

VIDEO PORTERO USANDO TARJETAS RASPBERRY PI 3

VIDEO INTERCOM USING RASPBERRY PI 3 CARDS

José Ignacio Vega Luna

Universidad Autónoma Metropolitana
vlji@correo.azc.uam.mx

Gerardo Salgado Guzmán

Universidad Autónoma Metropolitana
vlji@correo.azc.uam.mx

Mario Alberto Lagos Acosta

Universidad Autónoma Metropolitana
vlji@correo.azc.uam.mx

Francisco Javier Sánchez Rangel

Universidad Autónoma Metropolitana
vlji@correo.azc.uam.mx

José Francisco Cosme Aceves

Universidad Autónoma Metropolitana
vlji@correo.azc.uam.mx

Resumen

Se presenta el desarrollo de un video portero para edificios de oficinas integrado por un módulo maestro y tres módulos esclavos. El módulo maestro, instalado en la puerta del edificio, está compuesto por: una tarjeta Raspberry Pi, una cámara de video, una pantalla táctil, un micrófono USB y una bocina. Los módulos esclavos, instalados en las oficinas del edificio, están compuestos por: una tarjeta Raspberry Pi, una pantalla táctil, un micrófono USB y una bocina. Las funciones del video portero son: anunciar al visitante con un tono de timbre y comunicarlo con el residente a través de transmisión de audio y video en tiempo real, abrir la puerta y permitir al visitante registrar un mensaje. Se logró un alcance de 45 metros con línea de vista en la comunicación WiFi entre los módulos y el punto de acceso y un retraso máximo de la señal de video de 50 ms.

Palabras Claves: Micrófono USB, pantalla táctil, Raspberry Pi, video portero, WiFi.

Abstract

The development of a video intercom for office buildings is presented, composed of a master module and three slave modules. The master module, installed in the door of the building, is composed of: a Raspberry Pi card, a video camera, a touch screen, a USB microphone and a speaker. The slave modules, installed in the offices of the building, are composed of: a Raspberry Pi card, a touch screen, a USB microphone and a speaker. The functions of the video intercom are: announce the visitor with a ring tone and communicate it with the resident through audio and video transmission in real time, open the door and allow the visitor to register a message. A range of 45 meters was achieved with line of sight in the WiFi communication between the modules and the access point and a maximum delay of the 50 ms video signal.

Keywords: *Raspberry Pi, touchscreen, USB microphone, video itercom, WiFi.*

1. Introducción

Desde la aparición del interfono, los porteros electrónicos se han desarrollado a la par conforme avanza la tecnología, particularmente los sistemas digitales. En principio, los interfonos fueron analógicos y con pocas funciones. En la puerta de la vivienda o edificio estaba instalada una placa o panel con uno o varios botones, un micrófono, una bocina y una contrachapa eléctrica, mientras que en el interior se encontraba un timbre, una bocina, un micrófono y un interruptor para activar la contrachapa y abrir la puerta. Posteriormente, la bocina y micrófono fueron sustituidos por un auricular. A pesar de que los avances tecnológicos han propiciado cambios en este tipo de sistemas, sus funciones esenciales se han preservado: identificar la persona que intenta contactar al propietario, habilitar una conversación entre el propietario y visitante y automatizar el control del ingreso. Gracias al uso de estos sistemas, la seguridad y tranquilidad de viviendas y edificios de departamentos y oficinas ha aumentado considerablemente, ya que el propietario puede tomar la decisión de no abrir ante cualquier eventualidad o amenaza. El interfono ha evolucionado a lo que hoy se conoce como video portero. El video portero realiza las funciones de un interfono e incorpora cámaras de video y

pantallas táctiles, lo cual permite al propietario ver al visitante antes de entablar la conversación y permitirle o no la entrada, brindando un nivel adicional de seguridad y evitando situaciones incómodas, no deseadas o peligrosas [Myakala, 2017]. Con la gran explosión tecnológica de los últimos años, se han incorporado características estéticas y funciones a un video portero que han hecho que en la actualidad se tenga disponible una gran cantidad de modelos y proveedores y se conviertan en sistemas cada día más demandados y útiles. Algunos modelos integran una solución de circuito cerrado de televisión que permite grabar permanentemente lo que sucede en la puerta, otros cuentan con sensores de movimiento para activar el circuito cerrado de televisión. Otras funcionalidades incorporan: audio mejorado sin interferencias, cámaras de visión nocturna, cámaras angulares y de alta resolución, sensores de temperatura para el exterior, identificación del propietario y visitantes por medio de huellas digitales, tecnología Bluetooth y tarjetas RFIID, adaptaciones para personas con capacidades diferentes, buzón de mensajes y conexión a la red de datos. Algunos video porteros realizan la conexión a la red de datos usando tecnología WiFi, permitiendo al usuario acceder remotamente al portero para hablar y ver al visitante y controlar la puerta desde la Internet usando una computadora o dispositivo móvil.

