

Centro Multimedia (Phoenix)

González Pacheco Leonardo Alonso.

Universidad Nacional Autónoma de México

Ciudad de México.

16 de agosto 2021

Objetivo

Desarrollo de centro de entretenimiento codificado en Python para la reproducción de películas, videos, música, y fotografías donde se visualizan diferentes servicios de streaming utilizando la Raspberry Pi 3 model b con sistema Operativo Raspbian 10 buster.

1- Lista de Materiales de componentes electrónicos, incluyendo posibles equivalentes si así fuere necesario.

Raspberry pi 3 model b.

Cable de red ethernet que facilita la conexión a internet.

Fuente de alimentación de 5V a 2.5A DC para la Raspberry.

Teclado alámbrico para conectar a través de un puerto USB.

Monitor con HDMI

Mouse USB

Memoria microSD de 32GB

2- Descripción del funcionamiento de los componentes electrónicos relevantes (circuitos integrados, encapsulados, controladores, etcétera)

Para tener un desempeño correcto de la Raspberry es importante tener en cuenta algunos puntos que ayudan al cuidado de los componentes electrónicos y la manera correcta de usarlos.

En la documentación oficial de se observa lo siguiente

Para conectarse a una toma de corriente, todos los modelos de Raspberry Pi tienen un puerto USB (el mismo que se encuentra en muchos teléfonos móviles): USB-C para Raspberry Pi 4 o micro USB para Raspberry Pi 3, 2 y 1. (Raspberry Pi, s.f.)¹

Este producto debe operarse en un ambiente bien ventilado y, si se usa dentro de una caja, la caja no debe estar cubierta. Mientras esté en uso, este producto debe colocarse sobre una superficie estable, plana y no conductora. (Raspberry Pi Foundation, s.f.)²

La conexión de dispositivos incompatibles a la conexión GPIO puede afectar el cumplimiento y provocar daños en la unidad e invalidar la garantía.

Todos los periféricos utilizados con este producto deben cumplir con los estándares relevantes para el país de uso y ser marcados en consecuencia para garantizar que se cumplan los requisitos de seguridad y rendimiento. Los cables y conectores de todos los periféricos deben tener un aislamiento adecuado para que se cumplan los requisitos de seguridad relevantes. (Raspberry Pi Foundation, s.f.)²

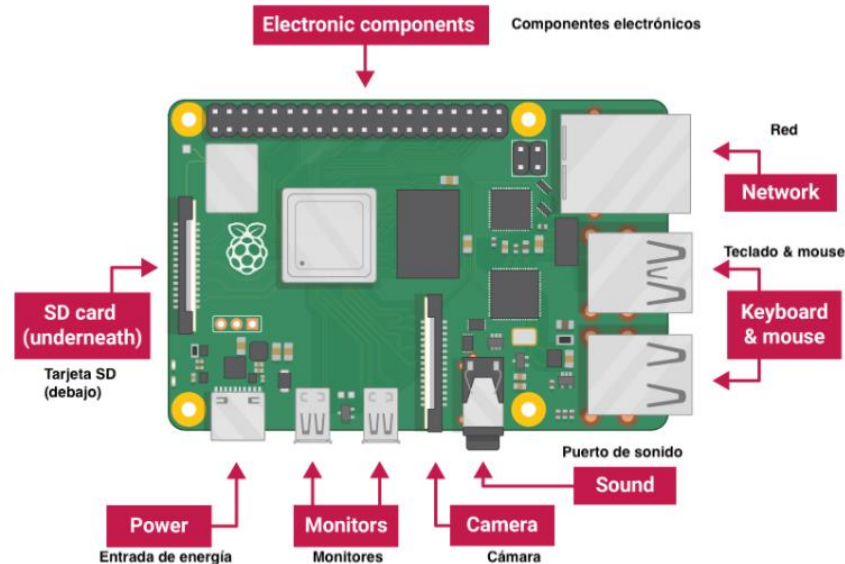


Figura 1: Componentes de la Raspberry

Imagen obtenida de: (2017). Retrieved August 16, 2021, from Raspberrypi.org website:
<https://projects.raspberrypi.org/es-LA/projects/raspberry-pi-setting-up/3>

En el esquema se pueden observar los componentes de la tarjeta; cada uno de ellos cumple con el objetivo de poder interactuar con entradas y salidas para el desarrollo de diferentes proyectos.

Descripción del funcionamiento de la tarjeta controladora (ej. Raspberry Pi) o circuito microcontrolador (ej. Microchip PIC) a utilizar, así como de los componentes relevantes de los mismos.

Para el funcionamiento de la tarjeta es necesario conectar dispositivos de entrada y salida a los puertos incluidos en la Raspberry porque facilitan la configuración e instalación del sistema operativo y ayudan a poner en marcha el proyecto que se quiera realizar. Es interesante la manera en la que interactúan cada una de las partes porque al ser un ordenador pequeño su aplicación para diferentes necesidades es muy amplia y práctica.

A continuación, se listan sus componentes y especificaciones obtenidas de la página oficial de Raspberry pi:

- CPU de 64 bits Broadcom BCM2837 de cuatro núcleos a 1,2 GHz.
- 1 GB de RAM.
- BCM43438 LAN inalámbrica y Bluetooth de baja energía (BLE) a bordo.
- 100 Ethernet base.
- GPIO extendido de 40 pines.
- 4 puertos USB 2.
- Salida estéreo de 4 polos y puerto de video compuesto.
- HDMI de tamaño completo.
- Puerto de cámara CSI para conectar una cámara Raspberry Pi.
- Puerto de pantalla DSI para conectar una pantalla táctil Raspberry Pi.
- Puerto microSD para cargar su sistema operativo y almacenar datos.
- Fuente de alimentación micro USB conmutada mejorada de hasta 2,5 A.

