

Proyecto Final

Centro Multimedia

Fundamentos de Sistemas Embebidos

Autor: Cervantes Guati Rojo Juan Andrés.

Objetivo

Implementar un centro multimedia utilizando una tarjeta controladora Raspberry Pi capaz de reproducir películas, video, música y fotografías.

Introducción

A continuación, se describe cómo realizar un centro multimedia considerando elementos de los sistemas embebidos que se vieron en clase. El sistema debe ser capaz de reproducir audio y video en bucle además de poder reproducir servicios en línea.

Antecedentes

La *Raspberry Pi* se describe como “una computadora completamente funcional en un pequeño paquete de bajo costo” puesto que permite desde navegar en la Web como para jugar, sin embargo, puede apoyar a la creación o innovación de circuitos y dispositivos

Raspbian es el sistema operativo más popular para Raspberry Pi, además de ser el único con soporte oficial. Raspbian es una distribución de Linux basada en Debian, optimizado para la Raspberry Pi y que permite a esta operar como una PC. La distro incorpora terminal y navegador web entre otros programas.

Un *Centro Multimedia* es un computador adaptado para reproducir música, visualizar películas e imágenes almacenados en un disco duro local o sobre una red de computadoras LAN (en algunos casos), visualizar películas DVD y a menudo visualizar y grabar emisiones de televisión.

Lista de materiales

- ❖ 1 Raspberri Pi 4 Modelo B
- ❖ 1 Mouse con entrada USB
- ❖ 1 Teclado con entrada USB
- ❖ 1 Dispositivo de audio con conexión 3.5mm para poder escuchar el sonido de salida
- ❖ 1 Conector micro HDMI, se puede utilizar adaptador HDMI a micro HDMI
- ❖ 1 Monitor o dispositivo de salida de video con conexión a HDMI o VGA(Usar adaptador)
- ❖ 1 fuente de alimentación regulada a 5V y al menos 2 amperios de salida

Descripción del funcionamiento de la tarjeta controladora

Se va a utilizar una Raspberry Pi 4 Modelo B, por lo cual se va a describir brevemente su funcionamiento:

Funciona con el sistema operativo Raspbian

Se va a utilizar una Raspberry Pi 4 Modelo B, por lo que se va a describir brevemente su funcionamiento:

Funciona con el sistema operativo Raspbian, una vez energizado el microcontrolador el kernel procede a cargar el sistema operativo. Se recomienda ir al siguiente link para conocer más sobre la instalación de Raspbian:

[Instalación Raspbian.](#)

Contamos con pines GPIO, estos nos serán de utilidad para recibir señales de otros dispositivos, ya sea de otro microcontrolador como un PIC o un Arduino, así como también de un circuito eléctrico.

Además contamos con entradas USB 2.0, USB 3.0 y entrada Ethernet. La raspberry Pi 4 Modelo B cuenta con soporte para conexiones Bluetooth y WiFi por lo que podremos realizar conexiones inalámbricas con otros dispositivos.

Configuración de la tarjeta controladora

Ejecución al momento de arranque

La configuración que se debe hacer en la Raspberry para este proyecto es hacer que nuestro programa se inicie después que el kernel cargue el sistema operativo, para ello se deben realizar los siguientes pasos:

- En una terminal (que puede abrirse con CTRL + ALT + T) ejecutar el siguiente comando:

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

- Se abrirá un archivo y debemos ir hasta abajo antes del renglón “exit 0”, justo ahí debemos escribir el comando para ejecutar nuestra aplicación. En este caso se ingresa el siguiente comando:

```
python3 /home/raspberry/Desktop/PiTV/piTV.py
```

De esta manera se indica que se debe ejecutar el comando después de la carga del sistema operativo.

