Universidad de Cuenca

Facultad de Ingeniería

Carrera de Electrónica y Telecomunicaciones

Diseño e implementación de un sistema de acceso inteligente para hogares

Trabajo de titulación previo a la obtención del título de Ingeniero en Electrónica y Telecomunicaciones.

Autores :

Carlos Andrés Guerrero Granda, C.I. 0302681911

Carlos Andrés Heredia Álvarez C.I. 0105402622

Director :

Ing.Darwin Fabián Astudillo Salinas, PhD C.I. 0103907036

Co-Director :

Ing. Andrés Marcelo Vázquez Rodas, PhD C.I. 0301496840

Cuenca - Ecuador

2018

Resumen

En la actualidad los sistemas de acceso inteligente permiten al administrador del hogar recibir una alerta cuando alguien se encuentra en la puerta del inmueble e incluso interactuar con el usuario desde un dispositivo inteligente. Estos sistemas reemplazan los métodos tradicionales de ingreso al hogar, puesto que brindan comodidad y facilidad en el acceso.

Una de las carencias de los sistemas tradicionales de acceso inteligente es la dependencia de objetos externos para la autentificación del usuario como tarjetas, llaves, etc. En este sentido gracias al desarrollo de las técnicas de reconocimiento facial se propone un sistema que permita gestionar el acceso al hogar mediante el reconocimiento facial de los usuarios, enfocándonos en la utilización de plataformas de código libre para la implementación electrónica y de software.

Este proyecto incluye el análisis de la problemática, un estudio del funcionamiento de las tecnologías involucradas en el reconocimiento facial y sus diversos métodos para lograr la misma. Además, se detalla el procedimiento usado para el proyecto, las plataformas de programación utilizadas y finalmente un análisis acerca de los diversos parámetros del sistema como la calidad de voz, video, latencia y sobretodo la efectividad del reconocimiento facial.

En concordancia con lo anterior los resultados demuestran que es factible la implementación de este tipo de sistemas ya que garantizan la seguridad en el acceso al hogar, además al hacer uso plataformas de software de libre el administrador puede realizar cambios en el sistema según sus requerimientos lo cual no sucede con plataformas propietarias.

Palabras Clave: Latencia, Domótica, Plataformas, Reconocimiento Facial.

Abstract

Currently, intelligent access systems allow the owner of the house to receive an alert when someone is at the door and even interact with the user from an intelligent device. These systems replace traditional methods of entry into the home, since they provide comfort and ease of access.

One of the shortcomings of traditional intelligent access systems is the dependence on external objects for user authentication such as: cards, keys, etc. In this sense, given the development of facial recognition techniques, we propose a system that allows access to the home to be managed through the facial recognition of users, focusing on the use of open source platforms for electronic and software implementation.

This project includes the analysis of the problem, a study of the mode of operation of the technologies involved in facial recognition and the methods to achieve it. In addition, it details the procedure used for the project, the programming platforms used and finally an analysis of the various parameters of the system such as: voice quality, video, latency and the effectiveness of facial recognition.

In accordance with the above, the results show that the implementation of this type of systems is feasible since they guarantee security in the access to the home, by using free software platforms the administrator can make changes in the system according to their requirements, which does not It happens with proprietary platforms.

Keywords: Latency, Plataforms, Domotic, Facial Recognition.

Índice general

Índice de figuras

Índice de tablas

Cláusula de Propiedad Intelectual

Yo, Carlos Andrés Heredia Álvarez, autor del trabajo de titulación "Diseño e Implementación de un Sistema de Acceso Inteligente para Hogares", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 23 de Octubre de 2018.

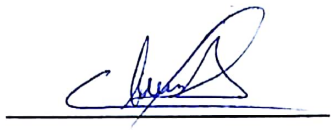
Carlos Andrés Heredia Álvarez

C.I: 0105402622

Cláusula de Propiedad Intelectual

Yo, Carlos Ernesto Guerrero Granda, autor del trabajo de titulación "Diseño e Implementación de un Sistema de Acceso Inteligente para Hogares", certifico que todas las ideas, opiniones y contenidos expuestos en la presente investigación son de exclusiva responsabilidad de su autor.

Cuenca, 23 de Octubre de 2018.



Carlos Ernesto Guerrero Granda

C.I: 0302681911

Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Yo, Carlos Andrés Heredia Álvarez en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "Diseño e Implementación de un Sistema de Acceso Inteligente para Hogares", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 23 de Octubre de 2018.

Carlos Andrés Heredia Álvarez

C.I:0105402622

Cláusula de licencia y autorización para publicación en el Repositorio Institucional

Yo, Carlos Ernesto Guerrero Granda en calidad de autor y titular de los derechos morales y patrimoniales del trabajo de titulación "Diseño e Implementación de un Sistema de Acceso Inteligente para Hogares", de conformidad con el Art. 114 del CÓDIGO ORGÁNICO DE LA ECONOMÍA SOCIAL DE LOS CONOCIMIENTOS, CREATIVIDAD E INNOVACIÓN reconozco a favor de la Universidad de Cuenca una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra, con fines estrictamente académicos.