(Raspberry pi, s.f.)³

3- Información sobre el cuidado de la salud y advertencias de riesgos (ej. quemaduras, electrocución, etcétera)

En forma general solo es cuidar la fuente de alimentación que está conectada a las Raspberry esté regulada a no más de 5V a 3A, con el fin de no hacer que se queme la tarjeta Raspberry y evitar quemaduras físicas.

Si bien los disipadores de calor no son necesarios para evitar daños por sobrecalentamiento en el SoC, el mecanismo de estrangulamiento térmico se encarga de eso, un disipador de calor o un pequeño ventilador le ayudarán si desea reducir la cantidad de estrangulamiento térmico que tiene el lugar. Dependiendo de las circunstancias exactas, montar el Pi verticalmente también puede ayudar con la disipación del calor, ya que hacerlo puede mejorar el flujo de aire. (Raspberry Pi, s.f.)⁴

4- Información sobre el cuidado de los componentes electrónicos delicados (ej. memorias) que así lo requieran, tales como uso de pulsera antiestática, acumulación de grasa en terminales, límites de temperatura, etcétera.

Las Raspberry Pi con bolsa antiestática: Cuenta con una bolsa antiestática cuya función es reducir la estática (acumulación de carga eléctrica) en la tarjeta debido a que puede ocasionar daños en los componentes.

MicroSD: Como el dispositivo es muy pequeño tener precaución de no extraviarlo.

Adaptador microSD: Conseguir un adaptador ajustable a la microSD, además algunos adaptadores cuentan con un bloqueo el cual debe estar desactivado a la hora de conectarse.

Fuente de alimentación: Cuidar que la fuente de alimentación esté regulada de 5V a 3A o puede ser menos para no quemar la Raspberry Pi.

Cable de red ethernet: Cuidar que no se enrolle dicho cable.

Caja protectora: Para evitar el maltrato de la tarjeta se pueden adquirir aditamentos especiales como lo es una caja protectora que evita accidentes e incrementa la durabilidad para la Raspberry.

Hay varios problemas conocidos en el modo de inicio Ethernet. Dado que la implementación de los modos de arranque está en el propio chip, no hay otras soluciones que no sean usar tarjeta SD con solo el archivo bootcode.bn. (Raspberry Pi, s.f.)⁴

Toda la electrónica de almohadillas está diseñada para 16 mA. Ese es un valor seguro por debajo del cual no dañará el dispositivo. Incluso si configura la fuerza de la unidad en 2mA y luego la carga para que salgan 16 mA, esto no dañará el dispositivo. Aparte de eso coma no hay una corriente segura máxima garantizada. (Raspberry Pi, s.f.)⁴

5- Configuración de la tarjeta controladora (ej. Raspberry Pi) o circuito microcontrolador (ej. Microchip PIC a utilizar, incluyendo puertos, periféricos, configuraciones del sistema operativo, paquetes y librerías a instalar, etcétera.

Al utilizarse la Raspberry Pi 3 model b de forma física fue necesario contar con un cable ethernet, una fuente de alimentación y una microSD donde con ayuda de un adaptador se conectó a la computadora para guardar la distribución del sistema operativo a utilizar; el seleccionado fue:

Raspbian 10 buster obtenida de la página oficial:

<https://www.raspberrypi.org/software/operating-systems/>

Se instaló el Sistema Operativo en la SD usando el siguiente programa

<https://www.balena.io/etcher/>

ya que esto nos ayuda a bootear la memoria lo que significa que se puede llevar a cabo el arranque de la Raspberry a través de la USB.

Para realizar las configuraciones contar con un teclado y mouse que se coloque en la tarjeta facilita el proceso de instalación del sistema operativo

Sino se cuenta con ellos es necesario tener una conexión de manera remota a la Raspberry esto se hace de la siguiente manera:

1- Conectar la fuente de alimentación a la Raspberry Pi para que pueda encender y un cable ethernet del modem a la tarjeta.

2- Instalar nmap en la computadora para poder rastrear los puertos que se encuentran abiertos y ver las direcciones ip que se están conectando a nuestra red.

enlace para la descarga del programa: <https://nmap.org/download.html>

3- Instalar Putty en la computadora ya que es un programa que permite conectarnos a través de un canal seguro SSH y ayuda a poder ingresar de manera remota a la terminal del sistema operativo instalado en la Raspberry. Enlace para la descarga del programa: <https://www.putty.org/>

4- Instalación de VNC Viewer. Ayuda a mostrarnos de manera gráfica el Sistema Operativo que corre en la tarjeta. Enlace para la descarga: <https://www.realvnc.com/es/connect/download/viewer/>

5- Una vez instalado nmap se escanea el segmento de red donde se conectan nuestros dispositivos al modem de internet para visualizar con cuál ip está teniendo comunicación la Raspberry pi.

