estar especificando cada vez que se utilice el icono, se hizo un bucle `for` para asignar su tamaño a cada ícono. Este proceso se repite para cada ícono, asignando así el tamaño para cada uno.



Figure 1: Menú del Centro Multimedia

Ahora, para lograr que los botones al interactuar con ellos abran las aplicaciones externas, se hizo de la siguiente forma:

```
if create_button_with_icon('netflix', btn_netflix_rect, GRAY, LIGHT_BLUE,
"Netflix"): subprocess.Popen(['chromium-browser', '--kiosk', 'https://www.netflix.com'])
return STATE_MAIN\_MENU
```

Se llama a la función `create_button_with_icon` que crea el botón y detecta si ha sido presionado. En caso de ser presionado, llama a la función `subprocess.Popen` que se encarga de ejecutar comandos externos. Dentro de la función se especifica el nombre del programa a ejecutar, en este caso `chromium-browser`, en modo kiosko y se asigna la URL de Netflix. El `return` permite que el Centro Multimedia continúe ejecutándose en segundo plano, para que una vez cerrada la aplicación Netflix, se pueda regresar al Centro Multimedia. Este proceso se repite para todas las aplicaciones.

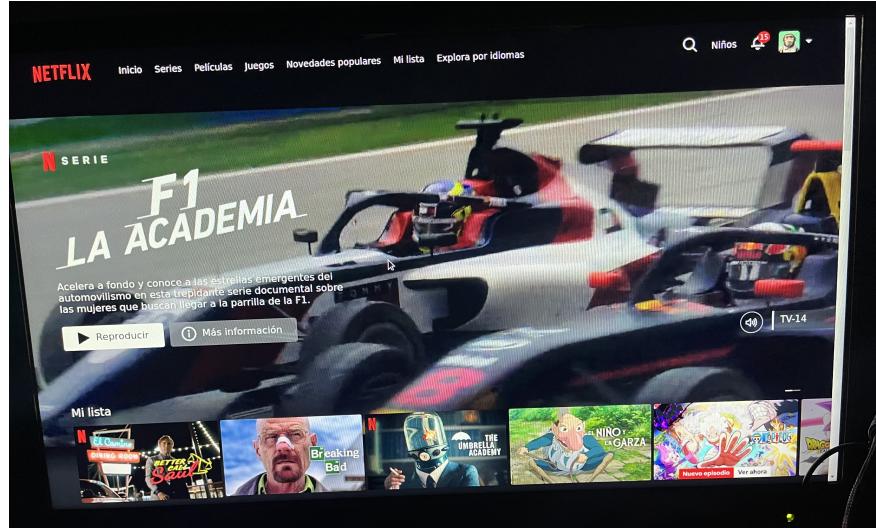


Figure 2: Netflix ejecutado

8.2 Detección USB

La detección de USB es importante porque permite interactuar con medios externos a partir de nuestro Centro Multimedia. Se creó un hilo `usb_monitor_thread_func` que se ejecuta de forma separada a la interfaz, para saber cuándo una USB ha sido conectada o desconectada.

Se usa la librería `pyudev` que permite detectar eventos que suceden en el sistema, permitiendo interactuar con los dispositivos que han sido conectados. Para la detección y montaje, primero obtiene la ruta del dispositivo `dev_path = device.device_node` y monta la USB en caso de que no lo esté `auto_mount(dev_path)`. Se utiliza `udisksctl` que previamente hemos instalado.

Una vez montada la USB, se escanean los archivos multimedia con `get_media_files(mount_point)`, escaneando todos los archivos y directorios existentes. Se buscan archivos a partir de su extensión, se agregan varias extensiones para que detecte no solo un tipo.

```
if lower_file.endswith('.jpg', '.jpeg', '.png', '.gif', '.bmp'):
    photos.append(file_path)
elif lower_file.endswith('.mp3', '.wav', '.flac', '.ogg'):
    music.append(file_path)
elif lower_file.endswith('.mp4', '.avi', '.mkv', '.mov', '.webm'):
    videos.append(file_path)
```

Estos archivos, dependiendo de su extensión, se almacenarán en las listas previamente creadas: `photos`, `music` y `videos`. Estos datos que se guardan en las listas se copian al diccionario `usb_data`. Aquí `usb_data` actúa como la parte importante para la reproducción del contenido de la USB.