En este trabajo se presenta la implantación de un video portero realizado a petición de un edificio de oficinas. Se solicitó que el video portero realice las siguientes funciones: emitir un tono de timbre cuando una persona llama a la puerta, aceptar o rechazar la llamada en la oficina, entablar el dialogo entre el visitante y el residente, apertura de puerta, buzón de mensajes cuando el residente esté ausente y terminar el dialogo. Se solicitó además que el sistema use pantallas táctiles, tecnología WiFi, sea confiable y fácil de instalar y usar. Se acordó realizar un sistema piloto con tres oficinas y extender su instalación a 10 oficinas más después de probarlo y evaluarlo. Tomando en cuenta las funcionalidades y características anteriores, se propuso un video portero compuesto por un módulo maestro y tres módulos esclavos. El módulo maestro se instaló en un panel de la puerta de entrada del edificio y cada módulo esclavo en una de tres oficinas. Los dos tipos de módulos se construyeron en base a una tarjeta Raspberry Pi 3 B+ con sistema operativo

Raspbian. Uno de los puntos más importantes de este trabajo es la forma de implantar la transmisión de audio y video en tiempo real o streaming. Existen varias técnicas y procedimientos para realizar el streaming. Particularmente, Raspbian cuenta con varias herramientas y puede integrar funciones de biblioteca de software libre que se invocan desde un programa en Python para la captura y reproducción de audio y video. Una vez realizada la captura, puede utilizarse alguna técnica o herramienta para realizar el streaming desde Python en Raspbian, una de las más poderosas es ffmpeg. La herramienta ffmpeg es una colección de software libre que puede grabar, convertir, mezclar y realizar streaming usando una biblioteca de codecs incorporados. Con ffmpeg se obtienen los tiempos de latencia bajos, menos de 50 ms, es fácil de utilizar y genera programas sencillos. Otra alternativa para realizar streaming es usar un servidor de protocolo de streaming en tiempo real (RTSP-Real Time Streaming Protocol) en conjunto con una herramienta que genere el streaming, por ejemplo picamera. Existen bibliotecas de código libre para instalar y configurar un servidor RTSP en Raspbian. Lo único necesario es dirigir la captura de audio y video al servidor RTSP y éste procesará las solicitudes de streaming de los clientes a través de la red. A pesar de la facilidad de uso de este método, presenta algunos problemas de compatibilidad y comunicación con algunas versiones de sistemas operativos de clientes, sobre todo en dispositivos móviles [Banu, 2017]. La razón principal de usar la tarjeta Raspberry Pi en este trabajo fue porque existe una gran cantidad de aplicaciones y bibliotecas desarrolladas por la comunidad de código abierto para Raspbian de fácil instalación, configuración y uso. En lo que respecta a video porteros con arquitecturas similares a la del trabajo aquí presentado desarrollados los últimos años, la mayoría de ellos usan comunicación alámbrica a través de interfaces Ethernet. Por otro parte, la tecnología se está enfocando más en el desarrollo de sistemas de video vigilancia y algoritmos y técnicas más eficientes y seguras de streaming. Un sistema de video vigilancia transmite las señales de audio y video que capta de un área para su registro y análisis. Los sistemas de video vigilancia se han vuelto parte integral de monitoreo de la seguridad urbana y de diferentes tipos de instalaciones debido a la reducción del costo de cámaras de video y sensores para detección automática de eventos

[Konda, 2017]. Se han creado recientemente sistemas de video vigilancia basados en tarjetas Raspberry Pi-3 y cámaras USB para abrir remotamente desde la Internet la puerta de un edificio o vivienda [Myakala, 2017], no cuentan con las funcionalidades de un video portero como el realizado en este trabajo. Otros trabajos recientes en este campo se han concentrado en la creación de algoritmos más eficientes para la detección de movimiento, la segmentación de streaming y algoritmos de compresión H.264, usando vectores de movimiento y compresión algoritmos. Trabajos adicionales se han centrado en el análisis de ambientes complejos, escenarios urbanos y áreas donde se concentran multitudes, como por ejemplo aeropuertos y estaciones de trenes, realizando seguimiento de personas y monitoreo de actividad, incluyendo detección de objetos abandonados y vigilancia de puertas para detectar amenazas potenciales [Conci, 2016]. Varios trabajos recientes han tenido como objetivo llevar a cabo video vigilancia remota, desde dispositivos móviles, en hogares de adultos mayores y niños [Chang, 2016], combinando inclusive diferentes tipos de sensores y actuadores para detectar y evitar intrusos, robos, fuego y fugas de gas con tarjetas Raspberry Pi 3 [Joshi, 2017], inteligencia artificial [Arnold, 2014] y control de electrodomésticos para conformar soluciones de Internet de las Cosas (IoT-Internet Of Things) [Watanabe, 2013]. En lo que respecta a las últimas investigaciones referentes a streaming, se han desarrollado plataformas para transmisión de video de ultra alta definición (UHD) para aplicaciones complejas, como realidad aumentada, que dominaran el tráfico en la futura quinta generación (5G) de redes móviles [Nightingale, 2018]. Estas plataformas y los algoritmos usados se basan en vistas múltiples de cámaras de alta definición [Zhang, 2018], uso de tasas de bits adaptivas (ABR) y ranuras de tiempo derivativas [Zahran, 2018]. Otras investigaciones de este tipo consideran que el tráfico de la Internet se triplicará en 2021 y que un alto porcentaje del tráfico será video, por lo que proponen el uso de protocolos nuevos como el Protocolo de Streaming Dual Flexible (FDSP), el cual combina la confiabilidad de TCP y baja latencia de UDP. Analiza las partes críticas del video para transmitir las vía TCP y el resto vía UDP [Gatimu, 2018]. Tomando en cuenta lo anterior, las aportaciones del trabajo aquí presentado son:

- No es un sistema de video vigilancia que usa componentes de tecnología actual y compactos, así como software de código libre.
- Utiliza tecnología WiFi disponible actualmente en casi cualquier oficina u hogar, evitando modificar la construcción donde se instala.
- Las herramientas software usadas son las últimas versiones de su clase, permitiendo transmitir audio y video en tiempo real con mínima latencia en una red inalámbrica.

2. Métodos

La metodología seguida en el desarrollo de este sistema consistió de dos fases. La primera fase fue dividirlo en dos tipos de módulos: el módulo maestro y los módulos esclavos, como se muestra en la figura 1.



Figura 1 Diagrama funcional del sistema.