Comando ingresado en cmd con segmento de red

```
C:\Users\Temp>nmap -sn 192.168.0.0/24 -oG -
# Nmap 7.92 scan initiated Sun Aug 15 11:11:21 2021 as: nmap -sn -oG - 192.168.0.0/24
Host: 192.168.0.1 () Status: Up
Host: 192.168.0.6 () Status: Up
Host: 192.168.0.12 () Status: Up
Host: 192.168.0.253 () Status: Up
Host: 192.168.0.10 () Status: Up
# Nmap done at Sun Aug 15 11:11:59 2021 -- 256 IP addresses (5 hosts up) scanned in 38.27 seconds
```

Figura 2: ips conectadas al modem

ip identificada con la que la Raspberry se conecta = 192.168.0.12

6- Con el programa Putty se establece una comunicación por medio de un canal SSH

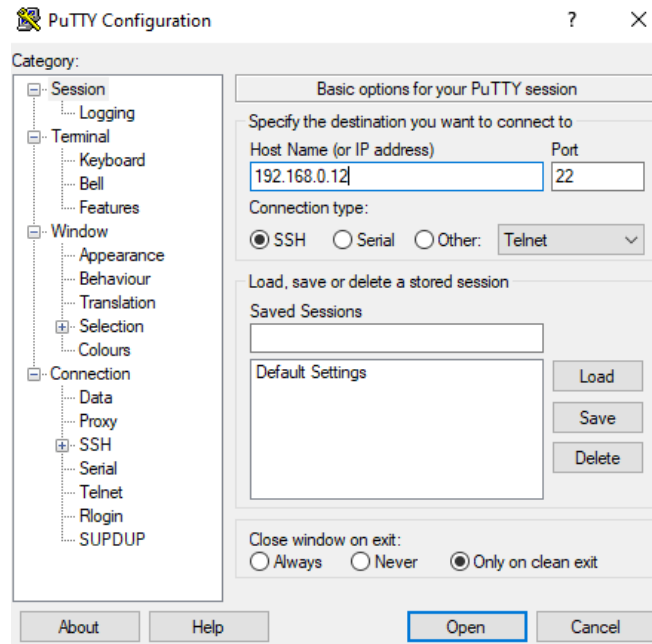


Figura 3: Acceso al canal seguro SSH con Putty

Aparecerá una terminal donde se debe poner el usuario y contraseña para acceder al Sistema operativo.

Los que están por defecto en Raspbian son:

Usuario: pi

Password: raspberry

7- Al ingresar a la terminal de la Raspberry se instala la siguiente paquetería.
apt-get install screenfetch para visualizar la información del Sistema Operativo.


```

pi@raspberrypi ~$ sudo fdisk /dev/mmcblk0
Command (m for help): Partition type
  p  primary (1 primary, 0 extended, 3 free)
  e  extended (container for logical partitions)
Select (default p): Partition number (2-4, default 2): First sector (2048-606207
99, default 2048): Last sector, +/-sectors or +/-size(K,M,G,T,P) (532480-6062079
9, default 60620799):
Created a new partition 2 of type 'Linux' and of size 28.7 GiB.
Partition #2 contains a ext4 signature.

Command (m for help):
Disk /dev/mmcblk0: 28.9 GiB, 31037849600 bytes, 60620800 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x97709164

Device      Boot  Start      End  Sectors  Size Id Type
/dev/mmcblk0p1      8192    532479    524288   256M c W95 FAT32 (LBA)
/dev/mmcblk0p2    532480 60620799 60088320  28.7G 83 Linux

Command (m for help): The partition table has been altered.
Syncing disks.

pi@raspberrypi ~$ sudo apt-get install -y realvnc-vnc-server realvnc-vnc-viewer
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
E: No se ha podido localizar el paquete realvnc-vnc-server
pi@raspberrypi ~$ sudo apt-get install -y realvnc-vnc-server realvnc-vnc-viewer
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
El paquete indicado a continuación se instaló de forma automática y ya no es nec
esario.
  libmicrodnso
Utilice «sudo apt autoremove» para eliminarlo.
Paquetes recomendados:
  cups
Se actualizarán los siguientes paquetes:
  realvnc-vnc-server realvnc-vnc-viewer
2 actualizados, 0 nuevos se instalarán, 0 para eliminar y 189 no actualizados.
Se necesita descargar 90585 kB de archivos.
Se utilizarán 12.3 kB de espacio de disco adicional después de esta operación.

```

Figura 6: Instalación de VNC viewer

10- se corre el comando vncserver para iniciar VNC

```

pi@raspberrypi ~$ vncserver
Mark or unmark the service to be started at boot time with:
systemctl (enable|disable) vncserver-x11-serviced.service

Installed systemd unit for VNC Server in Virtual Mode daemon
Start or stop the service with:
systemctl (start|stop) vncserver-virtuald.service
Mark or unmark the service to be started at boot time with:
systemctl (enable|disable) vncserver-virtuald.service

Procesando disparadores para mime-support (3.62) ...
Procesando disparadores para hicolor-icon-theme (0.17-2) ...
Procesando disparadores para gnome-menus (3.31.4-3) ...
Procesando disparadores para man-db (2.8.5-2) ...
Procesando disparadores para shared-mime-info (1.10-1) ...
Procesando disparadores para desktop-file-utils (0.23-4) ...
pi@raspberrypi ~$ vncserver
VNC(R) Server 6.7.2 (r42622) ARMv6 (May 13 2020 19:34:20)
Copyright (C) 2002-2020 RealVNC Ltd.
RealVNC y VNC son marcas comerciales de RealVNC Ltd y están protegidas por los
registros de marcas y/o por solicitudes de registro en trámite en la Unión
Europea, los EE. UU. de América y otras jurisdicciones.
Protegido por la patente del Reino Unido 2481870, la patente de Estados Unidos
8760366 y la patente de la UE 2652951.
Consulte https://www.realvnc.com para obtener información sobre VNC.
Para ver los reconocimientos de terceros, consulte:
https://www.realvnc.com/docs/6/foss.html
OS: Raspbian GNU/Linux 10, Linux 4.19.97, armv7l

Generando clave privada... Terminado
En algunas distribuciones (concretamente en Red Hat), la experiencia puede
mejorar si se ejecuta vncserver-virtual conjuntamente con el servidor Xorg del
sistema, en lugar de la versión anterior integrada en Xvnc. Probablemente sean
compatibles más aplicaciones y entornos de escritorio. Si desea más información
acerca de esta implementación alternativa, consulte:
https://www.realvnc.com/doclink/kb-546

Aplicaciones en ejecución en /etc/vnc/xstartup

Lema de VNC Server: "Harris unicorn Pegasus. Stamp heroic Austin."
firma: a6-00-cb-56-4c-96-4c-c7

El archivo de registro es /home/pi/.vnc/raspberrypi:1.log
El nuevo escritorio es raspberrypi:1 (192.168.0.12:1)
pi@raspberrypi ~$