Asimismo, autorizo a la Universidad de Cuenca para que realice la publicación de este trabajo de titulación en el repositorio institucional, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Cuenca, 23 de Octubre de 2018.

Carlos Ernesto Guerrero Granda

C.I: 0302681911

Dedicatoria

A mis padres y Dios,

El presente trabajo lo dedico a Dios, por ser el inspirador y darme fuerza para continuar en este proceso de obtener uno de los anhelos más deseados.
A mis padres Magdalena y Milton, por su amor, trabajo y sacrificio en todo este tiempo, gracias a ustedes he logrado llegar a este punto y alcanzar mis metas. Ha sido el orgullo y el privilegio de ser su hijo, son los mejores padres.
A mis hermanos Fernando y Valeria por estar siempre presentes y por el apoyo moral, que me brindaron a lo largo de esta etapa de mi vida.

Andrés Heredia

A mis madres

El presente trabajo está dedicado a mi familia por haber sido mi apoyo a lo largo de toda mi carrera universitaria y a lo largo de mi vida, en especial a mis madres Mariana y Nube quienes me brindaron su cariño y su apoyo incondicional, y a todas las personas que aportaron a mi formación tanto profesional y como ser humano.

Carlos Guerrero

Agradecimientos

A mis familiares.

A mis tías/os, mis abuelas y abuelo Mariana, Robertina y César, mis progenitores Milton y Magdalena y a todos aquellos que siempre me apoyaron, creyeron en mi y que participaron directa o indirectamente en la elaboración de esta tesis.
¡Gracias a ustedes!

Andrés Heredia

A mis familiares.

A mi familia, por haberme dado la oportunidad de formarme en esta prestigiosa universidad y haber sido mi apoyo durante todo este tiempo.
¡Gracias a ustedes!

Carlos Guerrero

A los maestros.

A todos quienes marcaron cada etapa de mi vida universitaria pero especialmente a los Ingenieros Fabián Astudillo y Andrés Vázquez por su gran apoyo y motivación para la elaboración de esta tesis.

Los Autores

Abreviaciones y Acrónimos

ACK Acknowledgement. 12
AMR Adaptive Multi-Rate. 36
API Application Programming Interface. 33

CPU Central Processing Unit. 18, 29, 35
CSI Camera Serial Interface. 33, 43

DNS Domain Name System. 13

FR Full Reference. 14

GND Ground. 32
GPIO General Purpose Input/Output. 24, 31
GPU Graphics Processing Unit. 29
GSM Global System for Mobile communications. 36

HDMI High-Definition Multimedia Interface. 33
HVS Human Visual System. 13, 15, 16

IAX Inter-Asterisk eXchange Protocol. 10, 35, 63
IDE Integrated Development Environment. 36
IEC International Electrotechnical Commission. 17
IEEE Institute of Electrical and Electronics Engineers. 17
IETF Internet Engineering Task Force. 10
INEC Instituto Nacional de Estadística y Censos. 1
IoT Internet of Things. 4, 28, 33
IP Internet Protocol. 4, 11, 13, 18, 28, 35
IPv6 Internet Protocol version 6. 36
ISP Internet Service Provider. 36
ITU International Telecommunication Union. 14
ITU-T International Telecommunication Union - Telecommunication Standardization Sector. 10, 17, 19

JPEG Joint Photographic Experts Group. 17
JSON JavaScript Object Notation. 34, 39

LBPH Local Binary Patterns Histograms. 8, 9, 19, 24

MCTF International Electrotechnical Commission. 17
MGCP Media Gateway Controller Protocol. 35
MIPI Mobile Industry Processor Interface. 33
MJPEG Motion Joint Photographic Experts Group. 17, 20, 56
MOS Mean Opinion Score. 19, 53, 54

NR No-Reference. 14

OpenCV Open Source Computer Vision Library. 7, 8, 22, 24, 44, 49, 50, 59, 60
OTG On-The-Go. 32

PBX Private Branch Exchange. 10, 35
PSNR Peak Signal-to-Noise Ratio. 15, 55–57

QoS Quality Of Service. 49

RR Reduced Reference. 14
RTCP Real Transmission Control Protocol. 19
RTP Real Time Protocol. 12, 19, 35

SDK Software Development Kit. 34
SIP Session Initiation Protocol. 10–13, 19, 20, 35, 36, 54, 63, 79
SRTP Secure Real-time Transport Protocol. 36
SSIM Structural Similarity Index. 15, 16, 55, 56

TCP Transmission Control Protocol. 37
TLS Transport Layer Security. 36
TTL Time To Live. 18

URI Uniform Resource Identifier. 12, 13
USB Universal Serial Bus. 32, 33, 43

VCEG Video Coding Experts Group. 17
VoIP Voice Over Internet Protocol. 4, 10, 18, 19, 35, 36, 49, 53, 54, 60, 61, 63
VP8 Video Compression Format. 17

WiFi Wireless Fidelity. 4, 33, 43
WLAN Wireless Local Area Network. 16, 17

XMPP Extensible Messaging and Presence Protocol. 10

ZRTP Zimmermann Real-Time Transport Protocol. 36

Capítulo 1

Introducción

En este capítulo, se presenta el problema a tratar, así como la justificación, el alcance y los objetivos general y específicos a cumplir durante esta investigación.