- No se debe olvidar cambiar los permisos del archivo rc.local para permitir que pueda ejecutarse, esto se realiza con el siguiente comando:

```
sudo chmod +x /etc/rc.local
```

- Puede comprobarse el cambio con el siguiente comando para ver los permisos de los archivos:

```
ls -l
```

Listo, con estas configuraciones nuestra aplicación se ejecutará al momento de arrancar nuestra Raspberry Pi. Se puede hacer la prueba reiniciando el microcontrolador.

De igual manera vamos a necesitar varias bibliotecas para poder ejecutar las tareas que vamos a realizar, a continuación se muestran las bibliotecas necesarias así como el comando para instalarlas.

Bibliotecas

pip3: Nos va a ayudar a instalar y administrar paquetes de software escritos en Python.[4]

```
sudo apt install python3-pip
```

tkinter: Nos va a permitir realizar interfaces gráficas en Python.[5]

```
sudo apt install tk
```

pillow: Nos dará algunas capacidades en el procesamiento de imágenes.[6]

```
sudo apt install pillow
```

cv2(opencv): Nos permitirá manejar imágenes y para el manejo de reproducción de video.[7]

```
pip3 install opencv-python
```

imutils: Permite realizar varias operaciones con las imágenes como rotación, traslación, escalamiento, etc.[8]

```
pip3 install imutils
```

audioplayer: Va a permitir reproducir sonidos de tipo mp3 o wav.[9]

```
pip3 install audioplayer
```

pyudev: Nos puede servir para manejo de eventos de USB.[10]

```
pip3 install pyudev
```

Widevine DRM (MUY IMPORTANTE!!!!): Esto nos va a permitir reproducir el contenido de los servicios de streaming.[11]

```
sudo apt-get install libwidevinecdm0
```

En caso de contar con una Raspberry de 64bits es necesario cambiar el navegador por la versión de 32-bits, se puede realizar con el siguiente comando:

```
sudo apt install chromium-browser:armhf libwidevinecdm0
```

Para regresar a la versión de 64-bits se utiliza el siguiente comando:

```
sudo apt install chromium-browser:arm64 libwidevinecdm0-
```

Con todo esto ya se podrá reproducir el contenido de cualquier servicio de streaming en el navegador como Netflix, HBO, Spotify, etc.

Desarrollo de módulos

Vamos a comenzar hablando del código raíz de la aplicación, piTV.py. Se van a describir de manera general todos los módulos que se necesitan para realizar las diferentes funciones que se buscan realizar. El código con la documentación se puede encontrar en el repositorio de GitHub, el enlace se adjunta al final del documento.

Interfaz de menú principal

Comenzamos con la interfaz del menú principal donde debemos mostrar tres botones para poder reproducir Netflix, Spotify y ejecutar el contenido del dispositivo USB. Las interfaces se realizan con tkinter por lo cual debemos colocar los botones con la clase Button y acomodarlos en la ventana, hay diferentes maneras de acomodar los elementos de la UI. Se recomienda revisar la documentación de [tkinter](#) para saber más sobre los elementos que podemos ocupar y cómo organizarlos en la UI.

Nosotros nos centraremos en este apartado en la funcionalidad, no tanto en el aspecto gráfico.

Para indicarle que nuestro botón tiene que realizar una acción al ser presionado le agregamos el argumento "command=nombreFuncion". En nuestro caso se presenta un ejemplo para el botón de Netflix:

```
btn_netflix = Button(frame, image = netflix_logo, command=playNetflix, text="NETFLIX",  
compound="top",width=30, height=30, bg=color_black, fg=color_white, highlightbackground=color_dark_gray,  
activebackground="#272727")
```

Para el caso de los demás botones es lo mismo, debemos adjuntarle una función a cada uno. El funcionamiento de estos botones se describe a continuación.

Apertura de servicio de streaming Netflix

Como podemos observar la función que se manda a llamar en el ejemplo anterior es **playNetflix()**, se encargará de abrir el servicio de streaming de Netflix en un navegador web y lo hace de la siguiente manera:

Se genera una ventana que tendrá un botón de RETURN en caso de querer volver al menú principal y pasando 100 milisegundos se va a ejecutar la función `openNetflixBrowser()`, que se encarga de abrir un navegador con la URL de netflix <https://www.netflix.com>. La pausa de 100 ms es para darle tiempo a la interfaz de aparecer antes de que se abra el navegador para una mejor apariencia de la interfaz.