```

Figura 7: Se inicia vncserver

11-Se abre el programa VNC para establecer una nueva conexión.

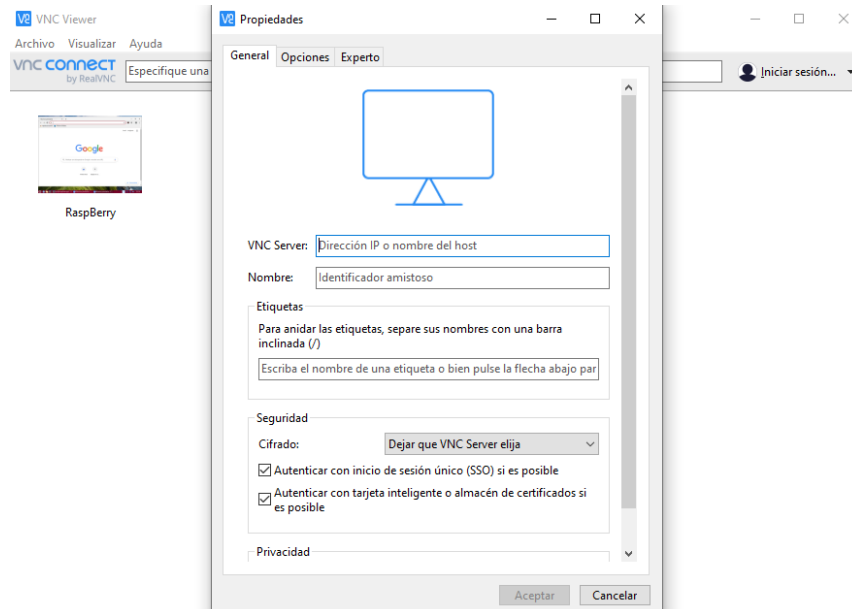


Figura 8: se coloca ip y nombre de conexión

Se coloca la dirección ip y el nombre que escojamos para la conexión.

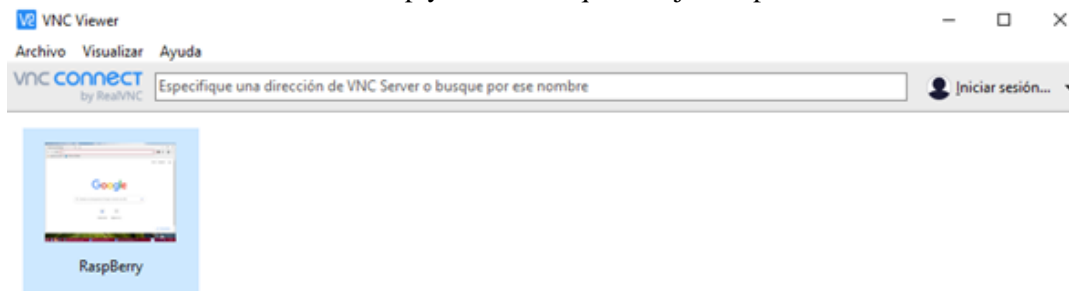


Figura 9: Se visualiza la interfaz

Para el desarrollo del centro multimedia se instaló python3 y se ejecutaron los siguientes comandos para instalar paquetes que permitieran usar el reproductor multimedia vlc

```
sudo apt-get install python3
```

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

Se clonó en repositorio donde está nuestro código para poderlo ejecutar con Python desde la terminal de la Raspberry

<https://github.com/LeoAlonsoGonzalez/CentroMultimediaPhoenix>

6- Desarrollo de los componentes de software (módulos) para cada una de las funciones que desempeñará la tarjeta

Se codificaron en Python las funcionalidades para obtener el diseño completo del proyecto y se utilizaron las siguientes librerías:

Instalación de pygame para controlar la clase de reproducción de archivos de audio.

sudo pip install pygame.

La instalación de librería vlc Nos facilitó reproducir el contenido con el reproductor VLC.
pip3 install python-vlc Con vlc.Media player(“archivo”).