1.1. Identificación del Problema

Según el INEC en la Encuesta de Victimización y Percepción de Inseguridad del 2011, el robo a viviendas en el sector urbano se ubicó en alrededor del 4 %, es decir 4 de cada 100 hogares han sido victimas de robo a viviendas, he ahi la necesidad de contar con sistemas que nos permitan controlar el acceso al mismo [14].

El término domótica hace referencia a la tecnología que se encuentra adaptada para controlar y automatizar los hogares, constituye el dominio y la supervisión de todos los elementos que integran una edificación compuesta por oficinas o una vivienda [15].

Las técnicas de automatización actuales se implementan mediante un microcontrolador o una computadora. El microcontrolador no puede ejecutar múltiples programas a la vez, ademas es difícil controlar los dispositivos que se conectan al mismo. Sin embargo, gracias al avance tecnológico se tiene computadoras de bolsillo, la Raspberry Pi [16] es una de ellas al ser de una sola placa y de bajo consumo energético es ideal para este tipo de proyectos.

Es así como, labores tan comunes como abrir una ventana o la puerta se convierten en un problema; pero con la implementación de la domótica en el hogar se disminuye el grado de dificultad y aumenta la seguridad al realizar estas tareas. En el mercado actual se cuenta con un gran número de aplicaciones de muy alto nivel sobre el tema, pero no ofrecen la opción de reconocimiento facial para el ingreso de los usuarios a sus hogares, limitándoles a llevar alguna

credencial o llaves para entrar a mismo. Estos sistemas en su mayoría tienen costos elevados y están pensados más en generar una experiencia de seguridad para los domicilios.

1.2. Justificación

El estilo de vida actual es muy acelerado y gracias al avance de la tecnología nos ha permitido elevar el nivel de *confort* en nuestros hogares, permitiendo que los electrodomésticos y artefactos de uso cotidiano del hogar sean capaces de realizar tareas de forma casi autónoma.

En este sentido la domótica se encarga de la integración y de la regulación de todos estos sistemas, de esta manera la casa es capaz de detectar la presencia de personas, temperatura, nivel de luz, etc, y al mismo tiempo es capaz de interactuar con nosotros por diversos medios como teléfonos inteligentes(smartphones), pantallas táctiles, PC, etc.

Este proyecto está diseñado para brindar confort y accesibilidad al momento de acceder al hogar, el objetivo es permitir al propietario la posibilidad de acceder a una vivienda, mediante el uso de un smartphone o tableta y decidir en tiempo real si permitir o no el acceso del usuario. Se extiende dicho control al acceso del garaje del hogar. El trabajo está enmarcado dentro del proyecto: "Prototipo de expansión de funcionalidades para dispositivos de automatización industrial clásicos utilizando plataformas embebidas de bajo costo", el cual actualmente está en desarrollo por parte del departamento de Ingeniería Eléctrica, Electrónica y Telecomunicaciones de la Universidad de Cuenca. El aporte antes mencionado busca determinar si el uso de plataformas de bajo costo y de código abierto para el control de aplicaciones domésticas provee resultados satisfactorios, lo cual brinda la posibilidad de extender su uso a posibles áreas de la automatización industrial.

1.3. Alcance

Se busca que el propietario de la vivienda disponga del control completo sobre el acceso al hogar siendo capaz de establecer una llamada de voz o vídeo mediante un smartphone o tablet con el sistema de acceso para distinguir a la persona que desea ingresar al domicilio [17],[18],[19].

La contribución de este trabajo se centra en el desarrollo de un prototipo e implementación de un diseño domótico para acceso al hogar (hogar y garaje), se busca que el propietario de la vivienda disponga del control completo sobre el acceso al hogar siendo capaz de establecer una llamada de voz o vídeo mediante un smartphone o tablet con el sistema de acceso para distinguir a la persona que desea ingresar al domicilio.

Se propone analizar varios métodos de notificación de acceso al usuario entre los que se destacan streaming de vídeo, videollamada o envío de varias fotos simultáneas, de todos los anteriormente

mencionados se analizará la factibilidad técnica de llevarlo a cabo con el fin de tener el mejor rendimiento para el sistema con la menor complejidad [20][21]. Para poder comandar los actuadores se analizará la factibilidad de implementar un servidor web en la plataforma de hardware escogida.

Se propone efectuar diferentes experimentos para evaluar la funcionalidad del sistema los cuales tienen en consideración variables como la latencia, pérdida de paquetes, ancho de banda y protocolo de comunicación utilizado en la conexión. Se plantean pruebas que permitan determinar el retardo de paquetes entre el sistema de acceso principal y el dispositivo Android, analizar la calidad de la llamada de voz o video extremo a extremo haciendo uso de métodos subjetivos u objetivos como los propuestos en [22].