Apertura de servicio de streaming Spotify

La implementación de este módulo es la misma que la del módulo de Netflix, solamente se cambia la ruta que debe abrir el navegador web, se utiliza la función `openSpotifyBrowser()` y utiliza la URL <https://www.spotify.com>.

Para estos dos módulos anteriores vamos a estar ocupando la biblioteca `webbrowser` para abrir el navegador de la Raspberry Pi.

Cerrar los servicios de streaming

Para este módulo vamos a ocupar la biblioteca de `subprocess` para matar los procesos que estén ejecutando el navegador, esto debido a que no podemos cerrar la ventana del navegador con los botones de su ventana.

Al momento de oprimir el botón de RETURN se cierra la interfaz y se procede a hacer un filtrado de los procesos que se están ejecutando en el tiempo actual, se puede acceder a los procesos con el siguiente comando:

```
ps -e
```

Se implementa la función `kill_process()` la cual se encarga de eliminar todos los procesos que estén ejecutando el navegador. Después de realizar el filtrado se procede a matarlos usando lo siguiente:

```
os.kill(int(pid), SIGTERM)
```

Detección de USB

Para la detección de USB utilizamos la función `monitorUSB()` la cual se ejecuta cada segundo para estar a la escucha de la entrada de una USB.

Para realizar este módulo se va a estar revisando constantemente la carpeta `"/media/raspberry/"` que es donde se almacenan los dispositivos extraíbles. Cada segundo se va a estar revisando, en caso de encontrar una nueva carpeta la guardamos como USB actual y activamos una bandera para indicar que se ingresó una USB, luego se procede a revisar el contenido de la USB para decidir que se tiene que realizar dependiendo del tipo de archivos que contenga. Además habilitamos el botón de Media del menú principal para que pueda ser presionado.

Al momento de desmontarla se desactiva la bandera que indica que una USB se encuentra conectada y deshabilitamos el botón Media del menú principal.

Galería de imágenes

Si nuestra memoria USB contiene imágenes se va a desplegar una nueva ventana con la opción de ver todas las imágenes en forma de galería, podemos movernos de izquierda a derecha y viceversa. Esto puede ser realizado utilizando las bibliotecas `cv2`, `PIL(pillow)` y `mutils`. El funcionamiento de colocar la imagen en pantalla puede visualizarse en la función `putImage()` la cual carga la imagen, realiza un escalamiento de ser necesario y lo coloca en el label correspondiente.

Reproductor de audio

Esto se puede lograr con la biblioteca `audioplayer` que contiene métodos para toda acción relacionada con los sonidos, podemos reproducir, pausar, reanudar y detener un sonido.

Para esto se creó una interfaz con botones que realizan las diferentes acciones. Los sonidos se reproducen en loop y podemos cambiar de un archivo a otro. Es necesario especificar la ruta del archivo de audio que puede ser de la siguiente manera:

```
songToPlay = AudioPlayer("sonido.mp3")
```

Se reproduce y elegimos que se encuentre en loop y que sea no bloqueante para permitirle al usuario seguir utilizando la interfaz:

```
songToPlay.play(loop=True, block=False)
```

Reproductor de video

Esto se puede visualizar en el código videoUI.py

El reproductor de video funciona de una manera similar que la galería de imágenes, ya que ocupa las mismas bibliotecas. La diferencia es que se va a ir cargando frame por frame como se ve en la función visualize(). Los videos pueden ser reproducidos en forma de galería uno tras otro en loop o seleccionando el video nosotros, esto se encuentra implementado con las funciones chooseVideoToPlay() y playGalleryMode() respectivamente.