Para ligar lo anterior explicado con los botones y que al hacer clic reproduzca lo seleccionado, se implementa la siguiente lógica (para video, por ejemplo):

```
if create_button_with_icon('videos_logo', btn_videos_rect, GRAY, LIGHT_BLUE, "Videos"):
    if usb_data['videos']:
        play_video_selection_vlc(usb_data['videos'])
        return STATE_USB_VIDEO_SELECTION
    else:
        return STATE_USB_LOADING
```

Se crea el botón y detecta cuando ha sido presionado. Se verifica que la lista `videos` no esté vacía y, en caso de que contenga las rutas de los videos, se llama a `play_video_selection_vlc` para que sepa qué va a

reproducir. En caso de que la lista se encuentre vacía, entonces se mostrará en pantalla que está cargando o que se inserte la USB. Este proceso se repite de forma similar para imágenes y música.



Figure 3: Menú de USB

Continuando con el ejemplo de video, `play_video_selection_vlc` se encarga de verificar que haya videos a reproducir y se crea un submenú que se encarga de dibujar. Se añade el botón para volver al menú principal, botones individuales para los videos que detectó y un botón para reproducir en modo presentación. Finalmente, se detecta cuando el usuario da clic a alguno de los botones que se crearon para seleccionar un video específico o en modo presentación y se ejecuta la función `play_video_slideshow_vlc`, que crea un archivo de lista de reproducción temporal `.m3u`, que contiene las rutas de los videos y que llama a la función `play_media_vlc_external()` pasándole la ruta del archivo `.m3u`, permitiendo reproducir los videos. Este proceso es similar para fotos y música.