Se plantea analizar diferentes plataformas de código libre para la implementación electrónica y de software, se involucra, además, el análisis de técnicas enfocadas en el reconocimiento de rostros y comunicación inalámbrica entre dispositivos [23].

1.4. Objetivos

1.4.1. Objetivo General

Diseñar e implementar un sistema de acceso inteligente para hogares usando técnicas de procesamiento digital de imágenes.

1.4.2. Objetivos Específicos

- Realizar un análisis de factibilidad usando varios métodos de notificación entre el sistema de acceso y los clientes móviles (propietarios de la vivienda).
- Usar técnicas de reconocimiento facial para efectuar el control de acceso en el hogar.
- Implementar un sistema de control de acceso remoto para la apertura de la puerta de un garaje.
- Analizar diversas plataformas de software libre para la implementación electrónica y de software.
- Realizar experimentos con el objetivo de analizar la calidad del método de notificación al usuario
- Comparar el sistema implementado con tecnologías de acceso para hogares tradicionales.

Capítulo 2

Fundamentos Teóricos

Este capítulo presenta las diferentes tecnologías involucradas dentro del contexto de un sistema de acceso inteligente para hogares, partiendo desde una base general en la que se explican conceptos tales como: métodos y técnicas para reconocimiento facial, VoIP, *videostreaming* y técnicas de codificación de video.

2.1. Introducción

En la actualidad se define a la domótica como el conjunto de tecnologías aplicadas al control y automatización inteligente de una vivienda [24]. La domótica tiene una estrecha relación con el IoT, aunque no necesariamente implica la conexión de los dispositivos a la misma. Los elementos domóticos generan o procesan información la cual es entregada a los usuarios de manera sencilla.

Uno de los beneficios de la domótica y en el que está enmarcado este proyecto es la seguridad, la cual constituye un factor importante que evita el acceso no permitido de usuarios no registrados al hogar. Como se puede apreciar en la Figura 2.1 se dispone de varios dispositivos domóticos que brindan seguridad: 1.) Cámaras IP; 2.) Sensor de movimiento; 3.) Alarmas en puertas y ventanas; y, 4.) Porteros inteligentes para acceso.

Los porteros inteligentes son dispositivos que por lo general integran un timbre y una cámara que funcionan cuando un usuario se acerca al dispositivo. Se busca que estos dispositivos funcionen en conjunto con una red WiFi y un teléfono inteligente al cual se le enviará video en tiempo real, brindando al dueño de casa la posibilidad de visualizar al visitante que se encuentra en su puerta. Además de enviar video, los porteros inteligentes tienen la capacidad de ofrecer algunas otras características de seguridad y comodidad, como comunicación por voz, infrarrojos/visión nocturna, detección de movimiento y control de acceso al hogar de forma remota. Si bien un timbre tradicional podría ser suficiente, existen una serie de instancias en las que un portero

Figura 2.1: Domótica aplicada a la seguridad [1].

inteligente mejoraría el confort y eliminaría la necesidad del uso de llaves para ingresar al hogar. Por ejemplo, algunas situaciones en las que instalar un dispositivo de acceso inteligente resulta útil se describen a continuación:

- Cuando un usuario se encuentra en la puerta, pero no se desea permitir el acceso hasta que se sepa quién está del otro lado.
- Cuando el dueño de casa no se encuentra cerca de la puerta y desea mantener un control remoto de la misma.
- Cuando no se esperan invitados.

A este tipo de dispositivos se les puede agregar una etapa de reconocimiento facial para ejercer control sobre la apertura y cierre de la puerta del hogar, es decir añadir la característica para apertura automática en caso de efectuar el reconocimiento del rostro de las personas que viven en el hogar. El reconocimiento de rostros es una tarea que los seres humanos lo hacemos de forma rutinaria y sin mucho esfuerzo. En la actualidad con el aumento de la capacidad computacional lograda en los dispositivos móviles o PCs de bajo costo, ha surgido un gran interés en el procesamiento digital de imágenes para incluirlo en una variedad de aplicaciones tales como: autentificación biométrica, interacción entre humano y máquina, etc [25].

El reconocimiento facial presenta una ventaja sobre los métodos tradicionales como lector de huella o lector de iris debido a que resulta más natural para el usuario. En la actualidad el término biométrico se refiere al estudio relacionado con la autentificación, y se define como el acto de establecer o confirmar un hecho como auténtico, es decir, trata de discernir si las asunciones sobre él o algo son verdaderas [26]. A continuación, se listan varias técnicas biométricas categorizadas:

1. **Biométricas físicas:** incluye mediciones físicas y caracterización del rostro, lector de huella, escaneo de iris, etc.
2. **Comportamientos biométricos:** involucra el rendimiento de una persona durante la ejecución de cierta tarea específica.
3. **Químicos biométricos:** involucra la medición de parámetros como el olor y su composición química en una persona.