La segunda fase, la cual consistió en seleccionar los componentes adecuados con los cuales pudieran implantarse las funciones solicitadas. A continuación, se explica la arquitectura de los dos tipos de módulos y la funcionalidad del sistema.

El módulo maestro

En la figura 2 se muestra la arquitectura del módulo maestro. Se utilizó una tarjeta de memoria SD de 16 GB en la tarjeta Raspberry para instalar el sistema operativo Raspbian kernel 4.9 y el buzón de mensajes. La pantalla táctil utilizada en el módulo maestro para implantar la interfaz de usuario fue el dispositivo Pi+TFT de 3.5", el

cual tiene una resolución de 480x320 y se conectó al puerto SPI de la tarjeta Raspberry Pi. La interfaz de usuario se realizó usando pygame. La herramienta pygame es un conjunto de bibliotecas que pueden usarse en un programa de Python para la implantación de videojuegos, programas multimedia e interfaces gráficas de usuario, ya que permite mostrar texto, imágenes y sonidos en una pantalla táctil y controlar la posición del cursor. Esta herramienta se instala por defecto con la versión de Raspbian para Raspberry Pi.

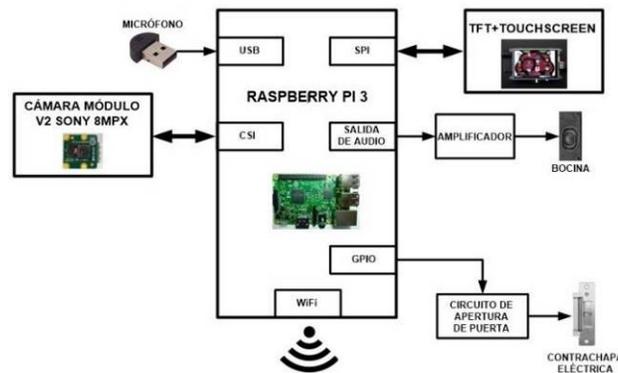


Figura 2 Arquitectura del módulo maestro.

La cámara de video del módulo maestro es el dispositivo denominado módulo de cámara para Raspberry V2 conectado a la interfaz CSI de la Raspberry Pi. La cámara de video cuenta con un sensor de alta resolución Sony IMX219 de 8 Megapíxeles. Permite capturar fotografías con una resolución máxima de 3238x2464 en formatos JPEG, GIF, BMP y PNG y video de alta definición. Existen bibliotecas de código abierto para usar la cámara y manipular fotos y video que pueden invocarse desde el shell de Raspbian o desde un programa en Python. La cámara puede controlarse usando el comando raspinstall, sin embargo, en este trabajo se utilizó la biblioteca python-picamera de Python en caso de que a futuro en el sistema sea necesario modificar las características de captura de video. La cámara se habilitó a través de la herramienta raspi-config de Raspbian y posteriormente se instaló la biblioteca python-picamera utilizando el comando: `sudo apt-get install python3-picamera`. El módulo maestro cuenta también con un micrófono mini USB el cual no necesita driver y se conectó a un puerto USB de la

Raspberry utilizando un cable de extensión USB para poder instalar el micrófono en el panel del video portero. El micrófono y la cámara de video tienen dos funciones: capturar las señales de audio y video del visitante para transmitir las por streaming al módulo esclavo y que el residente o persona en la oficina pueda ver y escuchar al visitante y permitirle registrar un mensaje cuando el residente se encuentre ausente. En sentido inverso, por seguridad el visitante solo puede escuchar al residente, no puede verlo en la pantalla táctil del módulo maestro. Para este fin, se conectó a la salida de audio de la Raspberry Pi un amplificador y un altavoz de 4 Ω . El amplificador clase D monoaural de 2.5 Watts empleado fue el dispositivo PAM8302. El módulo maestro puede activar el actuador de una contrachapa eléctrica por medio de un circuito conectado a una terminal GPIO configurada como salida. El programa que se ejecuta en el módulo maestro se realizó en Python 3.6 y realiza las siguientes funciones: inicializa temporizadores, la pantalla táctil, la cámara de video, la interfaz WiFi y la terminal GPIO, a continuación entra en un ciclo continuo donde explora cada 0.5 segundos la pantalla táctil. Cuando un visitante arriba a la puerta de entrada del edificio deberá tocar la pantalla. A continuación, se mostrará en la pantalla el mensaje de bienvenida y un menú de botones con el número de cada oficina. En el momento de seleccionar una oficina, el módulo maestro inicia la captura de audio y video y lo transmite por streaming al nodo esclavo correspondiente. El módulo maestro identifica a cada módulo esclavo a través de la dirección IP establecida en su interfaz WiFi. La comunicación entre el módulo maestro y los módulos esclavos se llevó a cabo usando intercambio de mensajes con sockets bajo el esquema cliente-servidor. El programa usa tres sockets a través de los puertos 5000, 5001 y 5002. El puerto 5000 es utilizado por el módulo maestro para transmitir por streaming el audio y video del visitante, el puerto 5001 es usado por el módulo esclavo para transmitir por streaming el audio del residente y el puerto 5002 es usado por el módulo esclavo para transmitir mensajes de respuesta y ordenes al módulo maestro. Posteriormente, el programa espera un mensaje por el puerto 5002. En caso de que el mensaje recibido sea "AUSENTE" muestra en la pantalla táctil un aviso indicando que el residente no se encuentra y solicita al visitante que registre un mensaje cuya duración no sea más

