

Proyecto final: Centro multimedia.

Objetivo

- El alumno hará uso de los conocimientos adquiridos durante el curso en proyecto que implica la realización de un centro del multimedia, probará que ha adquirido conocimientos robustos sobre cómo operan los sistemas embebidos, así como de sus características y limitaciones.
- Aprender a utilizar la tarjeta Raspberry Pi de forma física o simulada.

Material

Los materiales necesarios para el desarrollo del proyecto son:

- Una computadora personal con 8 GB de RAM, un procesador Intel Core i3, AMD-Ryzen 3 o superior.
- El espacio mínimo que se destinará a la imagen que correrá dentro de nuestra máquina virtual será de 30 GB y 2 GB de memoria RAM en adelante.
- La imagen del sistema operativo (.iso) Raspbian.
- El software de Oracle VM VirtualBox.
- Descarga de python3 y VLC.

¿Qué es la Raspberry Pi?

Se trata de un ordenador muy pequeño, similar a una tarjeta de crédito y consta de varios elementos. Tiene una base que es una placa y sobre esta es montado un procesador, una memoria RAM, y un chip gráfico. Este aparato tecnológico fue creado en el año 2006 para la enseñanza de informática en las escuelas. Hoy en día existe una página llamada www.raspberrypi.org, en la cual enseñan a la programación de este desde cero, tienen foros y su comunidad de desarrolladores. (Vargas Daniel, 2019, p.2)

La raspberry pi tiene un software de open source, que significa código abierto esto se puede modificar por parte de un programador, puede trabajarse en cualquier sistema operativo incluso en la última versión de Windows 10, las raspberry pi tienen una memoria RAM, una GPU (unidad de procesamiento gráfico), procesador Broadcom, puertos USB, HDMI, Ethernet, 40 pines GPIO (pines de entrada y salida), estos se pueden controlar por el usuario en tiempo de ejecución y un conector para cámara, lo único que no tiene es cable de alimentación ni el disco duro para ello se reemplaza por una memoria USB. (Castro, 2014).

Funcionamiento de la Raspberry Pi

Dada la situación actual derivada de la pandemia, no se trabajó con una Raspberry Pi física, sin embargo, se desarrolló el proyecto en el sistema operativo de la Raspberry Pi para PC, el cual nos permite simular nuestros programas con una fidelidad muy cercana a la de una Raspberry Pi física.

Las funcionalidades de una Raspberry Pi, pueden ser varias, ya que al ser tan pequeño es de gran utilidad, además permite poder hacer código libre por ello no hay fronteras en cuanto a las posibilidades del dispositivo gracias a los aportes que hacen los programadores en los foros y demás. (Lucas, 2019).

Información sobre el cuidado de la salud y advertencias de riesgos

Se trabajó una máquina virtual para realizar nuestro proyecto por esa misma razón se tienen las siguientes consideraciones:

- El hacer uso de una máquina virtual conlleva una gran responsabilidad, se le pide al usuario que no exceda los recursos que se le proponen para poder ocuparla en su computadora, esto es para evitar una baja de rendimiento en su equipo o algún daño que pueda causar un exceso de procesamiento.
- Existe la posibilidad de que la máquina virtual que se ejecuta en su equipo se cierre por algún fallo y pierda todo su progreso realizado en ella, por eso mismo se recomienda que haga copias de seguridad cada 8 min, para evitar pérdidas significativas.
- Se recomienda el uso de reguladores de voltaje para proteger el equipo contra fallas eléctricas.
- Evite consumir cualquier alimento o bebida sobre su ordenador para preservar la salud de este.
- En caso de una tormenta eléctrica, se debe desenchufar la toma de corriente de su ordenador, y no conectarlo hasta que esta pase.

Configuración del proyecto

Para poder hacer uso del proyecto se deben realizar los pasos siguientes:

1. Descarga de VirtualBox.

Por la eficacia que presento VirtualBox para este proyecto, se le recomienda descárgalo, sabemos que existen diversos programas útiles para la virtualización de un .iso, sin embargo, la mayoría no son gratuitos, el ver este panorama VirtualBox es una opción muy conveniente.

Se proporciona la liga para que pueda descargarlo e instalarlo de acuerdo con su sistema operativo.

<https://www.virtualbox.org/wiki/Downloads>

2. Descarga de la imagen del sistema operativo (.iso) Raspbian y su instalación en VirtualBox.

Se te proporciona la liga en donde puedes descargar la imagen de Raspbian.