El término reconocimiento de rostros corresponde a dos escenarios el uno llamado identificación y el otro autentificación o verificación. En cualquier escenario los rostros de las personas se añaden a una base de datos en el sistema, este conjunto se le conoce como galería. Luego las imágenes de otras personas son comparadas con las de la galería en un escenario de reconocimiento una contra todas para encontrar la que mejor coincida por encima de un umbral [26]. La técnica de reconocimiento de rostros efectúa las siguientes operaciones:

- Detección de rostros.
- Reconocimiento de rostros (como se muestra en la Figura 2.2).

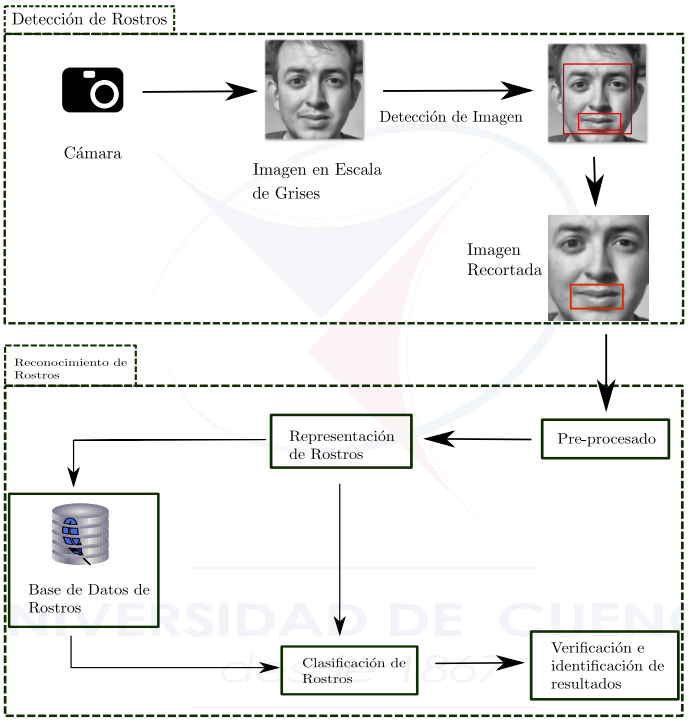


Figura 2.2: Diagrama de un sistema de detección/reconocimiento de rostros [2].

El reconocimiento de rostros es el paso inicial para localizar y extraer la región de la cara de la imagen. En esta etapa se toma el rostro extraído de la etapa de detección y se compara con una base de datos de imágenes previamente registradas. La etapa final es la de identificación y verificación. La identificación es el proceso de comparar una cara con dos o más conjuntos de rostros para determinar la coincidencia más probable, en tanto la verificación de rostros es el proceso de comparar un rostro de prueba con otro en la base de datos resultando en la aprobación o rechazo [26].

La tecnología de reconocimiento facial ha tenido muchos avances desde que el método Eigenfaces fue propuesto, en circunstancias donde las condiciones de luz o las expresiones faciales son las mismas o parecidas se puede tener altos índices de reconocimiento, especialmente si la base de

datos contiene un gran número de imágenes de rostros [26].

En condiciones muy controladas de imagen como las fotografías de pasaportes en [2] se registró que el índice de error es alto. Entrenar un sistema bajo ciertas condiciones específicas y hacer que sea posible la detección bajo condiciones diferentes es un reto, la principal limitación es el intensivo entrenamiento en el cual un conjunto de píxeles es etiquetado tomando así importantes recursos de hardware.

2.2. Reconocedores de Rostros

La detección de rostros tiene muchas aplicaciones en la comunicación de visión por computadora y sistemas de control automático. La detección de rostros es un método para encontrar un rostro en una imagen con varios atributos, requiere reconocimiento de expresiones, rastreo facial y estimación de postura. En una imagen la dificultad de detectar un rostro reside en que el mismo puede estar rígido o cambiar de tamaño, forma, color, etc [27].

El reconocimiento de rostros es un método de procesamiento de imágenes que localiza el rostro en una fotografía y extrae las características más relevantes para posteriormente compararlas con características similares de una base de datos utilizada en el entrenamiento del sistema. Tanto la detección como el reconocimiento de rostros se pueden realizar mediante el uso de librerías en el software OpenCV [28].

Detección de rostros mediante el clasificador Haar Cascade

La detección de objetos usando características de Haar es un enfoque basado en el aprendizaje automático en el que una función en cascada se entrena a partir de muchas imágenes positivas y negativas. El algoritmo necesita varias imágenes con caras y sin caras para entrenar al clasificador, las características necesitan ser extraídas del clasificador. Las características de Haar se muestran en la Figura 2.3.