Integración de módulos

La integración de todos los componentes mencionados anteriormente dan como resultado las siguientes implementaciones:

Menú al encendido que permitirá elegir entre servicios en línea para video (ej. Netflix, HBO-Go, Blim), música (ej. Spotify, Cloudfare, Deezer, etc) o reproducir el medio extraíble.

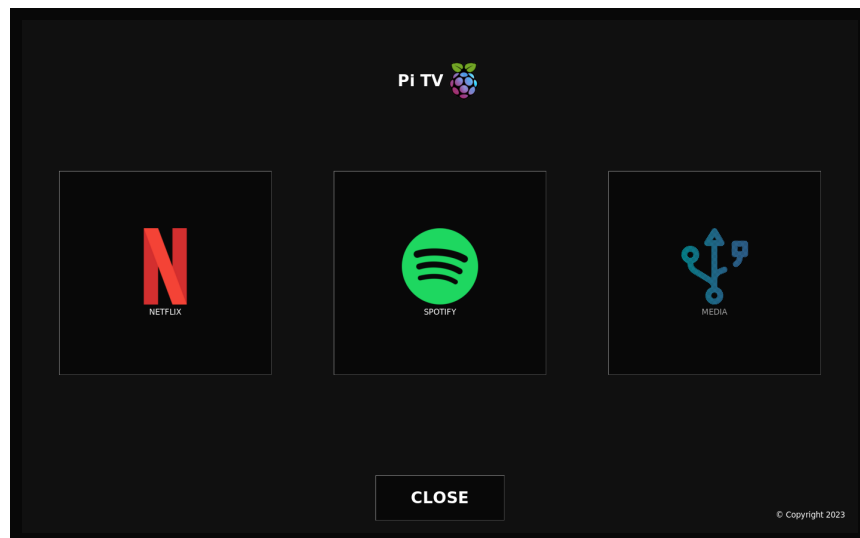


Figura 1. Menú principal de la aplicación

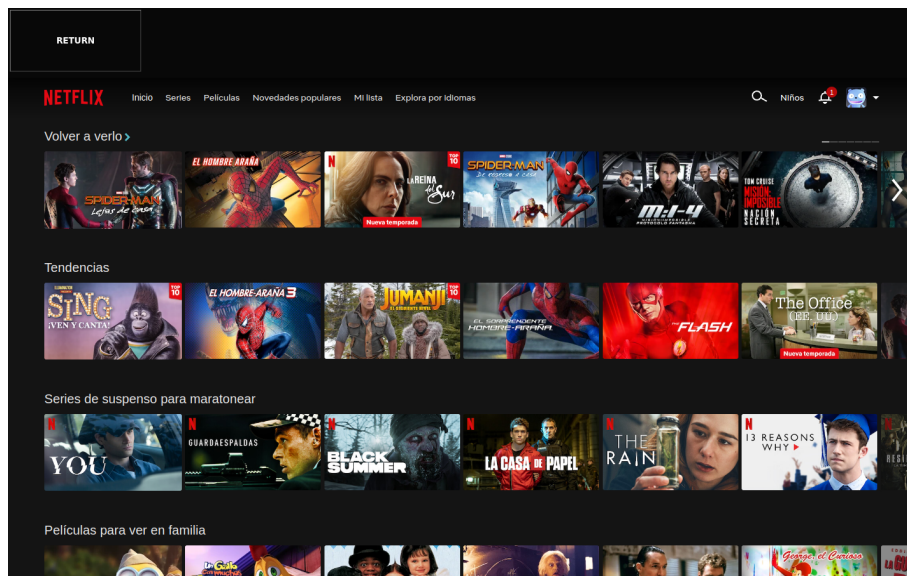