Instalación de librería tkinter para que sea amigable la interfaz de usuario; tkinter nos permite crear botones, labels y ventanas.

```
1 """
2 #####
3 Proyecto Final: Media Center (PHOENIX)
4 Fundamentos de Sistemas Embebidos
5 Semestre 2021-2
6 Creación: 16/08/2021
7 Última modificación: 17/08/2021
8
9 Integrantes:González Pacheco Leonardo Alonso
10             Hernández Hernández Edgar
11             Ruiz Aguilar Eduardo
12 #####
13 """
14
15 import pygame
16 import threading
17 import vlc
18 import os
19 from tkinter import *
20 from PIL import ImageTk, Image
```

Figura 10: Se muestran las librerías en código.

```
52 class Ventana:
53     #Constructor de la ventana
54     def __init__(self,ventana):
55         self.ventana=ventana
56         #valor mandado por el constructor, que es un
57         #objeto Tk, el cual permite crear ventanas con
58         #diferentes objetos dentro de ella
59
60         self.opcion = 0
61         ventana['background']='#000000'
62         ventana.title("Reproductor Phoenix")#Ponemos el titulo a la ventana
63         self.imagen = ImageTk.PhotoImage(Image.open(r'/home/pi/Documents/CentroMultimedia/Imagenes/netflix.png').resize(
64         self.boton = Button(ventana, image = self.imagen, width= 100, height=100,relief="flat", borderwidth=0, bg='#00
65
66         self.imagen1 = ImageTk.PhotoImage(Image.open(r'/home/pi/Documents/CentroMultimedia/Imagenes/Spotify.png').resize(
67         self.boton1 = Button(ventana, image = self.imagen1, width= 100, height=100,relief="flat", borderwidth=0, bg='#f
68
69         self.imagen2 = ImageTk.PhotoImage(Image.open(r'/home/pi/Documents/CentroMultimedia/Imagenes/Amazon.png').resize(
70         self.boton2 = Button(ventana, image = self.imagen2, width= 100, height=100,relief="flat", borderwidth=0, bg='#f
71
72         self.imagen3 = ImageTk.PhotoImage(Image.open(r'/home/pi/Documents/CentroMultimedia/Imagenes/Disney.png').resize(
```

Figura 11: Codificación para la creación de botones.

```

74     self.imagen4 = ImageTk.PhotoImage(Image.open(r'/home/pi/Documents/CentroMultimedia/Imagenes/USB.png')).resize((
75     self.boton4 = Button(ventana, image = self.imagen4, width= 200, height=200,relief="flat", borderwidth=0, bg='#
76
77     self.imagen5 = ImageTk.PhotoImage(Image.open(r'/home/pi/Documents/CentroMultimedia/Imagenes/exit.png')).resize((
78     self.boton5 = Button(ventana, image = self.imagen5, width= 100, height=100,relief="flat", borderwidth=0, bg='#
79     """En estas sección se muestran los métodos que se activan al presionar los
80     botones"""
81     def netflix(self):
82         self.opcion = 1
83         self.ventana.destroy()
84     def spotify(self):
85         self.opcion = 2
86         self.ventana.destroy()
87     def primeVideo(self):
88         self.opcion = 3
89         self.ventana.destroy()
90     def disney(self):
91         self.opcion = 4
92         self.ventana.destroy()
93     def memoria(self):

```

Figura 12: Métodos para el funcionamiento de los botones.

```

102     memoria USB"""
103     class Ventana2:
104         #Constructor de la segunda ventana
105         def __init__(self, ventana, opcion):
106             self.opcion=opcion
107             self.ventana=ventana
108             ventana.title("Interfaz Memoria")
109             ventana['background']='#000000'
110
111             ext_audio=[".mp3"]
112             musica=encontrar(ext_audio)
113
114             ext_video=[".mp4",".mkv"]
115             videos=encontrar(ext_video)
116
117             ext_imagenes=[".jpg",".png",".jfif",".jpeg"]
118             fotos=encontrar(ext_imagenes)
119
120             if(len(musica)==0 and len(videos)==0 and len(fotos)==0):
121                 self.lab=Label(ventana, bg="black", fg="white", font=('Arial',15),text="No hay contenido").place(x=570,y=1

```

Figura 13: Tipos de archivo que se leen en la memoria al insertarla.

```

422     if len(arr)!=0 and len(arr2)==0 and len(arr3)==0:
423         a=1
424     elif len(arr)==0 and len(arr2)!=0 and len(arr3)==0:
425         a=3
426     elif len(arr)==0 and len(arr2)==0 and len(arr3)!=0:
427         a=2
428         '''En dado que no sea el caso, se debe de lanzar la segunda
429         ventana para elegir que archivo reproducirá'''
430     else:
431         root=Tk()
432         a=0
433         root.attributes("-zoomed",True)
434         vent=Ventana2(root,a)
435
436         root.mainloop()
437         a=vent.regresa()#Este valor determina que archivo se
438                             #reproducirá
439         '''a==1 es audio'''
440     if a==1:
441         ent=pygame.mixer.init()

```

Figura 14: Librería pygame y método para reproducir de audio.

```

475 elif(valor==1):
476     comando="/usr/bin/chromium-browser %s"
477     nav=webbrowser.get(comando)#Se utiliza el método get para obtener el navegador
478     webbrowser.register("chrome",None,nav)#Con este método se registra el navegador Chrome
479                                     #con el nombre de chrome
480     nav=webbrowser.get("chrome")#Se obtiene el navegador para utilizarlo
481     nav.open("https://www.netflix.com/mx/login")
482
483 elif(valor==2):
484     comando="/usr/bin/chromium-browser %s"
485     nav=webbrowser.get(comando)
486     webbrowser.register("chrome",None,nav)
487     nav=webbrowser.get("chrome")
488     nav.open("https://accounts.spotify.com/es/login/?continue=https:%2F%2Fwww.spotify.com%2Fapi%2Fgrowth%2F12l-redi")
489
490 elif(valor==3):
491     comando="/usr/bin/chromium-browser %s"
492     nav=webbrowser.get(comando)
493     webbrowser.register("chrome",None,nav)
494     nav=webbrowser.get("chrome")

```

Figura 15: Implementación de los botones para que nos dirijan a las plataformas de entretenimiento.

```

503
504 else:
505     bandera=False#Para terminar con el ciclo while

```

Figura 16: Termina ejecución del programa.