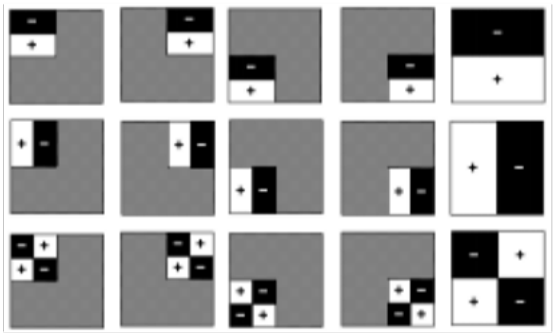


Figura 2.3: Filtros Haar rotados, trasladados y con cambios de escala [3]

2.3. Reconocimiento de Rostros

OpenCV incorpora la función de reconocimiento de rostros por defecto, se debe indicar en la función la ubicación de los rostros y se efectuará la evaluación y reconocimiento de caras en las imágenes. Se hace énfasis en el reconocedor de rostros LBPH el cual es usado en este proyecto. OpenCV incorpora otros reconocedores como: FisherFaces y Eingefaces, los cuales al igual que LBPH proveen buenos resultados aunque no se consideran debido a su alta sensibilidad lumínica.

- **Reconocedor Local Binary Patterns Histograms (LBPH)**
  Local Binary Pattern Histogram tiene un enfoque distinto a los algoritmos, Eigenfaces y Fisherfaces. En estos dos últimos se trata la imagen como un vector de datos en un espacio de muy alta dimensionalidad. En cambio LBPH describe las características locales de cada rostro. De esta manera, las características extraídas tendrán una baja dimensionalidad. Se basa en ir tomando pixeles vecinos respecto de un pixel central, el cual establece un valor de umbral. Si el valor del pixel central es menor, se etiqueta con un cero, y si es mayor se etiqueta con un uno, dicho proceso se esquematiza en la Figura 2.4, Estos valores obtenidos se concatenan para formar un número binario, que posteriormente se convierte en decimal, el cual será el nuevo valor del pixel. Por cada región se obtiene un histograma, que posteriormente se concatena para obtener una representación del rostro [4].
  Se obtiene un histograma por cada una de las partes extraídas, y posteriormente se combinan como se muestra en la Figura 2.5:
- **Reconocedor EigenFaces**
  El reconocedor EigenFaces observa todas las imágenes de entrenamiento de todas las personas como un todo y trata de extraer los componentes que son los más importantes y descarta el resto de características de los rostros. De esta forma, no solo extrae los componentes importantes de los datos de entrenamiento, sino que también ahorra memoria descartando los componentes menos significativos.

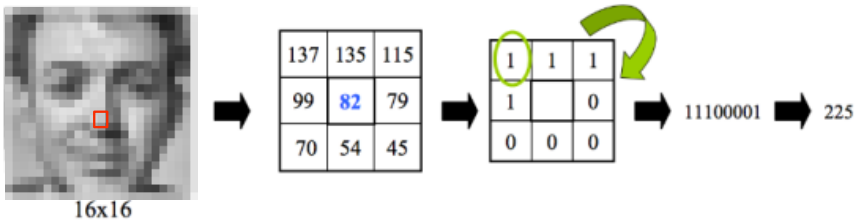


Figura 2.4: Funcionamiento algoritmo reconocedor LBPH [4].

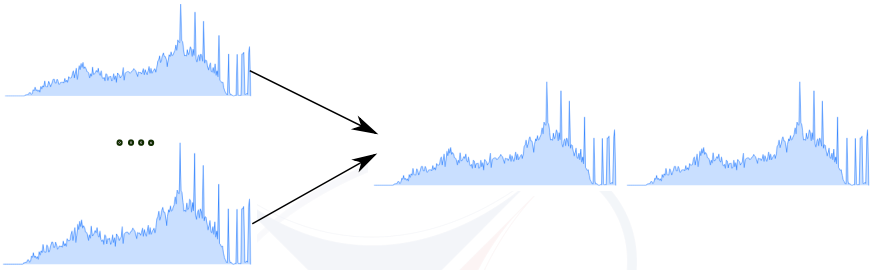


Figura 2.5: Fusión de histogramas reconocedor LBPH [4].

- **Reconocedor FisherFaces**
  Este algoritmo es una versión mejorada del reconocedor de caras EigenFaces. El algoritmo de Fisherfaces, en lugar de extraer características útiles que representan todas las caras de todas las personas, extrae características útiles que discriminan a una persona de otras. De esta manera, las características de una persona no dominan sobre las demás [27].

2.3.1. Comparación entre Reconocedores

En [29] se efectuó una comparación entre los algoritmos de los diferentes reconocedores de rostros para determinar el rendimiento que se puede obtener con cada uno de ellos. Se llega a la conclusión de que tanto los algoritmos Eigenfaces y Fisherfaces son muy sensibles a los cambios de nivel de los pixeles es decir a la iluminación, al cambio de expresiones faciales o simplemente a la variación de poses. En la Tabla 2.1 se aprecia la comparación de los algoritmos usando dos escenarios diferentes, el primero en una plataforma Intel y el segundo en una plataforma ARM, siendo LBPH el algoritmo con mas precisión.