de 30 segundos. El mensaje es capturado por la cámara de video y el micrófono en un archivo con formato MP4 y es transmitido al módulo esclavo. Si el mensaje recibido del módulo esclavo es "CONTESTAR", indicará que el residente desea hablar con el visitante y el módulo maestro escucha por el puerto 5001 el streaming de audio transmitido por el módulo esclavo. Es este momento, se lleva a cabo la comunicación entre las dos personas. Acto seguido, el programa del módulo maestro espera un mensaje por el puerto 5002. Si el mensaje recibido es "ABRIR", indicará que el residente desea permitir el acceso al visitante y se activa el actuador de la contrachapa eléctrica para abrir la puerta. Si el mensaje es "TERMINAR", indicará que el residente desea terminar la comunicación con el visitante sin permitir el paso. En cualquiera de los casos, "ABRIR" o "TERMINAR", el programa cierra finalmente los puertos usados, apaga la pantalla táctil y retorna al inicio del ciclo para explorar la pantalla nuevamente. En la figura 3 se muestra el diagrama de flujo seguido para realizar la programación del módulo maestro. La forma como fueron implantadas las funciones del módulo maestro se explican a continuación.

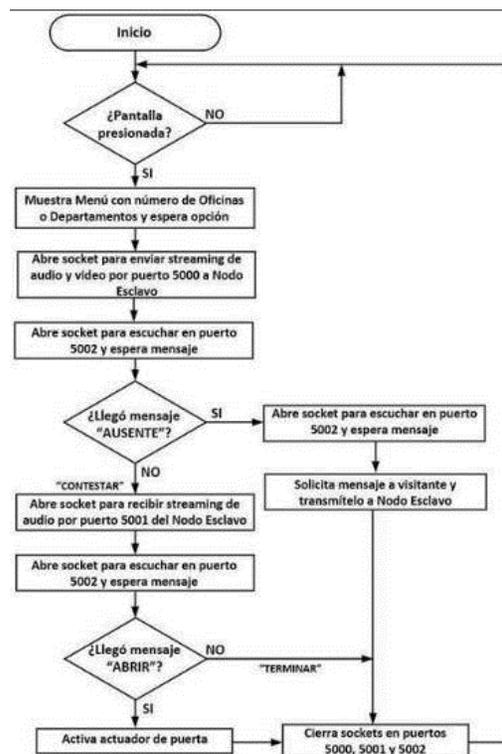


Figura 3 Diagrama de flujo del programa principal del módulo maestro.

Para configurar, reproducir el video y mostrar menús en la pantalla táctil se utilizaron las funciones de la biblioteca pygame para Python en Raspbian. Esta biblioteca se descargó desde el github: <https://bitbucket.org/pygame/pygame>. Para generar y transmitir el streaming de audio y video del módulo maestro a los módulos esclavos se usó la aplicación raspivid, las funciones de la biblioteca ffmpeg, el módulo de Python subprocess.Popen y la herramienta netcat. A continuación, se configuró la cámara usando las opciones por defecto: sin efectos especiales, modo de exposición y modo de balance de blanco en automático. Posteriormente, se instalaron las bibliotecas ffmpeg y netcat ejecutando los comandos siguientes: `sudo apt-get install ffmpeg` y `sudo apt-get install netcat` y finalmente se implantó el streaming mediante la siguiente secuencia de instrucciones:

```
cmd="raspivid -o - -t 0 -n -w 1280 -h 720 -fps 25 | ffmpeg -thread_queue_size 1024 -y -  
f h264 -framerate 25 -i - -thread_queue_size 512 -f alsa -ac 1 -i hw:1 -vcodec copy  
-acodec mp4"  
p_stream_av = subprocess.Popen(cmd, shell=True, stderr=subprocess.PIPE)  
p_stream_av | nc IP_Esclavo 5000
```

En la primera línea se ejecutó la instrucción raspivid para capturar el video en formato crudo H264, conocido como MPEG-4 parte 1, a 25 cuadros por segundo (fps-frames per second), entregándolo, por medio de una tubería, a la aplicación ffmpeg. La aplicación ffmeg toma desde la tubería el video, lo mezcla con la entrada de audio y lo convierte a un flujo de datos o stream en formato MP4. A pesar de que la cámara puede capturar video a una tasa máxima de 120 fps, se utilizó una tasa de 25 fps en la que el retraso o lag fue aceptable, menor a 50 ms.

En la segunda línea, se invoca al módulo de Python subprocess.Popen, el cual sirve para ejecutar un proceso externo e interactuar con el sistema operativo desde un programa en Python. En este caso el módulo subprocess.Popen ejecuta la primera de las tres líneas. En la tercera línea, el flujo de datos de la salida de la primera línea se entrega a la instrucción nc que pertenece a netcat. Netcat es una herramienta de red de software libre empleada para abrir puertos TCP/UDP y que un host se mantenga escuchando en esos puertos. Utilizando un puerto, netcat crea un socket para conectarse a un servidor o para que el host funcione como servidor,

en el primer caso transmite el flujo de datos de la entrada estándar de la instrucción nc por el socket, mientras que en el segundo caso envía por la salida estándar de la instrucción nc lo que recibe por el socket. De esta manera, la tercera línea crea un socket y se conecta al puerto 5000 del módulo esclavo para transmitir el streaming de audio y video que la instrucción nc recibe en su entrada estándar. Para reproducir el streaming de audio que recibe el módulo maestro de un módulo esclavo se usaron las herramientas omxplayer de Raspbian y netcat. El streaming de audio recibido se encuentra en formato MP3 y se reproduce mediante las instrucciones nc -l -p 5001 | omxplayer. La instrucción nc crea un socket usando el puerto 5001 de tal forma que el host trabaja como servidor y se mantiene a la escucha de un cliente en ese puerto. Cuando el cliente se conecta al puerto 5001, la instrucción nc envía a la herramienta omxplayer lo que recibe por el socket. El streaming de audio se reproduce en la bocina conectada a la salida de audio de la tarjeta Raspberry. Por último, para generar el archivo de audio y video que almacena el mensaje del visitante cuando el residente está ausente, se usó un procedimiento similar al utilizado para generar el streaming de audio y video, solo que, en lugar de transmitirlo al módulo esclavo a través de un socket, el audio y video se almacenaron en un archivo que se transmite al módulo esclavo para que éste último lo almacene en el buzón de mensajes.