<https://www.raspberrypi.org/software/raspberry-pi-desktop/>

Una vez que tengas la imagen, podrás configurar tu máquina virtual, en caso de que necesites algún tutorial para la configuración e instalación de tu imagen de Raspbian, se te proporciona la liga siguiente:

<https://cifpn1.com/tic/?p=2564>

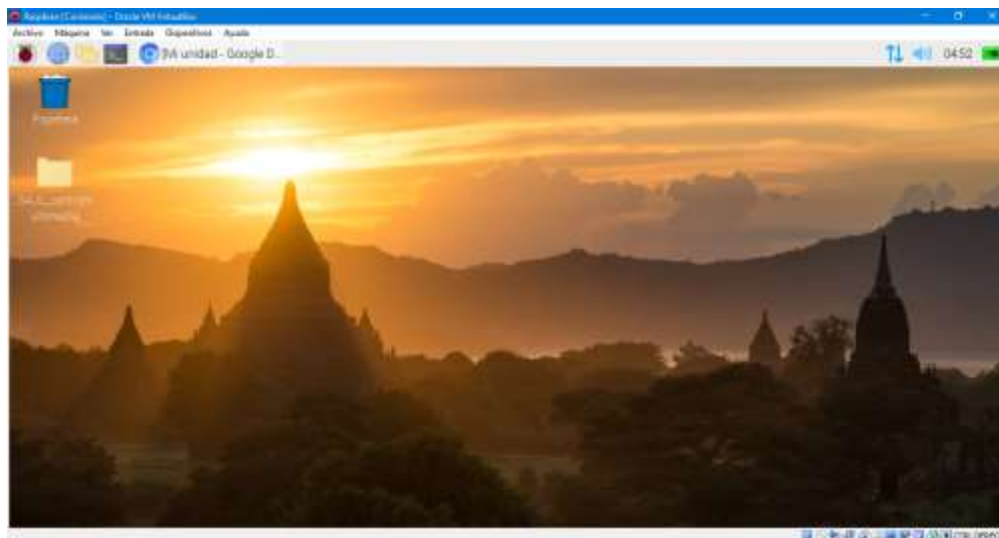


Figura 01. Pantalla inicial de Raspbian, es el resultado que se debe obtener después de instalar y configurar todo correctamente.

3. Instalación del lenguaje de programación y los paquetes a utilizar.

Instalaremos lo que es el lenguaje de programación Python 3, en su versión 3.7. Raspbian ya trae instalado Python en su versión 2.7, pero esa versión no la utilizaremos para nuestro fin por algunos problemas que puede presentar en la instalación de los paquetes. Abriremos nuestra terminal e ingresaremos los siguientes comandos.

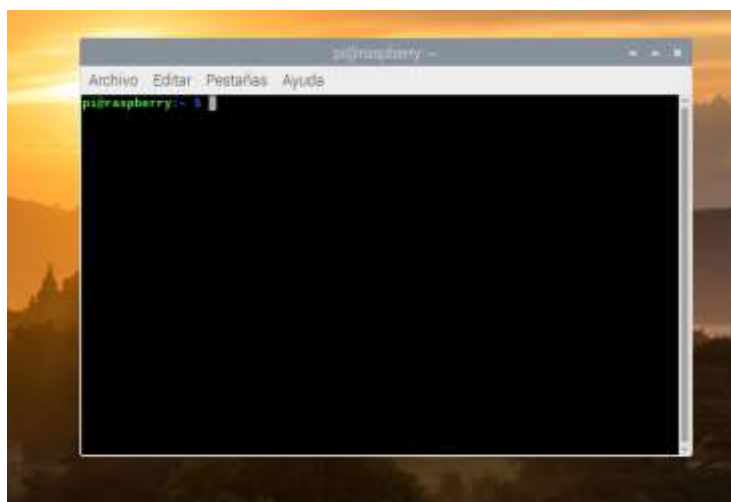


Figura 02. Se muestra la terminal de Raspbian.

Comandos que se ingresaran en la terminal:

sudo apt-get update

sudo apt-get install python3.7

sudo apt-get install python3-tk

sudo pip install Pillow

Como haremos uso del software de VLC, ingresaremos los siguientes comandos:

sudo apt install vlc

pip install python-vlc

4. Descargar el proyecto de Github.

Después de tener todo perfectamente configurado procederemos a descargar el proyecto desde el repositorio que fue cargado, se adjunta el link:

https://github.com/LuisArturoSandovalTole/Centro_de_Multimedia_2021_2

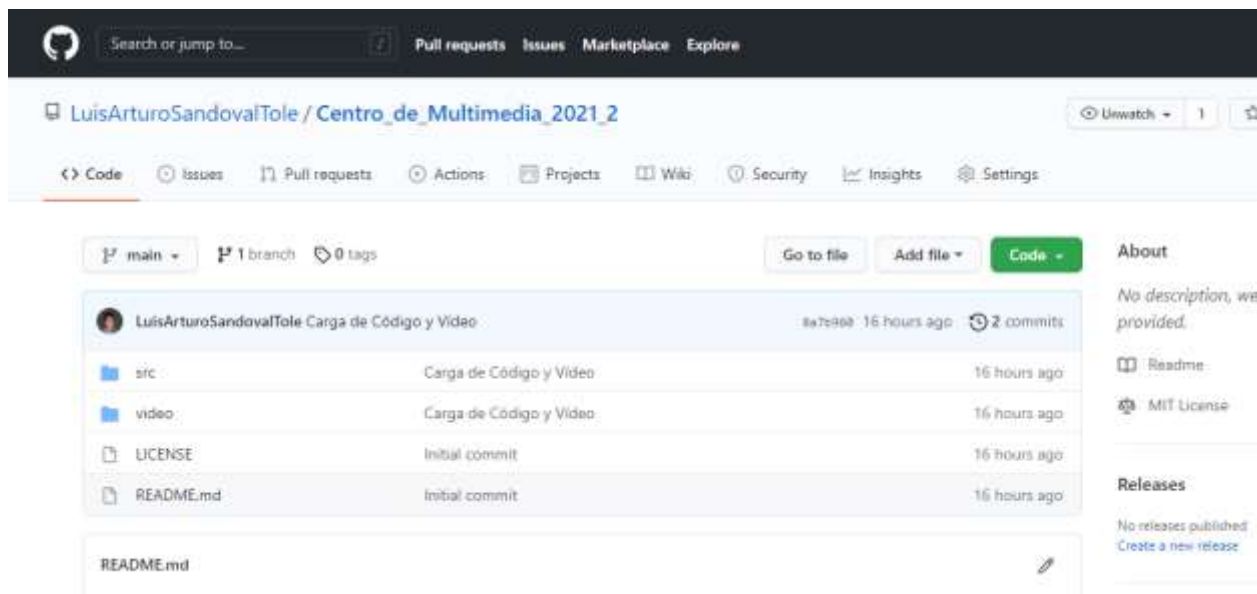


Figura 03. Imagen del repositorio de GitHub.

Dentro de la página del repositorio, le daremos click en la flecha que está dentro del botón verde que dice “Code”, y seleccionamos la opción que download ZIP, y se empezara a descargar el proyecto completo.

Una vez descomprimido el archivo estarán 3 carpetas donde se encuentra el código del programa a ejecutar, seleccionaremos la carpeta src, donde se encuentra el código de nuestro archivo, que se llamara SAJL_centromultimedia.py.

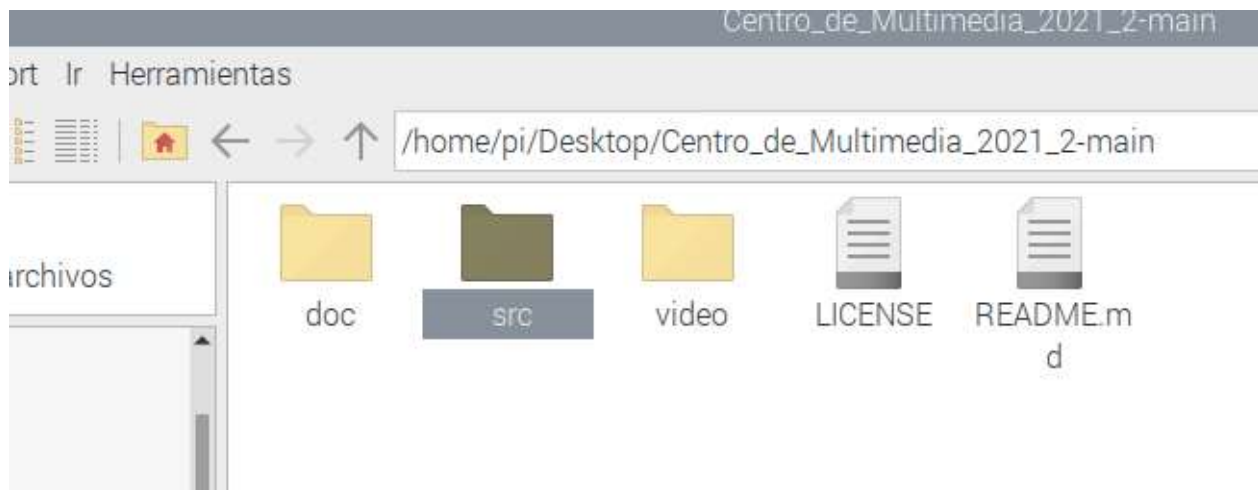


Figura 04. Archivos de nuestro proyecto.

5. Ejecución del proyecto.

Para poder ejecutar el proyecto en el código del programa es necesario tener archivos .png, que son los encargados de darle al proyecto el toque visual. Los requerimientos del proyecto son muy claros, solo se podía adjuntar en nuestra entrega: el documento, el código fuente y el video, por ende, no se adjuntaron estos archivos. Sin embargo, adjunto un link de Google drive donde se pueden descargar:

https://drive.google.com/drive/folders/1CNjPZ3o1PMQvkb_G63GgTb89e975tpn?usp=sharing

blim.png	hbogo.png	play.png	stop.png
cloud.png	música.png	retro.png	videos.png
fotos.png	net.png	spoti.png	

En la carpeta src, se colocarán todos los archivos .png.