Figura 2. Apertura de servicio de streaming Netflix

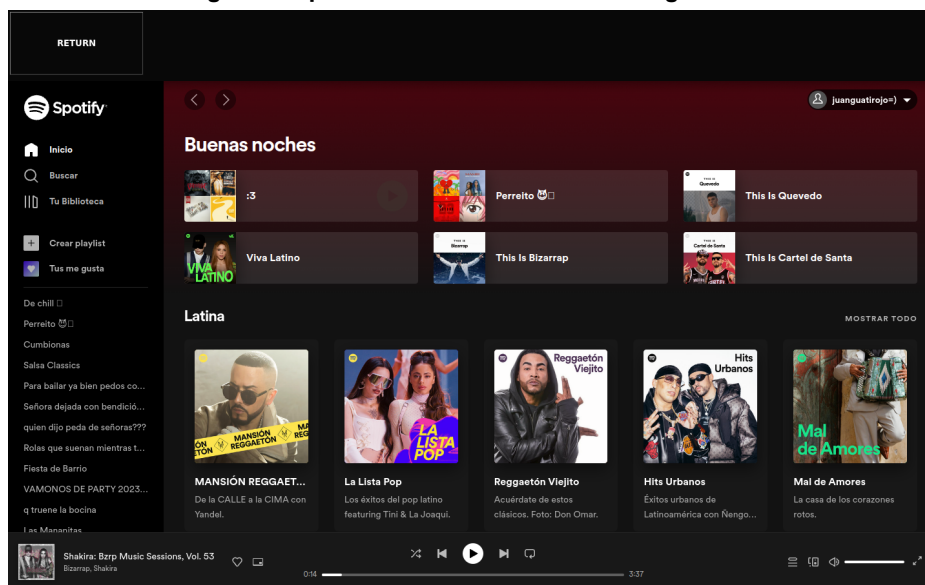


Figura 3. Apertura de servicio de streaming Spotify

Cuando se inserte un medio extraíble (memoria USB) con fotografías, estas serán reproducidas en modo presentación.

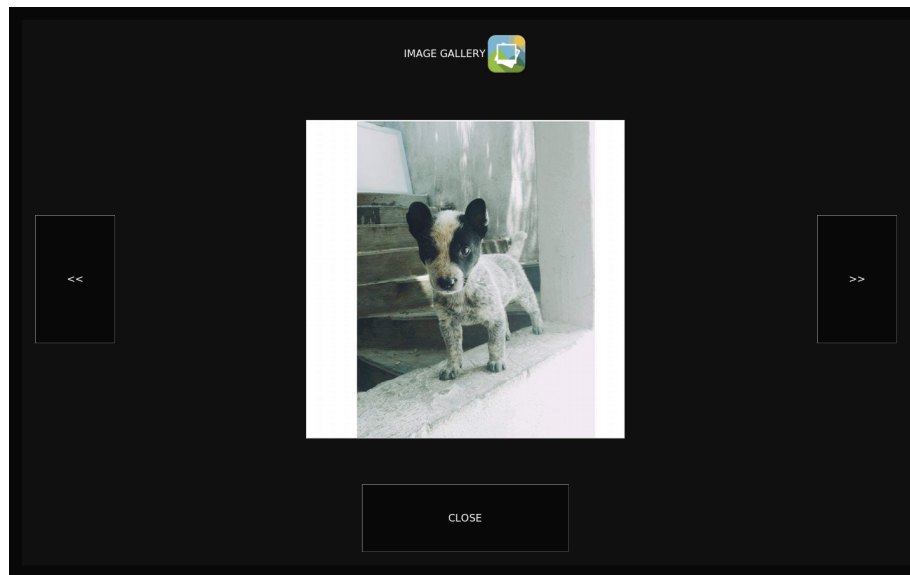


Figura 4. Apertura de Galería de Imágenes

Cuando se inserte un medio extraíble (memoria USB) con música, las pistas serán reproducidas una tras otra en bucle infinito.

