# Proyecto Final.
# *"Centro Multimedia"*

Hernández Alejo Ximena Gizell

29 de noviembre del 2025

## 1. Objetivo

Desarrollar un sistema multimedia para una Raspberry Pi que permitiera reproducir videos y música, leer documentos de una memoria USB y mostrar todo por medio de una interfaz gráfica sencilla e intuitiva.

## 2. Introducción

En este proyecto se desarrolló un sistema multimedia utilizando una Raspberry Pi, con la intención de crear una herramienta sencilla que permitiera reproducir videos, música y mostrar archivos desde una memoria USB. El sistema debía ser fácil de usar y tener una interfaz gráfica clara, pensada para usuarios con poca experiencia en tecnología. Mi participación en el proyecto se centró en diseñar la interfaz gráfica, programar el manejo mediante teclado y configurar el uso de un control Bluetooth para que el usuario pudiera navegar por el sistema sin necesidad de cables.

En este reporte se explica de manera sencilla cómo se construyó esta parte del proyecto y cómo se integra con el funcionamiento general del sistema para lograr una implementacion para usuarios ajenos al proyecto o conocimientos de la materia.

## 3. Antecedentes

Dado que nuestro objetivo es el poder entender y recrear este proyecto, primero es importante conocer algunas ideas básicas sobre los elementos que usamos. Esto ayudará a comprender mejor cómo funciona todo y por qué se hizo como se hizo.

### 3.1. Sistema embebido

Un sistema embebido es un sistema de cómputo diseñado para realizar una tarea específica dentro de un dispositivo. En el caso de este proyecto, la Raspberry Pi funciona como el sistema embebido que controla la interfaz gráfica, la reproducción de archivos y la comunicación con el control Bluetooth.

## 3.2. Raspberry Pi

La Raspberry Pi es una mini computadora del tamaño de una tarjeta. Aunque es pequeña, puede hacer muchas cosas: mostrar pantallas, reproducir videos, usar Bluetooth y ejecutar programas. Esto la hace perfecta para proyectos escolares y de prueba. [6]

## 3.3. Interfaz gráfica

Una interfaz gráfica (GUI) es la parte visual de un programa, donde el usuario ve botones, menús y texto. Es lo que hace que un sistema sea fácil de usar. En este proyecto usamos Pygame para diseñar la pantalla y los menús del sistema multimedia. [2]

Para optimizar el rendimiento, utilizamos Raspberry Pi OS Lite con una instalación mínima del X Window Server, instalando solo los componentes esenciales para mostrar gráficos sin el peso de un escritorio completo. [4]

## 3.4. USB y lectura

El sistema multimedia permite leer los archivos de una memoria USB. Para lograr esto, la Raspberry Pi debe detectar cuándo se conecta la memoria y acceder a los archivos dentro de ella. Esto se usa para mostrar listas de videos, música o documentos que el usuario puede abrir desde la interfaz.

Para hacer esto posible, usamos pyudev, que vigila los puertos USB. Cuando conectas tu memoria, pyudev lo detecta inmediatamente y el sistema automáticamente busca tus archivos de video, música y fotos para mostrarlos en pantalla. [8]

## 3.5. Bluetooth HID

Bluetooth HID es una tecnología especial que permite conectar dispositivos que funcionaran como un control inalámbrico. En este proyecto se usa para que el usuario pueda controlar la interfaz a distancia con un control inalámbrico. Funciona igual que un teclado pero sin necesidad de estar conectado. [1]

## 3.6. Reproducción de multimedia

El proyecto es capaz de reproducir videos y audio usando la librería python-vlc, que permite abrir archivos multimedia y controlarlos (pausa, play, detener). La interfaz gráfica envía las órdenes al reproductor para que el usuario pueda manejarlo de forma sencilla. [5]

## 3.7. Python como lenguaje de programación

El proyecto se desarrollo Python dado que así manejamos la interfaz, detectar el USB, controlar Bluetooth y reproducir archivos, todo dentro del mismo programa. [3]

## 3.8. WideVine DRM

Widevine DRM es la tecnología que usa Netflix, Disney+ y YouTube para proteger su contenido y que no se puedan hacer copias ilegales. En nuestro centro multimedia, Widevine permite que

puedas ver todas tus series favoritas en alta calidad directamente desde la Raspberry Pi, tal como lo harías en tu Smart TV o computadora normal. [7]

# 4. Materiales

Los materiales a usarse serán de fácil obtención, suponiendo que ya se cuenta con la Raspberry Pi y su respectivo sistema operativo.