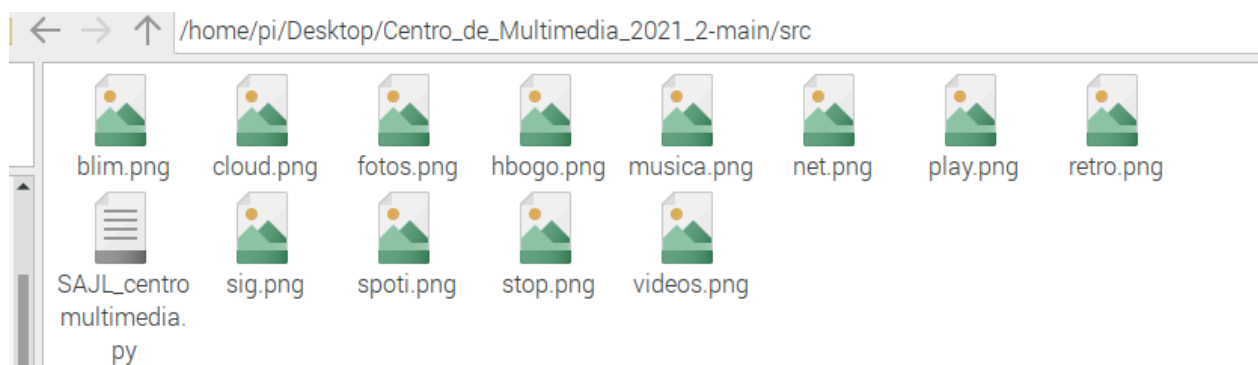


Figura 05. Carpeta src con archivos .png.

Abriremos nuestra terminal, nos colocaremos en nuestro archivo raíz, y ejecutaremos el SAJL_centromultimedia.py.



Figura 06. Ejecución de SAJL_centromultimedia.py



Figura 07. Resultado de la ejecución de SAJL_centromultimedia.py, ventana 1 "Streaming de Vídeo".



Figura 08. Resultado de la ejecución de SAJL_centromultimedia.py, ventana 2 “Música Digital”.

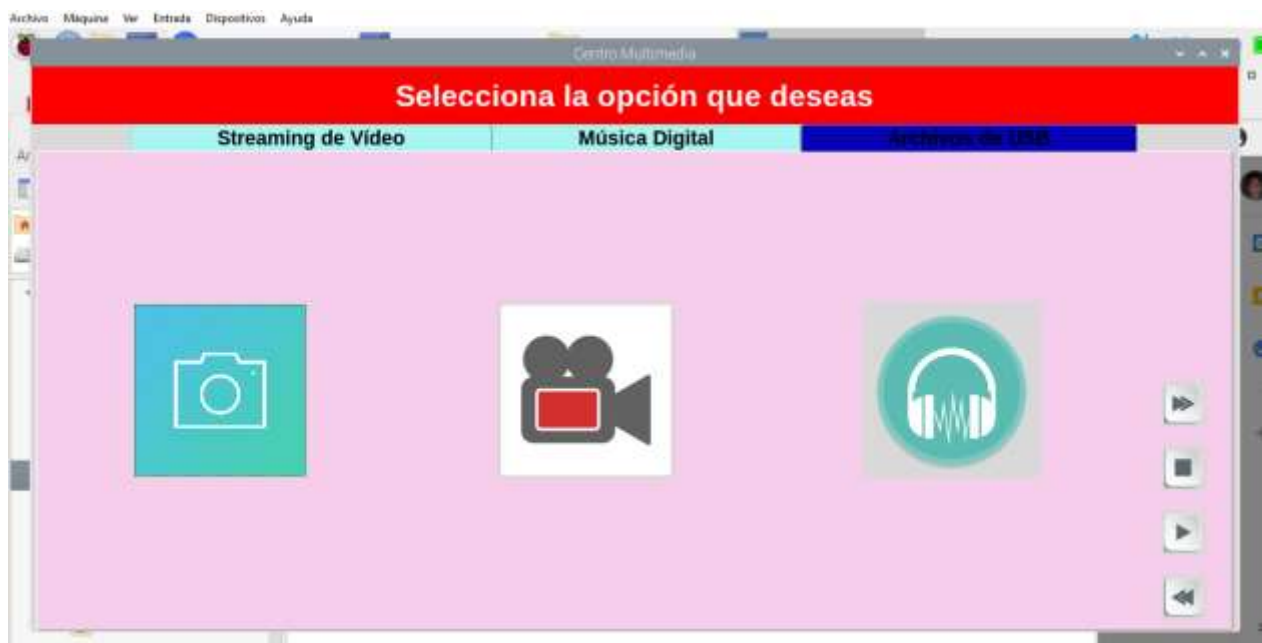


Figura 09. Resultado de la ejecución de SAJL_centromultimedia.py, ventana 3 “Archivos de USB”.

Para la ventana 3 de “Archivos de USB”, en el código de SAJL_centromultimedia.py, en la línea 24 se tendrá que colocar manualmente la ruta de la carpeta que contendrá los diferentes archivos que deseamos reproducir en nuestro Centro Multimedia, puede ser la ruta de la USB que ingreses o de cualquier carpeta que tengas en tu ordenador. En mi caso para fines demostrativos trabajé con una carpeta de mi ordenador llamada “archivos”.

```
24 ruta = r"/home/pi/Desktop/SAJL_centromultimedia/archivos"
```

Figura 10. Ruta de la carpeta con diferentes tipos de archivos.