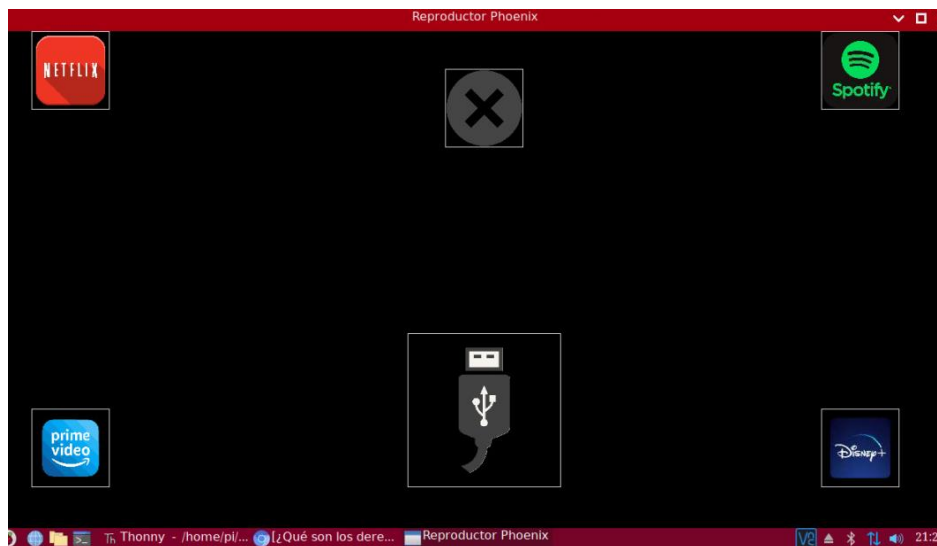
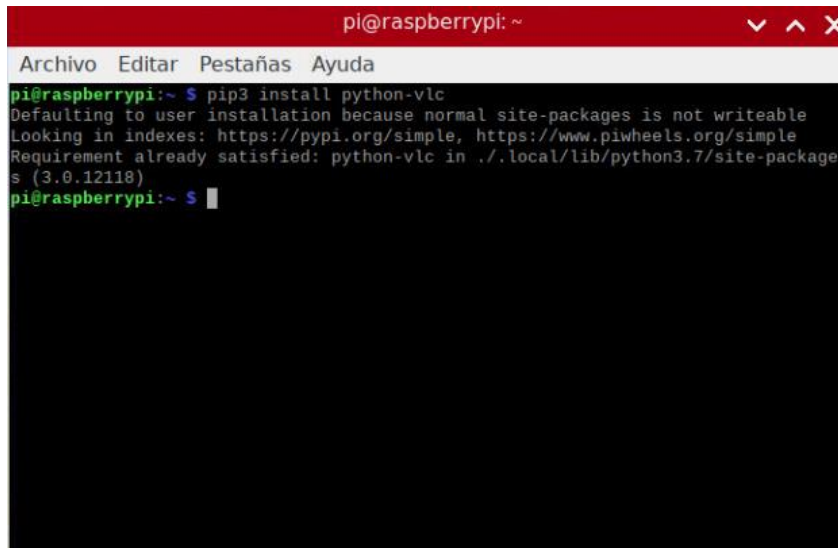


Figura 17: Interfaz creada con tkinter.

En la configuración de los componentes se usó el módulo OS que despliega directorios para observar lo que contiene la memoria USB

Librería webbrowser: Facilita la conexión a un navegador para hacer búsquedas en internet y dirigir al presionar un botón a las páginas de video, música y entretenimiento.



```
pi@raspberrypi: ~  
Archivo Editar Pestañas Ayuda  
pi@raspberrypi:~$ pip3 install python-vlc  
Defaulting to user installation because normal site-packages is not writeable  
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple  
Requirement already satisfied: python-vlc in ./local/lib/python3.7/site-packages (3.0.12118)  
pi@raspberrypi:~$
```

Figura 18: Instalación de vlc en terminal de Raspberry.

La captura nos muestra que ya se había instalado anteriormente, aquí se ejemplifico el comando correcto que se debe digitar.

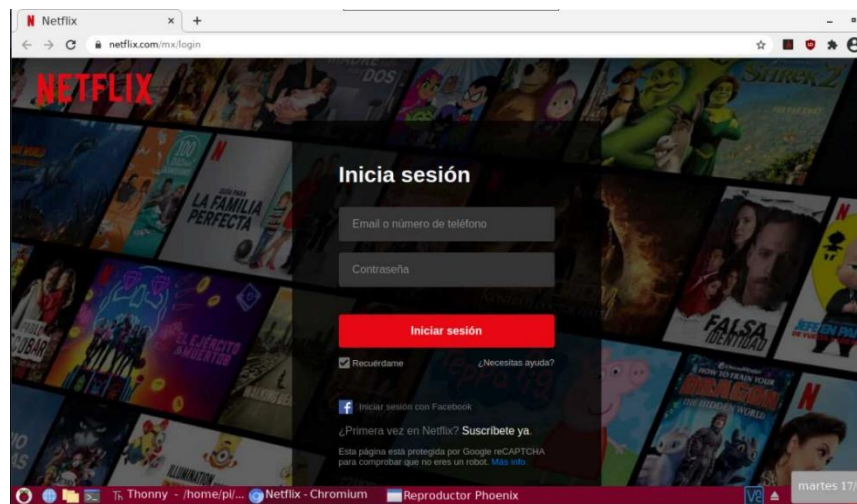


Figura 19: Direccionamiento de botones al navegador gracias a la librería webbrowser.