Tabla 2.1: Comparación algoritmos de reconocimiento facial

Método	Experimento 1			Experimento 2		
	Verdaderos Positivos	Falsos Positivos	Precisión (%)	Verdaderos Positivos	Falsos Positivos	Precisión (%)
Eigenfaces	509	1028	33	624	47	93
Fisherfaces	519	1018	34	540	93	85
LBPH	679	858	44	594	39	94

2.4. Protocolos VoIP

Los protocolos VoIP constituyen los lenguajes que son usados por los diferentes dispositivos VoIP para su conexión. La eficacia y complejidad de la comunicación va en función del protocolo VoIP implementado, los protocolos de uso común son:

- H.323: el cual es definido por la ITU-T [30].
- SIP: definido por la IETF [31].
- Megaco o H.248 [32].
- MiNet: protocolo propiedad de Mitel [33].
- CorNet-IP: protocolo propiedad de Siemens.
- IAX: protocolo originario para comunicaciones PBX. Asterisk.
- Jingle: protocolo abierto utilizado en tecnología XMPP.

Cada uno de los protocolos antes descritos poseen gran cantidad de documentación que trata acerca de su estructura y funcionamiento, para nuestro caso resulta de gran interés el protocolo SIP el cual será mencionado en los Capítulos 4 y 6.

2.4.1. Protocolo SIP (Session Initiation Protocol)

SIP es un protocolo de señalización que brinda mensajería instantánea. Fue desarrollado para configurar, modificar y desmontar sesiones multimedia, solicitar y entregar mensajes instantáneos a través de Internet [5].

Como su nombre lo indica, el protocolo permite que dos puntos finales establezcan sesiones de medios entre sí. Las principales funciones de señalización del protocolo son las siguientes:

- Ubicación de un punto final.
- Contacto con un punto final para determinar la disponibilidad de establecer una sesión.
- Intercambio de información de medios para permitir que se establezca la sesión.
- Modificación de sesiones de medios existentes.
- Desmontaje de sesiones de medios existentes.

La Figura 2.6 muestra la jerarquía en la pila del protocolo multimedia de Internet con la cual trabaja SIP:

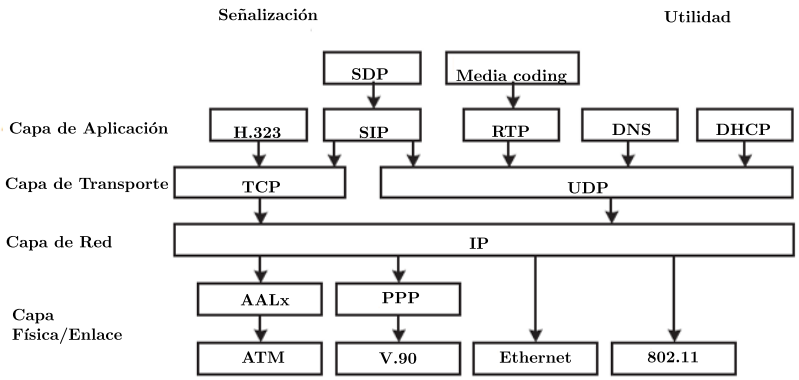


Figura 2.6: Pila de protocolos multimedia de internet [5]

- **Establecimiento de una sesión SIP:** La Figura 2.7 muestra un ejemplo simple de cómo se establece una sesión SIP. Se asume que ambos dispositivos están conectados a

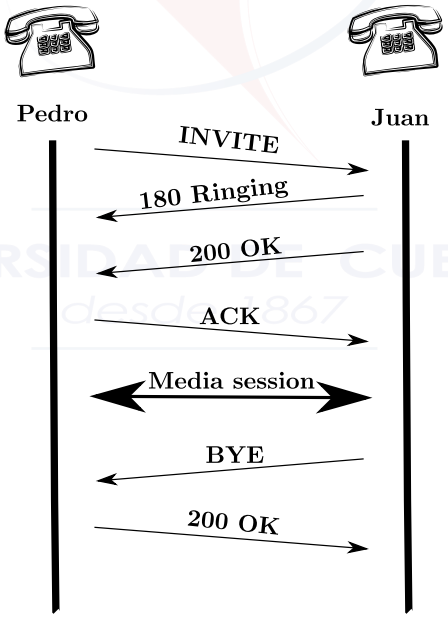


Figura 2.7: Ejemplo de establecimiento de sesión SIP [5]

una red IP, y cada uno conoce la dirección IP del otro.

El proceso se describe a continuación:

1. Comienza el intercambio de mensajes enviando un mensaje SIP: `INVITE` a la parte llamada, Juan. La solicitud `INVITE` contiene detalles como el tipo de sesión o llamada

que se solicita, podría ser una sesión de voz simple (audio), una sesión multimedia como una videoconferencia, o una sesión de juego.

2. `180 Ringing` es enviado en respuesta a `INVITE`. Este mensaje indica que la parte llamada ha recibido el `INVITE` y que se están llevando a cabo las alertas. La alerta podría estar sonando un teléfono, mostrando un mensaje en una pantalla, o cualquier otro método para atraer la atención [5].
3. Cuando el usuario decide aceptar la llamada, se envía una respuesta de `200 OK`. La respuesta también indica que el tipo de sesión de medios propuesta por la persona que llama es aceptada. El `200 OK` es un ejemplo de respuesta de *clase de éxito*.
4. El último paso es confirmar la sesión con un ACK. La confirmación significa que la parte que inicia la llamada recibió con éxito una respuesta del receptor, lo cual permite que la sesión de medios se establezca usando algún protocolo como por ejemplo RTP.
5. Una vez establecida la sesión de medios, la parte llamada origina la solicitud `BYE` y actúa como cliente SIP, mientras que el origen de la llamada actúa como el servidor SIP cuando responde. Esta es la razón por la que un dispositivo habilitado para SIP debe contener tanto el servidor SIP como el software cliente SIP: durante una sesión típica, ambos son necesarios. La respuesta de confirmación al `BYE` es un `200 OK` [5].