La función principal de nuestra ventana 3(Archivos de USB.), es filtrar los archivos conforme a su categoría, al presionar el icono que represente cada una de las categorías (Vídeos, música y fotografías) nos reproducirá a través de una lista todos los archivos correspondientes.

Para las fotografías acepta los archivos “.jpg”, “.png” y “.jpeg”.

Para los vídeos acepta los archivos “.avi” y “.mp4”.

Para la música acepta los archivos “.mp3”.



Figura 11. Carpeta que contiene los diferentes archivos que reproducirá nuestro centro multimedia.

6. Funcionamiento del Centro multimedia.

Para poder observar el funcionamiento del proyecto, hay un vídeo en la plataforma de YouTube donde se observa nuestra interacción con el programa y correcto funcionamiento

<https://www.youtube.com/watch?v=7McqukirNak&t=2s>

Desarrollo del software

En esta sección explicaremos como se realizó cada parte que compone nuestro programa, tomamos como base una página de internet que nos sirvió mucho para la realización de nuestra interfaz llamada: <https://pythontutorial.net/tkinter/>

La recomendamos bastante para aquellas personas que quieran sacarle provecho a la librería de Tkinter.

Importaremos todos los paquetes y librerías que usaremos para nuestro proyecto, podemos ver que entre todos ellos destaca la que es tkinter, que es fundamental para el diseño de nuestra interfaz, así como la de PIL que nos permite soportar diferentes formatos de multimedia.

Ya que vamos a utilizar el software de VLC para la reproducción de nuestros diferentes tipos de archivos, importaremos de vlc a Instance.


```
SAJL_centromultimedia.py%
1 #Autor: Sandoval Juárez Luis Arturo
2 #Licencia: MIT
3
4 import tkinter as tk
5 from tkinter import ttk
6 import PIL
7 from PIL import Image
8 from PIL import ImageTk
9 from vlc import Instance
10 import time
11 import os
12 import sys
13 import webbrowser
14
15 #####
```

Figura 12. Importar librerías y paquetes.

En la imagen siguiente encontramos la clase ReproduccionEnVLC, lo que hará en términos generales es realizar una lista de reproducción para cada tipo de categoría, como podemos observar hay diferentes opciones, poniendo de ejemplo la categoría de fotos, solo aceptara aquellos archivos que sean de tipo “.jpg”, “.png” y “.jpeg”. Todo esto se realiza con base en la ruta que se le dé a nuestro programa, ya sea la ruta de una carpeta de nuestro ordenador, o la de nuestro USB que vayamos a ingresar.

```
SAJL_centromultimedia.py%
17 class ReproduccionEnVLC: #Crea ReproduccionEnVLC
18     def __init__(self):
19         self.Repro = Instance('Bucle') #Lo generaremos en un Bucle
20
21     def añadeLista(self,opcion): #Añade a la lista dependiendo el tipo de archivo
22         self.mediaList = self.Repro.media_list_new() #Se crea una lista de reproducción nueva
23
24         ruta = r"/home/pi/Desktop/SAJL_centromultimedia/archivos"
25         archivos = os.listdir(ruta) #Se creará la lista de todas la dirección de los archivos
26
27         for s in archivos: #Se elige una opción y a partir de esa se mostraran los archivos de ese tipo
28             if opcion==1: #Visualizar FOTOS
29                 if '.jpg' in str(s) or '.png' in str(s) or '.jpeg' in str(s): #Si el archivo es png ó jpeg, estará presente.
30                     self.mediaList.add_media(self.Repro.media_new(os.path.join(ruta,s)))
31
32             if opcion==2: #Visualizar VIDEOS
33                 if '.mp4' in str(s) or '.avi' in str(s): #Si el archivo es .avi ó .mp4, estará presente.
34
35                     self.mediaList.add_media(self.Repro.media_new(os.path.join(ruta,s)))
36
37             if opcion==3: #Escuchar Música
38                 if '.mp3' in str(s): #Si el archivo es .mp3 estará presente.
39                     self.mediaList.add_media(self.Repro.media_new(os.path.join(ruta,s)))
40
41         self.listaRepro = self.Repro.media_list_player_new() #Lista de Reproducción Vacía creada en ReproduccionEnVLC
42         self.listaRepro.set_media_list(self.mediaList) #Reemplaza la lista anterior, con la nueva.
43 #####
```

Figura 13. Clase ReproduccionEnVLC.

Como sabemos VLC trae por default algunas funciones que son: play,next, pause, previous y stop. Lo que haremos en esta parte del código es definir cada una de esas funciones y añadírselas a la lista de reproducción que se haya creado, para poder interactuar con ella dependiendo de lo que se quiera realizar.