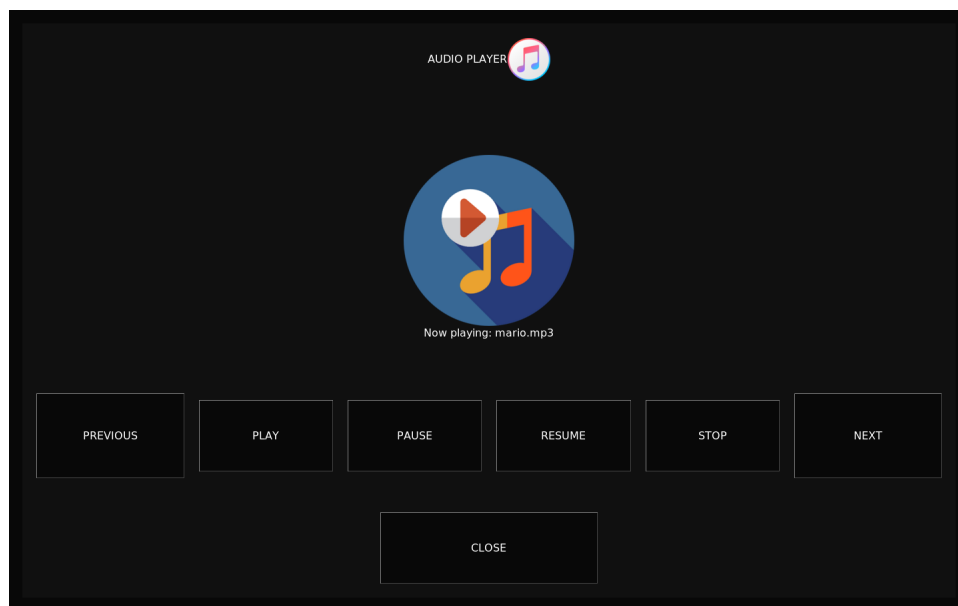


Figura 5. Apertura de Reproductor de Sonidos

Cuando se inserte un medio extraíble (memoria USB) con videos, deberá poderse elegir qué película reproducir o si los videos serán reproducidos uno tras otro en modo presentación.

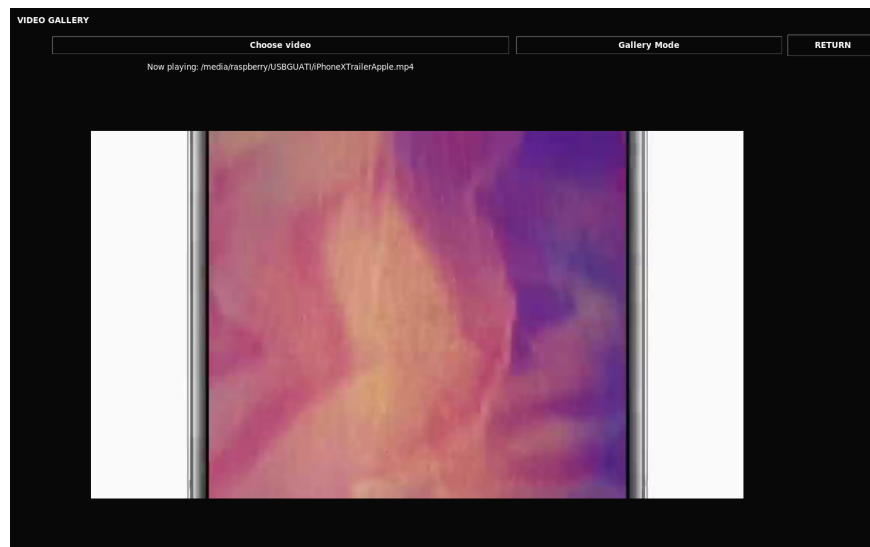


Figura 6. Apertura de Reproductor de video

Cuando se inserte un medio extraíble (memoria USB) con contenido mixto el aparato deberá preguntar qué acción tomar.