Figure 4: Reproducción de video

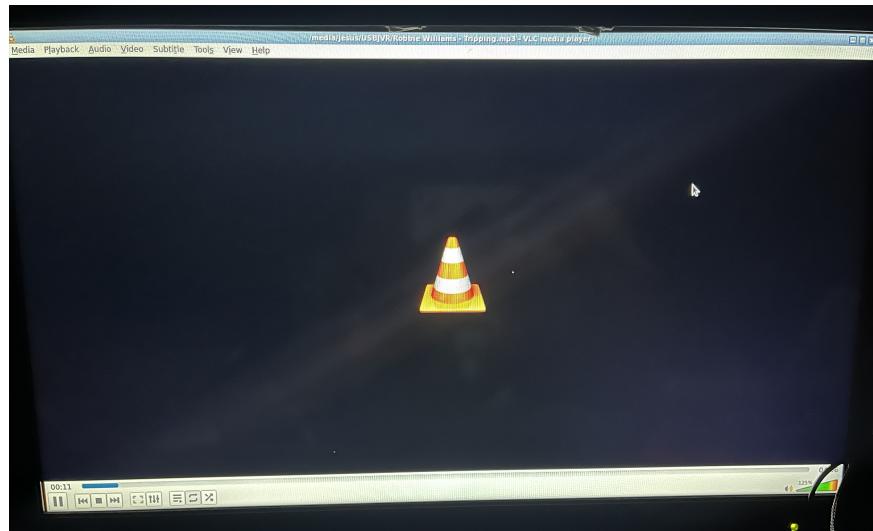


Figure 5: Reproducción de música

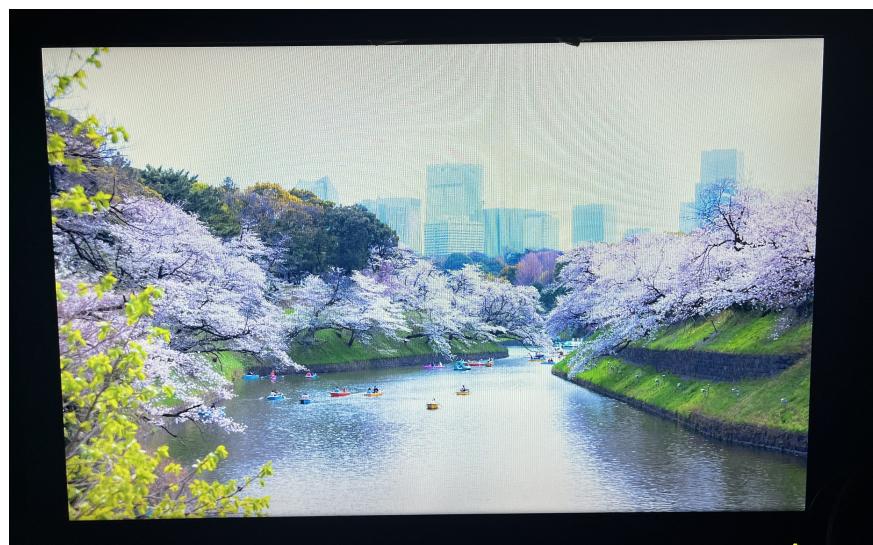


Figure 6: Reproducción de imágenes

8.3 Conexión Wi-Fi

Para conseguir conectar Wi-Fi, se creó un ícono en el menú principal, ubicado en la parte superior derecha y es un logo de Wi-Fi. Al presionarlo, abrirá un submenú que es una interfaz básica donde el usuario ingresa dos parámetros: el SSID del Wi-Fi al que se quiere conectar y la contraseña. Al hacer clic en conectar, el código guarda los datos ingresados por el usuario y, en resumen, la conexión se logra de la siguiente forma:

```
def connect_to_wifi_logic(ssid, password): # La función recibe 'ssid' y 'password'
    try:
        # ... asegurar NetworkManager ...
        subprocess.run(['sudo', 'nmcli', 'device', 'wifi',
```

```

    'connect', 'ssid', ssid, 'password', password],
    check=True, stdout=subprocess.PIPE, stderr=subprocess.PIPE)
print(f"Conectado exitosamente a la red Wi-Fi '{ssid}'.")
return True

# ... manejo de errores ...

```

Se utiliza la forma estándar y común para interactuar con el sistema operativo y lograr una conexión Wi-Fi desde la terminal. `sudo` es para que se ejecute con permisos de administrador, `nmcli` es la herramienta de línea de comandos de NetworkManager, `device wifi` indica que se va a interactuar con un dispositivo Wi-Fi, `connect` es la acción a realizar, `ssid` indica que el siguiente argumento es el nombre de la red, `password` indica que el siguiente argumento es la contraseña de la red. `ssid` y `password`, sin comillas, son los parámetros que ingresó el usuario.



Figure 7: Interfaz de Wi-Fi

Si se logra conectar, aparecerá una pantalla en verde, indicando que se pudo realizar la conexión.



Figure 8: Conexión WiFi exitosa

IMPORTANTE. El video demostrativo se encuentra en YouTube:
FUNDAMENTOS DE SISTEMAS EMBEBIDOS - PROYECTO - CENTRO MULTIMEDIA

IMPORTANTE. El repositorio del proyecto se encuentra en GitHub: Centro_Multimedia

9 Conclusión

El proyecto se concretó de forma exitosa, logrando establecer un Centro Multimedia en la Raspberry Pi. Se cumplió con el objetivo de reproducir películas, videos, música e imágenes, poniendo en práctica lo aprendido durante las clases de laboratorio y teoría de Fundamentos de Sistemas Embebidos.

La aplicación, desarrollada con Pygame, ofrece una navegación intuitiva para cualquier usuario. También se consiguió reproducir contenido multimedia desde una USB. Aunque no se logró el propósito de prescindir del teclado, sí se pudo detectar la USB, montarla, leer los archivos y reproducir el contenido multimedia encontrado en ella.

La conexión WiFi también cumple su propósito, si bien puede tardar máximo 30 segundos en confirmar la conexión exitosa, no hay errores en la conexión. Este proyecto resultó muy laborioso al ser realizado individualmente. No obstante, considero que los conocimientos adquiridos fueron mayores precisamente por la necesidad de resolver los problemas de forma autónoma. En general, creo que el proyecto está a la altura del nivel de las clases y prácticas realizadas durante el curso.

10 Referencias

- Halfacree, G.(2018). THE OFFICIAL Raspberry Pi Beginner's Guide How to use your new computer. Phil King. <https://n9.cl/lm5h0>
- Raspberry Pi. (n.d.). Raspberry Pi Documentation - Getting Started. [Www.raspberrypi.com](http://www.raspberrypi.com/documentation/computers/getting-started.html). [https://www.raspberrypi.com/documentation/computers/getting-started.html](http://www.raspberrypi.com/documentation/computers/getting-started.html)