7- Integración de los componentes de software o módulos en una solución de software

Para la integración de los componentes no se modularizo el código debido a que fue más fácil utilizar las librerías para la construcción de código. La compatibilidad del sistema Operativo ocasionaba algunos errores y de esta manera se prefirió realizar en un solo archivo toda la implementación.

El control de versiones utilizado facilito el manejo de las instrucciones programadas dentro de la Raspberry.

En lo que respecta a los módulos de hardware utilizados se emplearon los puertos USB para leer la memoria de la Raspberry, la entrada microSD para almacenar el sistema operativo y la entrada de ethernet para tener acceso a internet lo que ayudo a la integración de las funcionalidades y comunicación entre Software y los componentes para leer archivos multimedia.

La interfaz contiene diferentes botones para ir a Netflix, Amazon, Disney, Spotify, leer contenido de la memoria. Con el botón exit se utiliza `media.Stop()` que es un método.

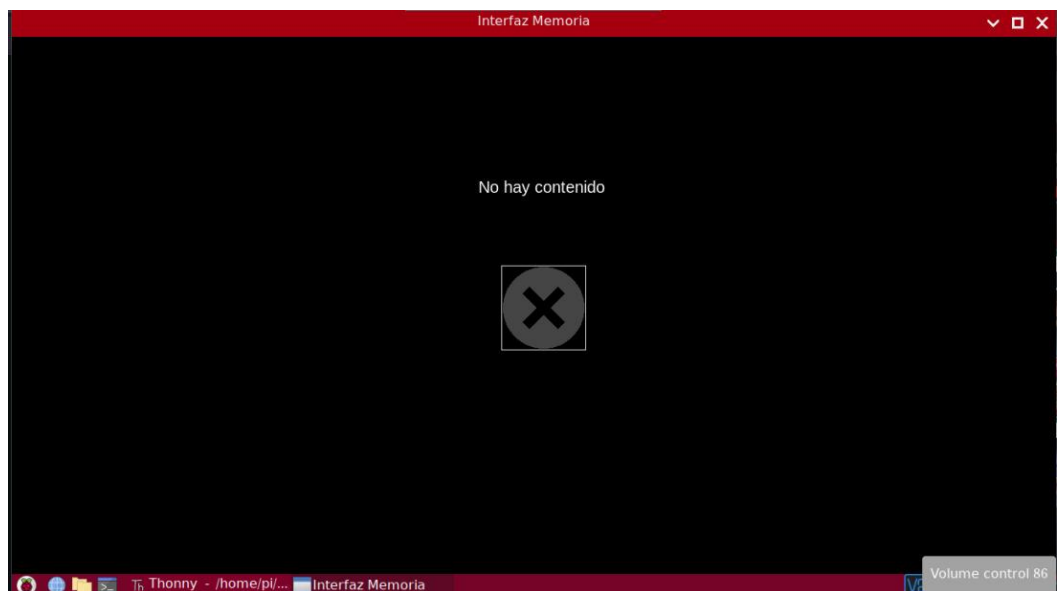


Figura 20: Pantalla que sale al insertar memoria y no tiene contenido.

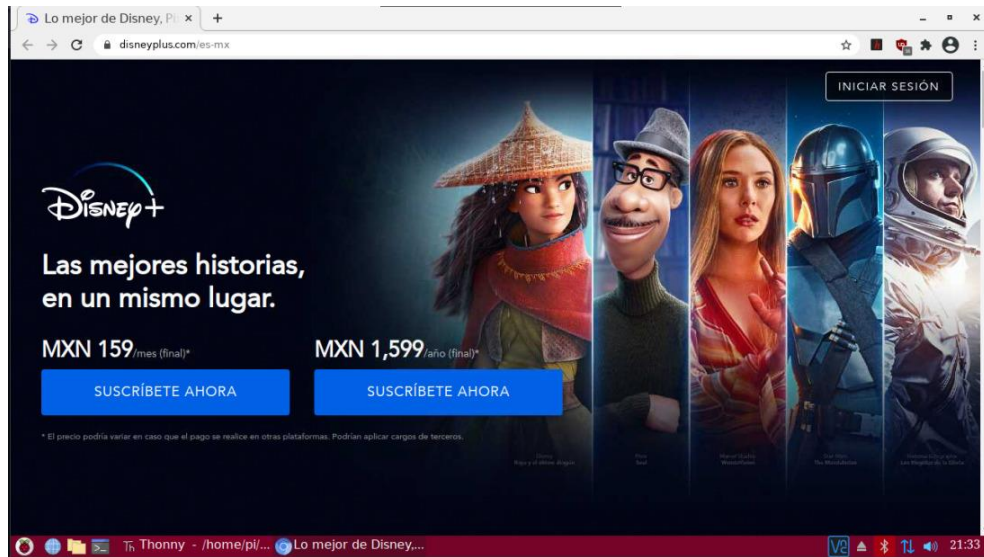


Figura 21: Botón que direcciona a Disney.