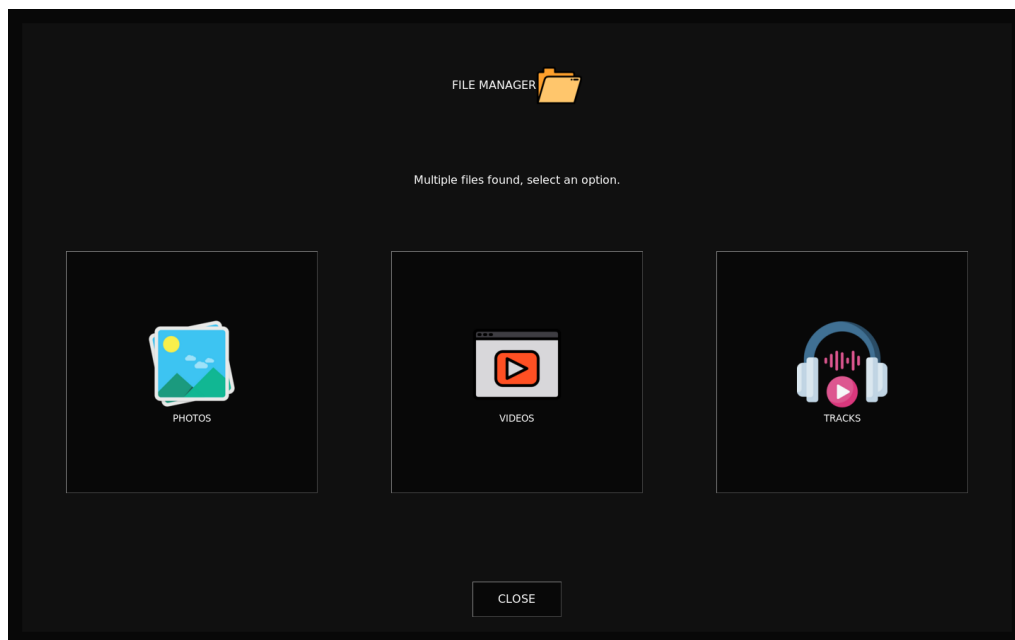


Figura 7. Apertura de Galería de Imágenes

Para ejecutar el código, se debe ejecutar el siguiente comando desde la carpeta src/ del directorio del proyecto.

```
python3 piTV.py
```


Conclusiones

Se logró implementar un centro multimedia utilizando una Raspberry Pi capaz de reproducir servicios de streaming como Netflix y Spotify, además de detectar cuando se ingresa un dispositivo extraíble. Se comprendieron los conceptos de un sistema operativo embebido y los requerimientos que se deben tomar en cuenta tanto de software como de hardware para un funcionamiento óptimo. Lo que más se complicó en el proyecto fue la detección del dispositivo extraíble y el poder arrancar la aplicación después de cargar el sistema operativo, pero fue un gran reto y se aprendió bastante sobre el cómo funciona la Raspberry Pi.