Por otro lado, se realiza la función repMúsica, misma que crea las listas de reproducción con la ayuda del objeto de la clase VLC, procederá a realizar las lista de reproducciones, a ejecutarlas, y si se está ejecutando a no pasar a otra, y después pasar a la siguiente.

```

43 #####
44 #Acciones de VLC
45
46 def play(self):#Empieza a reproducir
47     self.listaRepro.play()
48 def next(self):#Siguiente
49     self.listaRepro.next()
50 def pause(self):
51     self.listaRepro.pause()
52 def previous(self):#Anterior
53     self.listaRepro.previous()
54 def stop(self):#Pausa
55     self.listaRepro.stop()
56
57 def repMúsica(opcion):
58
59     reproduce.añadeLista(opcion)#Realiza lista de reproducción
60
61     reproduce.play()#Se estará reproduciendo
62
63     while reproduce.is_playing():#Si se está ejecutando, no habrá una paso a la siguiente.
64         time.sleep(1)
65
66     reproduce.next()#Reproduce la siguiente.
67     time.sleep(9)
68
69 #####

```

Figura 14. Acciones de VLC y la función repMúsica.

Lo que corresponde a continuación es la creación de nuestra interfaz que será lo que el usuario final percibirá gráficamente al ejecutar el proyecto.

Para este primer parte se realiza lo que es la ventana principal que contendrá toda la interfaz, tendrá un nombre y unas dimensiones que nosotros elegimos para que se pudiera observar de una manera correcta.

Después realizaremos el primer label que nos servirá para indicarle al usuario que seleccione la opción que desee, para esto solo se necesita indicarle que aparecerá en nuestra ventana principal, indicar que contendrá el texto, el color del fondo, tipo de letra y tamaño, y en donde se ubicara respecto a los ejes x & y de nuestra ventana.

```

70 #Realización de Ventana
71 ventana = tk.Tk()#Se crea la ventana
72 ventana.title("Centro Multimedia")#Nombre de la ventana
73 ventana.geometry("1280x720")#Dimensiones
74
75 #Realización de un Label
76 label1 = tk.Label(
77     ventana,
78     text='Selecciona la opción que deseas',#Texto
79     bg="red",#Color del fondo
80     fg="white",#Color de la letra
81     font=("Arial",24,"bold")#Características del texto
82 )
83 label1.pack(#Posición
84     padx=10,
85     pady=10,
86     fill='x'

```

Figura 15. Realización de nuestra ventana principal y nuestro primer label.

Vamos a darle forma al menú de selección indicando sus dimensiones, y el color que cambiara dependiendo si se ha seleccionado algunas de las opciones.

```
88 #Menu de Selección
89 style = ttk.Style()
90
91 try:
92     style.theme_create( "seleccion", parent="alt", settings={
93         "TNotebook": {"configure": {"tabmargins": [100, 0, 100, 0] } },#dimensiones del label
94         "TNotebook.Tab": {
95             "configure": {"padding": [90, 0],"font": {'Arial', '16', 'bold'}, "background": "#A9F5F2" },#Fondo de Todo
96             "map": {
97                 "background": [{"selected", "#0404B4"}],#Fondo de Color del Menú seleccionado
98                 "expand": [{"selected", [0, 0, 0, 0]}] } } } #dimensiones del boton
99     style.theme_use("seleccion")
100 except:
101
102     style.theme_use("seleccion")
```

Figura 16. Formato del menú de selección.

En esta parte del código definiremos las 3 ventanas que estaremos utilizando para diferentes propósitos, en este caso cada una tendrá su nombre característico, su fondo de color y la forma de expansión dentro de la ventana principal.

```
104 #Variable controlVentana que será llamada después
105 controlVentana=ttk.Notebook(ventana)
106
107 #Ventana 1 y fondo de color verde-Streaming de Vídeo
108 Streaming=tk.Frame(controlVentana,bg='#58FA58')#Fondo de color de la ventana
109 controlVentana.add(Streaming,text='Streaming de Vídeo')#Nombre de la ventana
110 controlVentana.pack(expand=1, fill="y")#Expande sobre Y
111
112 #Ventana 2 y fondo de color verde-Música Digital
113 MD=tk.Frame(controlVentana,bg='#F7D358')#Fondo de color de la ventana
114 controlVentana.add(MD, text='Música Digital')#Nombre de la ventana
115 controlVentana.pack(expand=1, fill="y")#Expande sobre Y
116
117 #Ventana 3 y fondo de color rosa-Archivos de USB
118 USB=tk.Frame(controlVentana,bg='#F6CEEC')#Fondo de color de la ventana
119 controlVentana.add(USB, text='Archivos de USB')#Nombre de la ventana
120 controlVentana.pack(expand=1, fill="y")#Expande sobre Y
```

Figura 17. Formato de las ventanas de nuestro proyecto.