- Equipo de computo con sistema operativo Linux derivado de Debian (Ubuntu, Mint, etc).
- Raspberry Pi de preferencia 4B.
- Tarjeta microSD clase 10 de almenos 16 GB con Raspberry Pi OS Lite.
- Pantalla HDMI o monitor VGA.
- Cable de corriente para la Rapberry Pi.
- Adaptador HDMI-MicroHDMI o VGA-MicroHDMI.
- Teclado USB provisional para la configuración inicial.
- Celular con app "Bluetooth Keyboard & mouse"que servira de control remoto.
- Memoria USB con fotos y videos.
- Adaptador USB-MicroSD

# 5. Funcionamiento y cuidado del Sistema

## 5.1. Funcionamiento de los componentes electrónicos

Aunque este proyecto no utiliza circuitos integrados adicionales o sensores externos, sí emplea componentes electrónicos básicos presentes en la Raspberry Pi. Aquí se describe su función general para que el lector entienda su importancia.

- Puertos USB: permiten conectar teclado, memoria USB u otros periféricos.
- Puerto microHDMI: envía la señal de video a la pantalla.
- Bluetooth HID: permite conectar un control inalámbrico.
- MicroSD: actúa como almacenamiento principal donde se instala el sistema operativo.
- Fuente de poder: alimenta la tarjeta para su funcionamiento estable.

Se recuerda que en este proyecto no se conectan circuitos integrados externos, ya que todo se maneja mediante software y periféricos estándar.

## 5.2. Funcionamiento de la tarjeta controladora

En este proyecto la Raspberry Pi funciona como el "cerebro" del proyecto. Al contar con un procesador, memoria RAM, puertos USB, Bluetooth y salida de video. La utilizamos para:

- Cargar el sistema operativo ligero (Raspberry Pi OS Lite).
- Administrar la interfaz gráfica del reproductor multimedia.
- Manejar las entradas mediante teclado para los primero pasos.
- Conectarse a un control Bluetooth para navegar por los menús.
- Ejecutar los programas que integran el sistema multimedia.

## 5.3. Cuidado de la salud y advertencias de riesgos

Aunque el proyecto es seguro, se deben recomienda considerar las siguientes precauciones:

- Evitar conectar o desconectar cables de energía con las manos mojadas.
- Evitar tocar la Raspberry Pi si está muy caliente después de un uso prolongado.
- Mantener ventilada la tarjeta para evitar sobrecalentamiento.
- No realizar modificaciones eléctricas mientras la tarjeta está encendida.

## 5.4. Cuidado de los componentes electrónicos delicados

Si bien en nuestro caso no tenemos componentes electrónicos muy delicados, para evitar daños en los componentes, se recomienda:

- No tocar los pines o conectores metálicos de la microSD y USB con los dedos.
- No tocar los pines o conectores metálicos de la microSD y USB con los dedos.
- Usar una carcasa o protector para la tarjeta.
- Evitar electricidad estática excesiva (frotar telas sintéticas o alfombras).

# 6. Configuración de la tarjeta controladora

Dado que la idea de este reporte-tutorial es que cualquier persona, aun con conocimientos básicos, pueda repetir la configuración sin perderse. Aunque todas las configuraciones fueron automatizadas mediante un script de inicialización incluido en el repositorio, se detallan las configuraciones más importantes.

## 6.1. Preparación del sistema operativo

Para el proyecto se utilizó Raspberry Pi OS Lite (64 bits) debido a que es una versión ligera, optimizada para proyectos embebidos y que permite instalar únicamente los componentes necesarios para la aplicación.