Repositorio GitHub

<https://github.com/JuanGR-FI/PiTV.git>

Video demostrativo

<https://youtu.be/0rp9RVsAHsM>

Bibliografía de consulta

- [1] Halfacree, G. (2018). THE OFFICIAL Raspberry Pi Beginner's Guide How to use your new computer. Phil King. [Archivo PDF].
- [2] Centro Multimedia. (s. f.). Los diccionarios y las enciclopedias sobre el Académico. <https://es-academic.com/dic.nsf/eswiki/245848>
- [3] Barr, M., & Massa, A. (2006). *Programming Embedded Systems: With C And Gnu Development Tools* (2.^a ed.). Oreilly & Associates Inc.
- [4] Muhammad. "How to install pip in Python 3 on Ubuntu 18.04?" Odoos S.A. <https://www.odoo.com/forum/help-1/how-to-install-pip-in-python-3-on-ubuntu-18-04-167715> (accedido el 3 de enero de 2023).
- [5] D. Pawan. "How to install Tkinter in Python?" Online Tutorials Library. <https://www.tutorialspoint.com/how-to-install-tkinter-in-python> (accedido el 23 de diciembre de 2022).
- [6] pipi.org. "Pillow". PyPI. <https://pypi.org/project/Pillow/> (accedido el 28 de diciembre de 2022).
- [7] pipi.org. "opencv". PyPI. <https://pypi.org/project/opencv-python/> (accedido el 30 de diciembre de 2022).
- [8] pipi.org. "imutils". PyPI. <https://pypi.org/project/imutils/> (accedido el 28 de diciembre de 2022).
- [9] pipi.org. "audioplayer". PyPI. <https://pypi.org/project/audioplayer/> (accedido el 27 de diciembre de 2022).
- [10] pipi.org. "piudev". PyPI. <https://pypi.org/project/pyudev/> (accedido el 27 de diciembre de 2022).
- [11] raspberrypi. "Raspberry Pi OS (64-bit) - Raspberry Pi". Raspberry Pi. <https://www.raspberrypi.com/news/raspberry-pi-os-64-bit/> (accedido el 14 de diciembre de 2022).
- [12] Barr, M., & Massa, A. (2006). *Programming Embedded Systems: With C And Gnu Development Tools* (2.a ed.). Oreilly & Associates Inc.
- [13] Halfacree, G. (2018). THE OFFICIAL Raspberry Pi Beginner's Guide How to use your new computer. Phil King.
- [14] Naty, L. (2004). *Sistemas operativos*. Instituto Tecnológico Superior de Acayucan.

- [15] Pérez-Tavera, I. H. (2021). Raspberry Pi. Vida Científica Boletín Científico De La Escuela Preparatoria No. 4, 9(17), 40-41.
- [16] Centro Multimedia. (s. f.). Los diccionarios y las enciclopedias sobre el Académico
- [17] Estrada, F. (2014). Sistema multimedia Raspberry. Madrid
- [18] Horan, B. (2015). Practica Raspberry. Barcelona
- [19] Emmet, E. (2022, 18 agosto). Installing Widevine DRM to the Raspberry Pi. Pi My Life Up. Recuperado 16 de enero de 2023, de <https://pimylifeup.com/raspberry-pi-widevine/>
- [20] Python, R. (2022, 19 septiembre). Instalar PIL / Pillow y aplicar efectos visuales. Recursos Python. <https://recursospython.com/guias-y-manuales/instalar-pil-pillow-efectos/>
- [] Marín, R. (2020, 10 junio). ¿Qué es OpenCV? Instalación en Python y ejemplos básicos. Canal Informática y TICS. <https://www.inesem.es/revistadigital/informatica-y-tics/opencv/>
- [21] Rosebrock, A. (2021, 8 mayo). My imutils package: A series of OpenCV convenience functions. PyImageSearch. <https://pyimagesearch.com/2015/02/02/just-open-sourced-personal-imutils-package-series-opencv-convenience-functions/>
- [22] KeepCoding, R. (2022, 5 septiembre). ¿Qué es Tkinter? KeepCoding Tech School. <https://keepcoding.io/blog/que-es-tkinter/>
- [23] audioplayer. (2020, 1 julio). PyPI. <https://pypi.org/project/audioplayer/>
- [24] Python os Module. (s. f.). <https://www.tutorialsteacher.com/python/os-module>
- [25] subprocess.run. (2013, 16 noviembre). PyPI. <https://pypi.org/project/subprocess.run/>
- [26] Python Examples of signal.SIGTERM. (s. f.). <https://www.programcreek.com/python/example/385/signal.SIGTERM>
- [27] Schmidt, E. R. (s. f.). webbrowser. Ernesto Rico Schmidt. <https://rico-schmidt.name/pymotw-3/webbrowser/index.htm>
- [28] pyudev – pure Python libudev binding — pyudev 0.21.0 documentation. (s. f.). <https://pyudev.readthedocs.io/en/latest/>
- [29] ZoomAdmin.com. (s. f.). How to install python3-pyudev ubuntu package on Ubuntu 20.04/Ubuntu 18.04/Ubuntu 19.04/Ubuntu 16.04. <https://zoomadmin.com/HowToInstall/UbuntuPackage/python3-pyudev>