Los módulos esclavos

Se construyeron tres módulos esclavos, todos con la misma arquitectura como la mostrada en la figura 4. En estos módulos se empleó una tarjeta Raspberry Pi 3 B+, una pantalla táctil Pi+TFT de 3.5" conectada al puerto SPI de la tarjeta Raspberry Pi, un micrófono mini USB conectado a un puerto USB de la Raspberry y un altavoz de 4 Ω conectado a la salida de audio de la Raspberry Pi por medio de un amplificador PAM8302.

Al mismo tiempo que el programa principal ejecuta el programa Escucha Módulo Maestro, explora si el residente toca la pantalla táctil. Al tocar la pantalla, el programa invoca la rutina Buzón de Mensajes. Esta rutina permite al residente ver los mensajes del buzón, escucharlos o eliminarlos. El buzón de mensajes se

almacena en la memoria SD de la tarjeta Raspberry. Al salir del buzón de mensajes el programa principal apaga la pantalla táctil después de 60 segundos.

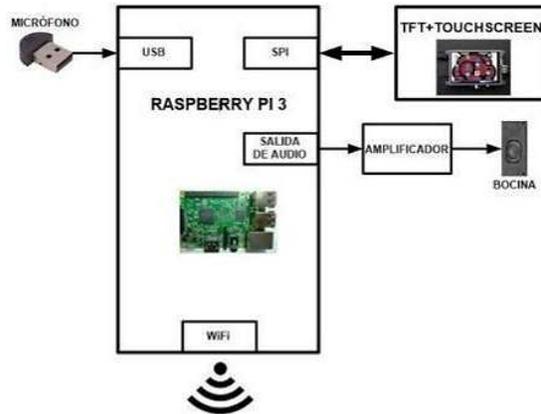


Figura 4 Arquitectura de los módulos esclavos.

En la figura 5 se muestra el diagrama de flujo del programa principal de los módulos esclavos.

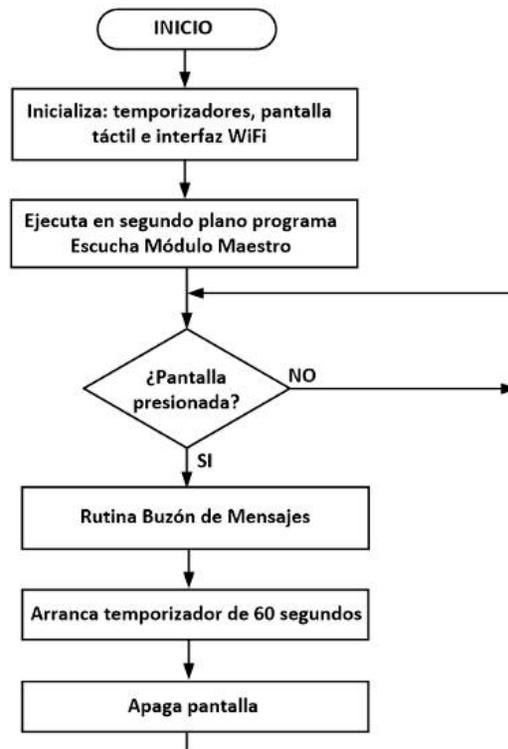


Figura 5 Diagrama de flujo del programa principal de los módulos esclavos

El programa Escucha Módulo Maestro inicia abriendo un socket usando el puerto 5000. Por medio de este puerto el módulo esclavo recibe, desde el módulo maestro, el streaming de audio y video que reproduce en la pantalla y en la bocina vía el amplificador cuando un visitante desea hablar con el residente. Al mismo tiempo, esta rutina reproduce un tono de timbre en la bocina, muestra en la pantalla el botón “CONTESTAR” y espera la respuesta del residente. Si el residente presiona el botón “CONTESTAR” se invoca la rutina Contestar, la cual se encarga de establecer y mantener la comunicación entre el visitante y el residente. Si el residente no presiona el botón “CONTESTAR” después de 5 tonos de timbre, el programa Escucha Módulo Maestro, termina la transmisión del streaming de audio, cierra el socket del puerto 5001 y transmite el mensaje “AUSENTE” al módulo maestro por el puerto 5002. El mensaje “AUSENTE” puede transmitirse aún en el caso que el residente se encuentre en la oficina y no presione el botón “CONTESTAR”, tal vez porque una vez que observó al visitante no desea abrir la puerta.

Después de transmitir el mensaje “AUSENTE”, el programa Escucha Módulo Maestro invoca la rutina Almacena Mensaje, la cual espera la recepción del archivo, con formato MP4, transmitido por el módulo maestro. Este archivo es el mensaje registrado por el visitante y es almacenado en el buzón. El programa Escucha Módulo Maestro termina apagando la pantalla después de 60 segundos, como se indica en el diagrama de flujo de la figura 6. En la figura 7 se indican las acciones realizadas por la rutina Contestar. Esta rutina inicia transmitiendo al módulo maestro el mensaje “CONTESTAR” usando el puerto 5002 y el streaming de audio que captura en el micrófono por el puerto 5001. Posteriormente, la rutina muestra en la pantalla táctil los botones “ABRIR” y “TERMINAR”. Mediante estos botones el residente puede indicar si desea abrir la puerta o terminar la comunicación con el visitante. Si el residente presiona el botón “ABRIR”, la rutina Contestar transmite, por el puerto 5002 al módulo maestro, el mensaje “ABRIR”. Después de presionar cualquiera de los dos botones anteriores, la rutina Contestar termina cerrando los sockets de los puertos 5000, 5001 y 5002 y apaga la pantalla después de 60 segundos. La forma como fueron implantadas las funciones de módulos esclavos

fue similar a la seguida en el módulo maestro. Para configurar, reproducir video y mostrar menús en pantalla táctil se utilizaron las funciones de biblioteca pygame.