## 6.2. Preparación inicial

Después de instalar la imagen del sistema operativo en la tarjeta microSD, se deben realizar las configuraciones iniciales:

- *Creacion de usuario y contraseña.* Dado que versiones recientes del sistema ya no incluyen un usuario por defecto, se debe crear uno nuevo durante el primer inicio.
- *Distribución de teclado.* Se recomienda elegir la distribución correcta (por ejemplo, "es-mx") para evitar problemas al usar comandos.
- *Conexión de red (Wi-Fi o Ethernet).* Para simplificar la configuración se pueden usar herramientas como nmtui-edit y nmtui-connect.
- *Inicio automático de sesión.* Permite que el sistema cargue directamente la interfaz gráfica sin pedir usuario y contraseña en cada arranque.
- *Habilitar SSH (opcional).* En caso de que se desee hacer pruebas o depurar el programa de manera remota.

Para esto dentro de repositorio en la carpeta *config* se incluye un script que automatiza casi todo el proceso llamado *userconf.txt*. aqui tenemos 2 manera de configurar.

1. Conecta tu microSd a tu equipo de computo, al tener instalado el sistema operativo veras 2 particiones de almacenamiento *bootfs* y *rootfs*. Copia este archivo en la partecion **bootfs**. Esto crea utomáticamente al usuario del sistema sin necesidad de usar teclado en la primera ejecución.
2. Al grabar el sistema operativo en la microSD puedes usar el programa *Raspberry Pi Image*.

## 6.3. Configuración

Después de lograr la preparación inicial se recomienda el obtener el repositorio de manera local para poder acceder fácilmente a los archivos. Para esto se recomienda la instalación de Git

```
# Instalacion de Git
sudo apt install git
# Version instalada
git --version
# Copiar repositorio
git clone https://github.com/RodrigoRoku/FSE-ProyectoFinal.git
```

De este modo obtendrás todos los archivos del proyecto en la carpeta FSE-ProyectoFinal, para poder comprobarlo usar los siguientes comandos.

```
# Verificar copia del repositorio, ls lista los archivos y carpetas de la ubicaion actual
ls
# Debe aparecer la carpeta FSE-ProyectoFinal, para ingresar a las carpetas usar el comando cd
cd FSE-ProyectoFinal
```

Al ingresar a esta carpeta veras 3 carpetas *config, src* y *vid*, ingresa a la carpeta *config* en este encontraras el archivo *configure.sh*, ejecutalo.

```
sudo bash configure.sh
```

Si tienes curiosidad que hace este archivo, te explico:

- Asigna la zona horaria.

- Activa el SSH.
- Configura el país para Wi-Fi.
- Activa el autologin en consola.
- Verifica y establece la conectividad a Internet.
- Actualiza el sistema operativo.
- Instala los paquetes necesario para la función del proyecto que incluye el entorno gráfico y el reproductor multimedia.
- Instala y configura el Bluetooth.
- Crea la estructura de carpetas y copia de archivos.
- Configura el entorno gráfico personalizado
- Reinicia para poder aplicar las configuraciones

Para este caso al volver a encender la Raspberry se activar automáticamente la interfaz del centro multimedia, pero, ¿cómo se controla?, para esto desde tu equipo de computo te conectaras mediante SSH a la Raspberry.

```
# Consultar IP
# Opcion 1
ip addr show
# Opcion 2
hostname -I

# Usuario es el nombre de usuario, este deberia ser pi
# Caso de no conocer el nombre de usuario
whoami

# Conexion a SSH
ssh usuario@ip
```

De haber añadido una contraseña al usuario esta se te solicitara al hacer la conexión SSH, si usaste la configuración del documento *userconf.txt* sera **raspberry**.