A partir de esta parte se crean botones para cada imagen que contendrán la ruta de la imagen que corresponde a cada una, además del diseño de estas, como el tamaño, y la ubicación. En el caso de los botones creados para la ventana de Streaming con ayuda de la librería webbrowser, usaremos una función especial webbrowser.open, que nos permitirá que la dirección url que se le guarde a cada uno de los botones se abra en una nueva pestaña del navegador actual que se esté utilizando. Para el caso de la segunda ventana de Música, se realiza lo mismo que la de Streaming de Vídeo.

```

122 #Icons de Streaming de Video
123
124 #Icon de Netflix
125 netflix = Image.open('net.png')#Dirección de la imagen
126 netflix = netflix.resize((400,200),Image.ANTIALIAS)#Tamaño del cuadro de la imagen
127 netflix = ImageTk.PhotoImage(netflix)#Colocar imagen en Netflix
128 ttk.Button(Streaming,image=netflix,#Al momento de seleccionar te redirecciona a el link, ubicación del icon
129             command=lambda : webbrowser.open("http://www.netflix.com", new=2, autoraise=True)).pack(padx=20,pady=40)
130
131 #Icon de HBO-GO
132 hbo = Image.open('hbogo.png')#Dirección de la imagen
133 hbo = hbo.resize((400,200),Image.ANTIALIAS)#Tamaño del cuadro de la imagen
134 hbo = ImageTk.PhotoImage(hbo)#Colocar imagen en hbo
135 ttk.Button(Streaming,image=hbo,#Al momento de seleccionar te redirecciona a el link, ubicación del icon
136             command=lambda : webbrowser.open("https://www.hbo.com", new=2,autoraise=False)).pack(padx=150,side="left")
137
138 #Icon de Blim
139 blim = Image.open('blim.png')#Dirección de la imagen
140 blim = blim.resize((400,200),Image.ANTIALIAS)#Tamaño del cuadro de la imagen
141 blim = ImageTk.PhotoImage(blim)#Colocar imagen en hbo
142 ttk.Button(Streaming,image=blim,#Al momento de seleccionar te redirecciona a el link, ubicación del icon
143             command=lambda : webbrowser.open("http://www.blim.com", new=2,autoraise=True)).pack(padx=20,side="left")
144
145 #####

```

Figura 18. Botones para la ventana de “Streaming de Vídeo”.

```

146 #Icons de Música Digital
147
148 #Icon de Spotify
149 spoti = Image.open('spoti.png')#Dirección de la imagen
150 spoti = spoti.resize((400,200),Image.ANTIALIAS)#Tamaño del cuadro de la imagen
151 spoti = ImageTk.PhotoImage(spoti)#Colocar imagen en Netflix
152 ttk.Button(MD,image=spoti,#Al momento de seleccionar te redirecciona a el link, ubicación del icon
153             command=lambda : webbrowser.open("http://www.spotify.com", new=2, autoraise=True)).pack(padx=20,pady=40)
154
155 #Icon de Cloudflare
156 cloud = Image.open('cloud.png')#Dirección de la imagen
157 cloud = cloud.resize((400,200),Image.ANTIALIAS)#Tamaño del cuadro de la imagen
158 cloud = ImageTk.PhotoImage(cloud)#Colocar imagen en hbo
159 ttk.Button(MD,image=cloud,#Al momento de seleccionar te redirecciona a el link, ubicación del icon
160             command=lambda : webbrowser.open("https://www.cloudflare.com", new=2,autoraise=False)).pack(padx=150,side="left")
161
162 #####

```

Figura 19. Botones para la ventana de “Música Digital”.

En esta sección los primeros 3 botones que se utilizan en la ventana de “Archivos USB”, serán los correspondientes a las listas creadas dependiendo de la categoría que pertenecen cada uno de nuestros archivos. De esta forma se llamará a ReproduccionEnVlc, y en cada imagen se le asigna las categorías definidas al principio, y el funcionamiento sé y estructura se realiza al igual como de un botón.

```

163 #####
164 #Icons de USB
165
166 reproduce = ReproduccionEnVLC()#Reproduce-llama a ReproduccionEnVLC
167
168 fotos = Image.open('fotos.png')
169 fotos = fotos.resize((180,180),Image.ANTIALIAS)
170 fotos = ImageTk.PhotoImage(fotos)
171 ttk.Button(USB,image=fotos,
172             command=lambda : repMusica(1)).pack(padx=100,pady=0,side="left")
173
174 videos = Image.open('videos.png')
175 videos = videos.resize((180,180),Image.ANTIALIAS)
176 videos = ImageTk.PhotoImage(videos)
177 ttk.Button(USB,image=videos,
178             command=lambda : repMusica(2)).pack(padx=100,pady=0,side="left")

```

```

180 musica = Image.open('musica.png')
181 musica = musica.resize((180,180),Image.ANTIALIAS)
182 musica = ImageTk.PhotoImage(musica)
183 ttk.Button(USB,image=musica,
184           command=lambda : repMusica(3)).pack(padx=100,pady=0,side="left")
185
186 #####

```

Figura 20. Botones para la ventana de “Archivo USB” para seleccionar las listas de reproducción dependiendo de la categoría de nuestra elección.