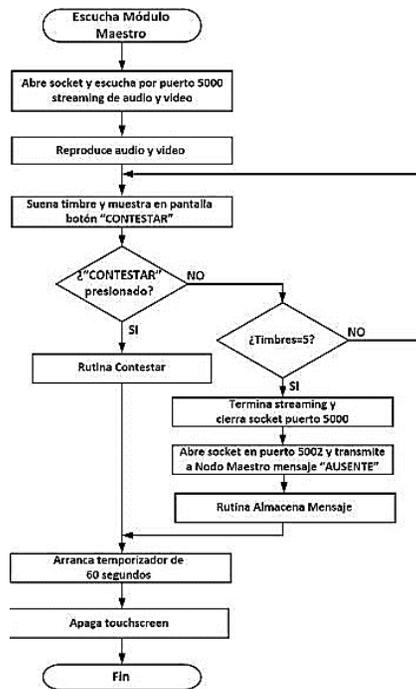


Figura 6 Diagrama de flujo del programa escucha módulo maestro.

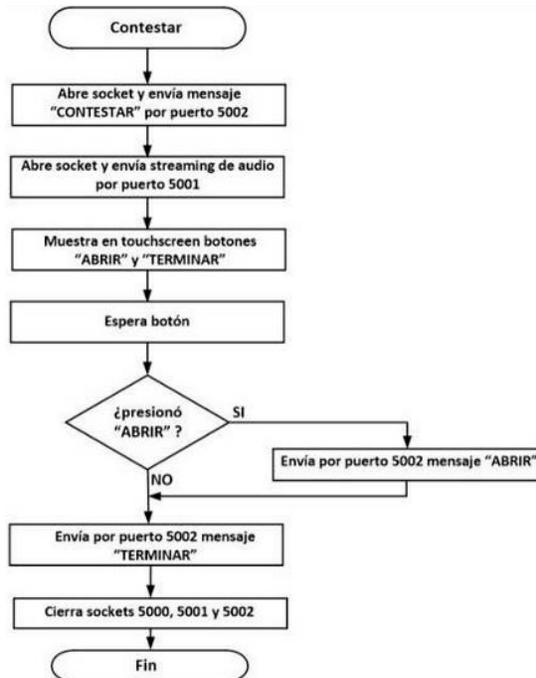


Figura 7 Diagrama de flujo de la rutina contestar.

Para generar y transmitir el streaming de audio en formato MP3 de este módulo al módulo maestro se usó la biblioteca ffmpeg, el módulo de Python subprocess.Popen y la herramienta netcat a través de la siguiente secuencia de instrucciones:

```
cmd="ffmpeg -thread_queue_size 1024 -y -i - -thread_queue_size 512 -f alsa -ac 1 -i  
hw:1 -vcodec copy -acodec mp3"  
p_stream_a = subprocess.Popen(cmd, shell=True, stderr=subprocess.PIPE)  
p_stream_a | nc IP_Maestro 5001
```

El flujo de datos en formato MP3 se envía a la entrada estándar de la instrucción nc. La instrucción nc de la tercera línea crea un socket y se conecta al puerto 5001 del módulo maestro para transmitir el streaming que recibe en su entrada estándar.

3. Resultados

Antes de realizar las pruebas al sistema, se llevaron a cabo los siguientes ajustes al sistema:

- En la configuración de la cámara de video se estableció la nitidez a 15, el contraste a 25 y la brillantez a 45 ya que las imágenes se mostraban demasiado claras y no se podía distinguir el rostro de las personas.
- Se estableció la rotación de la pantalla táctil a modo horizontal, o paisaje, modificando el parámetro `fbtft_device.rotate=90` en el archivo `/boot/cmdline.txt`.
- Usando el potenciómetro del amplificador de audio del módulo maestro se aumentó el volumen de la bocina ya que en el exterior el audio resultaba muy bajo usando la ganancia por defecto de 24 dB.

A continuación, se procedió a ejecutar cuatro grupos de pruebas. El primer grupo de pruebas tuvo como objetivo determinar la cantidad de espacio usada por el archivo generado al registrar un visitante un mensaje de 30 segundos en el buzón localizado en la memoria SD de la Raspberry de un módulo esclavo. Para llevar a cabo estas pruebas se generaron mensajes de 30 diferentes personas, usando en la captura de video 50 fps. El tamaño del archivo resultante fue de 10 MB en

promedio. Considerando que la capacidad de la memoria SD son 16 GB y el espacio usado por el sistema operativo Raspbian y bibliotecas utilizan 3.6 GB, restan más de 12 GB para el buzón de mensajes. El segundo grupo de pruebas tuvo como objetivo medir el alcance de la transmisión WiFi de los módulos esclavos. El punto de acceso WiFi Cisco WAP4410N está instalado en la planta baja del edificio, localizado a 4 metros del módulo maestro y con línea de vista. Para efectuar estas pruebas se ubicó un módulo esclavo en diferentes niveles y departamentos del edificio, a diferentes distancias del punto de acceso y con línea de vista. A continuación, se ejecutaron dos programas en el nodo: uno de ellos en segundo plano, su función fue transmitir continuamente un archivo al módulo maestro y el segundo programa ejecutó el comando iwconfig para registrar la velocidad de transmisión y nivel de la potencia de la señal WiFi recibida (RSSI-Received Signal Strength Indicator) desde el punto de acceso a cada lugar donde se ubicó el módulo esclavo. Los resultados indicaron que el alcance fueron 45 metros aproximadamente a una velocidad de 290 Mbps, un poco menor a los 300 Mbps que pueden lograrse teóricamente usando el estándar 802.11n. La distancia que existe entre el punto de acceso y la tercera planta del edificio son 30 metros. A una distancia mayor a 45 metros la potencia decreció aceleradamente y se perdió el enlace cuando el nivel cayó a los -83 dBm como se muestra en gráfica de figura 8.