En este momento harás uso de la aplicación "Bluetooth Keyboard & mouseütilizando un celular y una vez dentro, en la Raspebbry Pi ejecutar lo siguiente:

```
# Ingresar a la configuracion Bluetooth
bluetoothctl
# Configurar el bluetooth
power on
agent KeyboardOnly
default-agent
scan on
```

Es comando *scan on* iniciara la búsqueda de dispositivos cercanos estos tendrán su dirección MAC y el nombre del dispositivo busca el tuyo, al encontrarlo apaga la búsqueda.

```
# Apaga el scan
scan off
# Vincula tu telefono, si solicita un keypass dar yes desde la terminarl y vincular desde tu
    telefono.
pair XX:XX:XX:XX:XX:XX
# Conectar telefono (termial) alsalir el autentificador dar yes
connect XX:XX:XX:XX:XX:XX
# Permite que no tegas que hacer la cconexion cada que ingresas
trust XX:XX:XX:XX:XX:XX
```

Dentro la aplicación, ingresa al botón de *Nuevo dispositivo* y posteriormente a *Buscar dispositivo* y elige tu Raspberry, tendrá el nombre de usuario, automáticamente debería conectarse. Con esto la configuración quedaría completa.

# 7. Desarrollo de los componentes de software

En esta parte e describe cómo se desarrollaron los diferentes programas y módulos que permiten el funcionamiento del centro multimedia. Cada componente fue creado para cumplir una tarea específica, y al final todos se integraron para formar el sistema completo.

## 7.1. Interfaz gráfica

La interfaz gráfica fue desarrollada en Python utilizando la librería Pygame, ya que permite crear ventanas, dibujar elementos en pantalla y detectar entradas por teclado. En esta parte del software se incluyen los siguientes elementos:

- Pantalla principal en modo fullscreen, donde se muestran las secciones de botón de apagado, aplicaciones de streaming y reproductor multimedia desde USB.
- Cargas de imágenes (logos) para cada aplicación: Netflix, Disney+, Spotify y YouTube.
- Sistema de navegación por teclado usando flechas y Enter.
- Colores, tamaños de texto y distribución de elementos para hacer la interfaz más clara y fácil de usar.
- Ejecución de aplicaciones externas, como abrir el navegador en modo kiosco para acceder a los servicios de streaming.
- Reproductor multimedia desde USB

Todo esto está implementado en el archivo interfaz2.py, dentro de la clase graphicalInterface.

## 7.2. Módulo para leer carpetas y archivos desde USB

Para detectar automáticamente una memoria USB y revisar su contenido se creó el módulo usbManager.py, usando:

- La librería pyudev para detectar cuando se conecta o desconecta un dispositivo
- Comandos del sistema como udisksctl mount para montar la USB y findmnt para obtener la ruta del punto de montaje.
- El módulo analiza todos los archivos dentro de la memoria y los clasifica por tipo:
  - Fotos: jpg, png, jpeg, gif, avif
  - Videos: mp4, avi, wmv
  - Audio: mp3, wav, flac

Toda la información detectada se pasa a la interfaz para que pueda mostrarse al usuario.

## 7.3. Módulo para reproducir videos, fotos y música

La reproducción de contenido multimedia está a cargo del módulo mediaManager.py, que utiliza la librería python-vlc, la cual permite:

- Reproducir listas de archivos.
- Activar el modo pantalla completa.
- Para cada tipo de archivo se aplica un comportamiento diferente, por ejemplo:
  - Las fotos avanzan automáticamente cada 5 segundos.
  - Los videos y audio se reproducen de forma continua.

Este módulo se ejecuta como un subproceso, separado de la interfaz gráfica.

## 7.4. Control del teclado para navegar en la interfaz

El movimiento dentro de la interfaz se controla con las teclas:

- Flecha arriba/abajo → Cambiar de sección.
- Flecha izquierda/derecha → Moverse entre opciones.
- Tecla Enter → Activar la opción seleccionada.
- Esc → Cerrar aplicaciones externas o apagar el sistema.

Esto está implementado en la función navigation() dentro de interfaz2.py, lo que permite una interacción simple y sin necesidad de mouse.