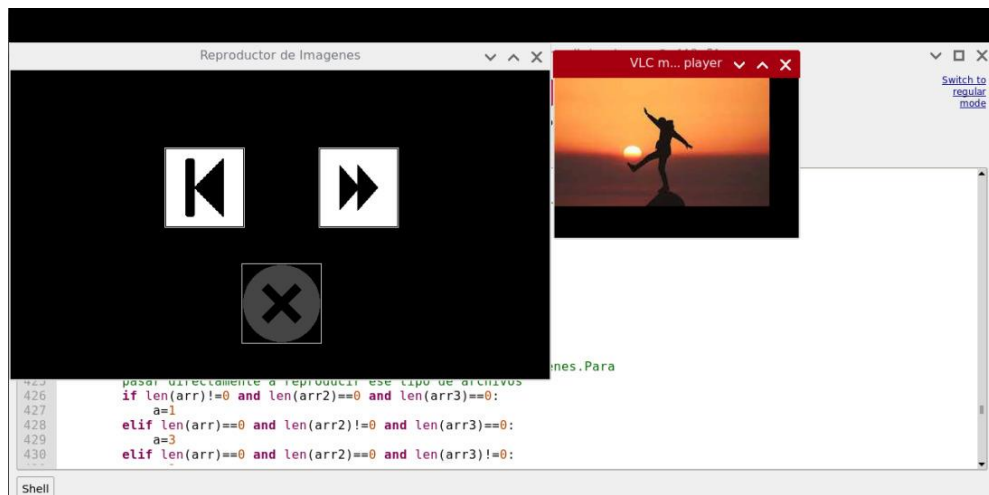


Figura 22: Memoria insertada en el puerto USB donde se pueden visualizar los archivos multimedia y adelantar o atrasar su contenido.

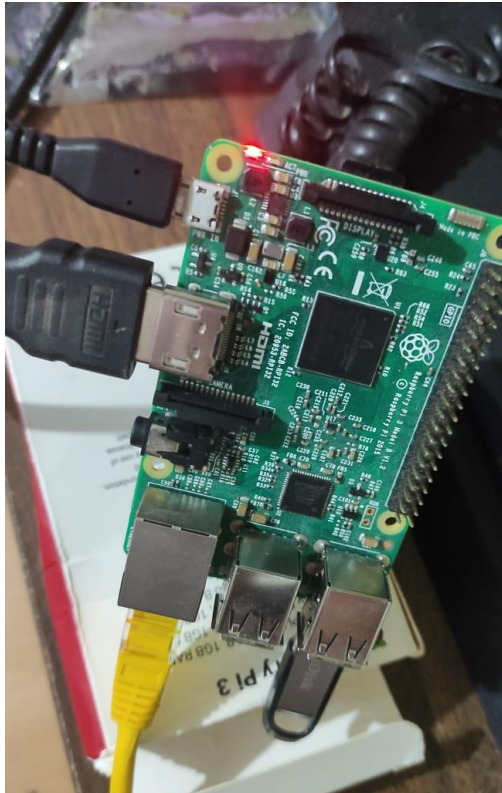


Figura 23: Conexiones de Raspberry al modem de internet, al monitor y memoria conectada al puerto USB.



Figura 24: Visualización de la conexión de la Raspberry.



Figura 25: Reproductor multimedia mostrado en monitor.

8- Conclusiones

Hubo complicaciones en cuanto a la compatibilidad de los recursos utilizados para poder generar una experiencia adecuada en el usuario; la versión de Python y las diferentes librerías descargadas genero algunos errores para mostrar adecuadamente los botones u observar los videos e imágenes sin embargo se soluciono al encontrar la documentación adecuada. El direccionamiento de los botones a las plataformas de streaming dio una solución eficiente para hacer que el reproductor multimedia fuera un proyecto más robusto. La interfaz y su forma de trabajar con los procesos programados cumplió el objetivo planteado ya que se logro entregar al cliente una forma de entretenerse amigable y no complicada.

Fue un reto lograr que se leyera correctamente la memoria con el puerto USB, pero Python facilita mucho las cosas porque un problema que parece complejo lo simplifica con sus librerías y manera sencilla de programar.

Las ventajas de tener un espacio de trabajo como lo es el de la Raspberry es su bajo costo y la enorme cantidad de casos prácticos que se pueden desarrollar para entretener, educar o generar dinero a través de proyectos.

9- Cuestionario

¿Con que librería se realizaron los botones para la interfaz gráfica?

Existen diferentes librerías que nos pueden ayudar a lograr este objetivo, pero Tkinter es sencilla de usar y su documentación es amplia.

¿Cómo se pausa un archivo de audio mientras se está reproduciendo?

Con `pygame.mixer.music.pause()`

¿Por qué se decidió crear el reproductor en una Raspberry física?

Porque la implementación se nos hizo interesante y para la configuración consideramos que traería menos problemas la importación de librerías específicas que agilizan el trabajo.

¿Qué herramienta fue necesaria para reproducir el video y cuál es su librería en Python?

Para la reproducción de video se tuvo que utilizar vlc y la librería que nos ayudó al funcionamiento del programa fue Python-vlc.

Bibliografía

¹Raspberry Pi. (s.f.). *Configuración de tu Raspberry Pi*. Recuperado el 16 de agosto de 2021, de <https://projects.raspberrypi.org/es-LA/projects/raspberry-pi-setting-up/1>

²Raspberry Pi Foundation. (s.f.). *Raspberry Overview*. Recuperado el 16 de agosto de 2021, de raspberry: <https://datasheets.raspberrypi.org/rpi3/raspberry-pi-3-b-plus-product-brief.pdf>

³Raspberry pi. (s.f.). *Raspberry Pi 3 Modelo B*. Recuperado el 16 de agosto de 2021, de <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>

⁴Raspberry Pi. (s.f.). *Documentación de Raspberry Pi*. Recuperado el 16 de agosto de 2021, de Hardware de Raspberry Pi: <https://www.raspberrypi.org/documentation/computers/raspberry-pi.html>

Figura 1: (2017). Retrieved August 16, 2021, from Raspberrypi.org website: <https://projects.raspberrypi.org/es-LA/projects/raspberry-pi-setting-up/3>