Para el caso de los botones de reproducción que serán utilizados para interactuar con nuestras listas creadas, se definirán igual como botones, pero con la excepción que se les pasara las funciones que realiza VLC.

```

187 #####
188
189 #Botón de Atras
190
191 atras = Image.open('retro.png')
192 atras = atras.resize((40,40),Image.ANTIALIAS)
193 atras = ImageTk.PhotoImage(atras)
194 ttk.Button(USB,image=atras,
195           command=lambda : reproduce.previous()).pack(pady=10,side="bottom")#Se manda a llamar a la función reproduce
196
197 #Botón de Play/Pause
198
199 play = Image.open('play.png')
200 play = play.resize((40,40),Image.ANTIALIAS)
201 play = ImageTk.PhotoImage(play)
202 ttk.Button(USB,image=play,
203           command=lambda : reproduce.play()).pack(pady=10,side="bottom")#Se manda a llamar a la función play/pause
204
205 #Botón de Stop
206
207 stop = Image.open('stop.png')
208 stop = stop.resize((40,40),Image.ANTIALIAS)
209 stop = ImageTk.PhotoImage(stop)
210 ttk.Button(USB,image=stop,
211           command=lambda : reproduce.stop()).pack(pady=10,side="bottom")#Se manda a llamar a la función stop
212
213 #Botón de Siguiente
214
215 sig = Image.open('sig.png')
216 sig = sig.resize((40,40),Image.ANTIALIAS)
217 sig = ImageTk.PhotoImage(sig)
218 ttk.Button(USB,image=sig,
219           command=lambda : reproduce.next()).pack(pady=10,side="bottom")#Se manda a llamar a la función next
220
221 ventana.mainloop()

```

Figura 21. Botones para la ventana de “Archivo USB” para la interacción con nuestras listas de reproducción.

Conclusiones

Como conclusiones se pudo hacer uso de los conocimientos adquiridos durante el curso que implicaron la realización de este proyecto, comprobando que se han adquirido conocimientos robustos sobre cómo operan los sistemas embebidos, así como de sus características y limitaciones.

También se realizó de manera correcta la simulación nuestro proyecto en el sistema operativo de Raspbian.

Se cumplieron los requerimientos del Centro Multimedia, dado el pequeño tiempo que se contó para interactuar con estas librerías con las que no se había trabajado en Python. Este proyecto es un logro personal que nos brinda un mayor número de herramientas, con las que podremos generar soluciones efectivas que logren impactar al mercado.

Cuestionario

Pregunta 1. ¿Qué paquete es necesario para realizar interfaces gráficas en Python? Escriba el comando que debe utilizar.

Es el paquete de Tkinter.

El comando que se debe utilizar es:

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

Pregunta 2. ¿Cómo se hace uso de “tk” para la versión de Python 2.7 y para la 3.7?

Python 2.7:

```
import Tkinter as tk
```

```
import ttk
```

Python 3.7:

```
Import tkinter as tk
```

```
from tkinter import ttk
```

Pregunta 3. ¿Qué paquete se debe utilizar para hacer uso de la función que se indica en la flecha gris, la línea 4 del código?

```
google = Image.open('google.png')
google = google.resize((220,120),Image.ANTIALIAS)
google = ImageTk.PhotoImage(google)
ttk.Button(Video,image=google,
            command=lambda : webbrowser.open("http://www.google.com", new=2, autoraise=True))
```



Se debe utilizar el paquete de webbrowser.

Pregunta 4. ¿Qué debería mostrar esa línea de código? Explique para qué sirve.

```
whatits = tk.Tk()
whatits ("" )
whatits ("1280x720")

whatits.mainloop()
```

Lo que debería mostrar sería una ventana con dimensiones de 1280x720, al solo ser ese fragmento de código no contendrá.

Este fragmento de código sirve para poder realizar la ventana donde se podrá observar e interactuar con la interfaz gráfica de nuestro proyecto.

Pregunta 5. Programa un botón llamado “Detente” que nos permita hacer uso de la función stop que viene en VLC.

```
ttk.Button(USB,text=" Detente ",  
           command=lambda : player.stop()).place(x=200, y=200)
```

Referencias

- Lucas, J. (2019, septiembre 20). Qué es Raspberry Pi | OpenWebinars. (Online). Available: <https://openwebinars.net/blog/que-es-raspberry-pi/>
- Pythontutorial. (2021). Tkinter. (Online). Available: <https://www.pythontutorial.net/tkinter/>
- Castro, A. (2014, enero 23). ¿Qué es Raspberry Pi, dónde comprarla y cómo usarla? (Online). Available: <https://computerhoy.com/noticias/hardware/que-es-raspberry-pi-donde-comprarlacomo-usarla-8614>
- Jose , F. (2020, mayo 5). Tutorial: cómo configurar una máquina virtual en VirtualBox. (Online). Available: <https://cifpn1.com/tic/?p=2564>
- Daniel, V.(2019). Raspberry pi: la tecnología reducida en placa. (Online). Available: <https://repository.usc.edu.co/bitstream/handle/20.500.12421/4250/RASPBERRY%20PI.pdf?sequence=3&isAllowed=y>
- Rakshitarora. (2021, enero 6). VLC module in Python – An Introduction. (Online). Available: <https://www.geeksforgeeks.org/vlc-module-in-python-an-introduction/>
- Fredrik, L. (2021). Pillow Installation. (Online). Available: <https://pillow.readthedocs.io/en/stable/installation.html>