# 8. Integración de los componentes de software

Al tener todos los módulos del sistema, se procede a su integración dentro de un programa principal que coordina todas las funcionalidades, este se creo el ejecutar el archivo *configure.sh*.

```
# Extracto de configure.sh
cat > /home/${USER}/.xinitrc << EOF
#!/bin/bash
sleep 1
...
```

Este programa es el encargado de ejecutar el main que ingresara a la interfaz gráfica, gestionar la detección automática de la memoria USB, cargar los archivos multimedia disponibles y permitir la navegación mediante teclado o dispositivos Bluetooth. Además, controla la reproducción del contenido seleccionado por el usuario.

# 9. Conclusiones

Este proyecto tenia como objetivo que que lográramos integrar diferentes componentes de hardware y software para construir un sistema embebido funcional y capaz ded simular un centro multimedia utilizando una Raspberry Pi como plataforma principal.A lo largo del proceso se logró configurar a su manera los módulos necesarios, como la detección de dispositivos externos, la interfaz gráfica, el control mediante Bluetooth HID y la reproducción de archivos como fotos, audio y video.

Su realización permitió entender como planificar la arquitectura del sistema, modularizar el código y verificar el funcionamiento de cada componente antes de la integración final. Además, se reforzaron conocimientos sobre la configuración de un sistema operativo, el manejo de periféricos y la optimización de recursos en entornos con capacidades limitadas.

# 10. Cuestionario

La siguiente sección contiene un conjunto de preguntas diseñadas para evaluar la comprensión del lector sobre los conceptos presentados a lo largo del desarrollo del proyecto. Estas preguntas abarcan temas como el funcionamiento de un sistema embebido, la configuración de periféricos, la integración del software y el uso de tecnologías como Bluetooth HID y el manejo de dispositivos USB. El objetivo es reforzar los conocimientos adquiridos y asegurar que el lector comprenda los fundamentos detrás de la construcción del centro multimedia.

1. *¿Qué función hace la Raspberry Pi en el desarrollo del centro multimedia?*
   Funciona como el controlador principal que ejecuta la interfaz, detecta dispositivos y reproduce contenido

2. *¿Qué ventajas ofrece utilizar Bluetooth HID frente a un control cableado?*
   Facilita la navegación, reduce errores y mejora la experiencia del usuario.

3. *¿Qué librería se utilizó para la reproducción de videos y audios, y por qué resulta adecuada para este tipo de proyectos*?
   Python-vlc, porque permite reproducir video, audio y listas de reproducción de manera simple.

4. *¿Qué realiza el script* `configure.sh` *al ser ejecutado?*
   El script configure.sh automatiza la configuración inicial del sistema.

5. *¿Qué aspectos se deben considerar para evitar daños o riesgos al manipular la Raspberry Pi y sus periféricos?*
   Evitar humedad, calor excesivo, tocar conectores, manipularla encendida o sin ventilación, y prevenir descargas estáticas.

# 11. Repositorio y Demostración

**Link al video demostración: Demostración video**

**Link al repositoriol: Repositorio**

# Referencias


[1] *Bluetooth HID Profile Specification*. Bluetooth SIG. 2020. URL: https://www.bluetooth.com/specifications/.

[2] Pygame Community. *Pygame Documentation*. 2023. URL: https://www.pygame.org/docs/.

[3] Python Software Foundation. *Python Language Reference*. https://www.python.org/. Accessed: 2024. 2024.

[4] Jasper St. Pierre. *X Window System Basics*. URL: https://magcius.github.io/xplain/article/x-basics.html.

[5] *python-vlc: VLC Media Player Python Bindings*. VideoLAN. 2023. URL: https://github.com/oaubert/python-vlc.

[6] *Raspberry Pi Documentation*. Raspberry Pi Foundation. 2024. URL: https://www.raspberrypi.com/documentation/.

[7] *Widevine DRM Overview*. Google LLC. 2023. URL: https://www.widevine.com/.

[8] Sebastian Wiesner. *pyudev: A libudev binding*. 2022. URL: https://pyudev.readthedocs.io/en/latest/.