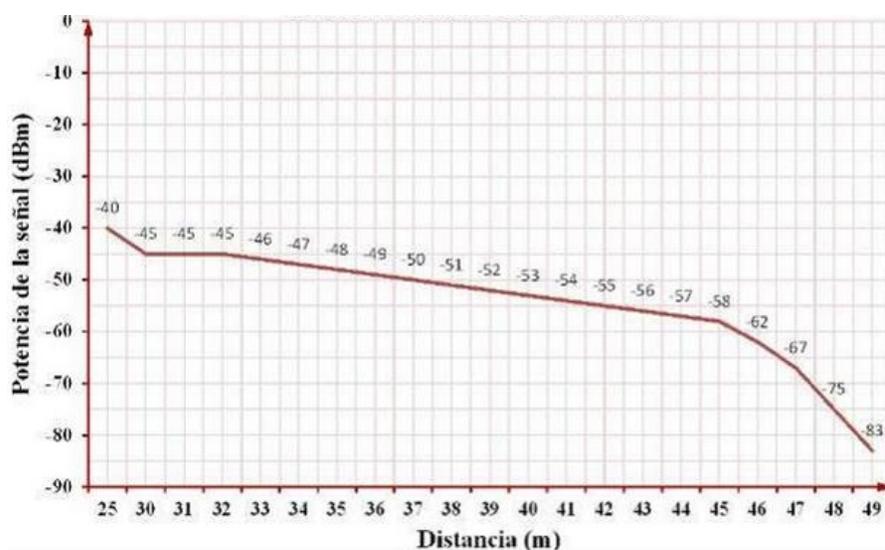


Figura 8 Alcance de la transmisión WiFi del sistema.

Adicionalmente, para verificar que el nivel de potencia reportado por el comando iwconfig fuera correcto, en cada punto donde se ubicó el módulo esclavo se colocó una computadora portátil para medir el nivel de RSSI utilizando la herramienta inSSIDer. Los valores de RSSI medidos con inSSIDer no tuvieron gran diferencia respecto a los indicados por el comando iwconfig. El tercer grupo de pruebas tuvo como objetivo determinar el tiempo de retraso en la transmisión de audio y video entre el módulo maestro y un módulo esclavo. En estas pruebas se tomaron las siguientes consideraciones:

- La velocidad entre los módulos esclavos y el punto de acceso es en promedio 280 Mbps.
- La captura de video se hizo con 50 fps. Antes de realizar estas pruebas se sincronizó el reloj de los módulos, usando el protocolo de tiempo de red (NTP-Network Time Protocol) y el servidor NTP maestro 3.north-america.pool.ntp.org.

A continuación, en el módulo esclavo se ejecutó un programa que abrió un socket, configuró al módulo como servidor usando netcat, esperó la recepción de audio y video desde el módulo maestro y registro la hora de inicio de la recepción. En el módulo maestro se ejecutó un programa que abrió un socket, configuró al módulo como cliente, transmitió el audio y video al módulo esclavo y registro la hora de inicio de la transmisión. Esta prueba se repitió 30 ocasiones, resultando un tiempo de retraso entre la transmisión y recepción de 50 ms en promedio, bastante aceptable, sin percibirse congelamiento en los cuadros del video mostrados en la pantalla TFT. El último grupo de pruebas tuvo como objetivo determinar el impacto de la cantidad de cuadros por segundo, usados en la captura de la señal de video, en el retardo de la señal transmitida entre los módulos. Se siguió el mismo procedimiento del grupo de pruebas anterior y en cada transmisión se modificó el parámetro de captura de cuadros por segundo. El valor inicial usado fue 20 fps, incrementándose hasta el máximo soportado por la cámara de video de 120 fps. Se observó que el retardo en la transmisión fue 40 ms entre 20 y 40 fps y entre 40 y 60 fps el retardo fue 50 ms. Para valores mayores a 50 fps el retardo en promedio

resultante fue 52 ms, como se muestra en la gráfica de la figura 9. En base a los resultados anteriores se consideró usar 50 fps en este trabajo.

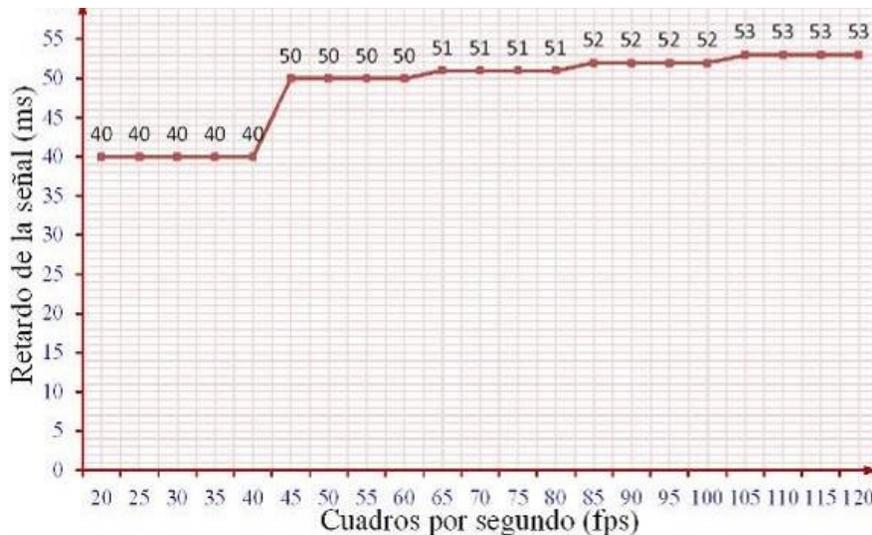


Figura 9 Impacto de fps en el tiempo de retardo de la transmisión de video.

4. Discusión

Se obtuvo como resultado un video portero basado en tarjetas Raspberry Pi 3 B+ que cumple con las funciones solicitadas y las especificaciones requeridas: confiable y fácil de instalar y usar. El costo aproximado del módulo maestro es \$70 USD y el del módulo esclavo \$50 USD, \$120 USD en total. El costo de video porteros actualmente disponibles en el mercado con características similares al desarrollado en este trabajo es en promedio \$250 USD. Se contempló inicialmente usar en este sistema una cámara web en lugar de la cámara para Raspberry V2 pero se descartó la idea por las siguientes razones: A) El costo de la cámara web más económica es casi tres veces el costo de la cámara de video usada en este trabajo, B) Los responsables del edificio no aceptaron la idea porque restaba privacidad a cada oficina, cualquier residente podría ver los mensajes que los visitantes estarían registrando para un residente y C) Se necesitaría un disco duro o de estado sólido para almacenar el audio y video que permanentemente registra la cámara web. El incluir la cámara web no está mal ni de más, incorpora funcionalidades, pero el costo del sistema aumenta y debe aprobarse por el usuario.

5. Conclusiones

El sistema construido fue probado y evaluado y, considerando los resultados obtenidos, se ha solicitado extender su operación a 7 oficinas adicionales y realizar una segunda versión que incorpore las siguientes funcionalidades:

- Un servidor web en los módulos esclavos para poder consultar desde la Internet el buzón de mensajes, comunicarse con el visitante y abrir remotamente la puerta.
- Un lector de huellas digitales en el módulo maestro que permita a personas registradas abrir la puerta.
- Un mecanismo de respaldo del buzón de mensajes.

Finalmente, el video portero desarrollado no fue solo una investigación ni un experimento ya actualmente se usa y resuelve una necesidad concreta.

6. Referencias

- [1] Arnold, O., Kirsch, L. & Schulz, A. An interactive concierge for independent living. IEEE 3rd Global Conference on Consumer Electronics (GCCE) Proceedings. Tokyo, Japan. Oct., 2014.
- [2] Banu, V. C., Costea, I. M. & Nemptanu, F. C. Intelligent video surveillance system. IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME) Proceedings. Constanta, Romania. Oct., 2017.
- [3] Chang, K. C., Liu, P. K. & Yu, C. S. Design of real-time video streaming and object tracking system for home care services. IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW) Proceedings. Nantou, Taiwan. May, 2016.
- [4] Conci, N., De Natale, F. G. & Messelodi, S. An integrated framework for video surveillance in complex environments. IEEE International Smart Cities Conference (ISC2) Proceedings. Trento, Italy. Sept., 2016.
- [5] Gatimu, K., Dhamodaran, A. & Johnson, T. Experimental study of low-latency HD VoD streaming using flexible dual TCP-UDP streaming protocol. 15th

- IEEE Annual Consumer Communications & Networking Conference (CCNC) Proceedings. Las Vegas, NV, USA. Jan., 2018.
- [6] Joshi, N. B. & Nalbalwar, S. L. A fall detection and alert system for an elderly using computer vision and Internet of Things. 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT) Proceedings. Bangalore, India. May, 2017.
- [7] Konda, K. R., Tefera, Y. T. & Conci, N. Real-time moving object detection and segmentation in H.264 video streams. IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB) Proceedings. Cagliari, Italy. June, 2017.
- [8] Myakala, P. R., Nalumachu, R. & Sharma, S. A low cost intelligent smart system for real time infant monitoring and cry detection. TENCON 2017-IEEE Region 10 Conference Proceedings. Penang, Malaysia. Nov., 2017.
- [9] Nightingale, J., Salva-Garcia, P. & Alcaraz Calero, J. M. 5G-QoE: QoE Modelling for Ultra-HD Video Streaming in 5G Networks. IEEE Transactions on Broadcasting. (Early Access), 2018.
- [10] Quadri, S. A. & Sathish, P. IoT based home automation and surveillance system. International Conference on Intelligent Computing and Control Systems (ICICCS) Proceedings. Madurai, India. June, 2017.
- [11] Watanabe, T., Mochizuki, R. & Kobayashi, T. HACCS: Home Appliance Control Concierge System: Extending Functions on Web Service. IEEE 37th Annual Computer Software and Applications Conference Proceedings. Kyoto, Japan. July, 2013.
- [12] Zahran, A. H., Raca, D. & Sreenan, C. ARBITER+: Adaptive Rate-Based InTElligent HTTP StReaming Algorithm for Mobile Networks. IEEE Transactions on Mobile Computing. (Early Access), April, 2018.
- [13] Zhang, X., Toni, L. & Frossard, P. Adaptive Streaming in Interactive Multiview Video Systems. IEEE Trans. on Circuits and Systems for Video Technology. (Early Access), March, 2018.