El esquema de direccionamiento es parte de una familia de direcciones de Internet conocidas como URI. Los SIP URI permiten manejar números de teléfono, parámetros de transporte y otros elementos necesarios en el establecimiento de una llamada, el esquema para una llamada SIP con servidor proxy se describe en la Figura 2.8.

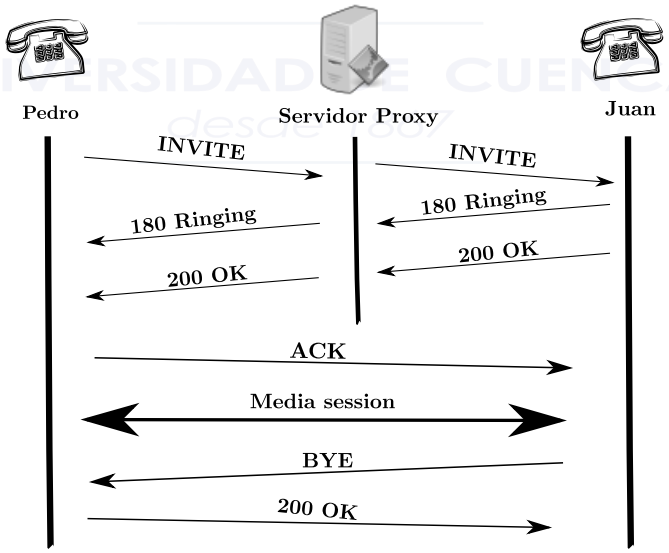


Figura 2.8: Ejemplo de llamada SIP con servidor proxy [5].

Un SIP URI se describe como el esquema de direccionamiento SIP, o la identificación de

una cadena de caracteres, para llamar a otra persona a través de SIP, permite establecer el número de teléfono de un usuario, y tiene un formato similar al del correo electrónico. El formato es `sip:user@host`, o algunas veces `sip:user@host.port` [34].
Un proxy SIP no configura ni finaliza sesiones, sino que se ubica en medio de un intercambio de mensajes SIP, recibe mensajes y los reenvía, puede haber múltiples proxies en una ruta de señalización.
Su modo de operación se describe de forma breve a continuación:

1. Para el esquema antes descrito Pedro no sabe exactamente dónde está conectado actualmente Juan y qué dispositivo está utilizando, se usa un servidor proxy SIP para enrutar el paquete `INVITE`. Primero, se realiza una búsqueda DNS del nombre de dominio SIP URI de Juan, que devuelve la dirección IP del servidor proxy que maneja ese dominio. El `INVITE` se envía a esa dirección IP [5].
2. El proxy busca el URI SIP en el URI de solicitud (`sip: werner. juan@munich.de`) en su base de datos y localiza a Juan. Este proceso se ejecuta en dos pasos:
   - Se ejecuta una búsqueda DNS por agente de usuario para localizar la dirección IP del proxy; posteriormente se ejecuta una búsqueda en la base de datos del proxy para localizar la dirección IP [5].
   - El `INVITE` se reenvía a la dirección IP de Juan con la adición de un segundo campo de encabezado marcado con la dirección del proxy [5].
3. A partir de la presencia de dos campos de encabezado Juan sabe que el `INVITE` se enruta a través de un servidor proxy. Una vez recibido el `INVITE`, Juan envía una respuesta de llamada `180 Ringing` al proxy [5].
4. El servidor `PROXY` envía la respuesta a la dirección en el primer campo de encabezado [5].
5. Un proceso similar al ya descrito permite que la llamada sea aceptada por Juan, que envía una respuesta de `200 OK` estableciendo la conexión.

2.5. Medidas de Calidad de Video

Existen diversas técnicas que permiten evaluar la calidad de una secuencia de video. Estas técnicas se dividen en dos categorías:

- Medidas Subjetivas de Calidad (2.5.1).
- Medidas Objetivas de Calidad (2.5.2).

2.5.1. Medidas Subjetivas de Calidad

Se denomina medida subjetiva de calidad de video cuando la evaluación es efectuada por uno o varios observadores, los cuales en función de su criterio califican a cada secuencia. Permite obtener una estimación de calidad según el HVS y constituye una medida de calidad muy

fiable.Según la ITU se recomienda entre 4 y 40 observadores para obtener una valoración de calidad de video aceptable [35]. Una cantidad de observadores inferior a 4 no brinda garantías estadísticas de resultados fiables, y más de 40 observadores producen una mejora pequeña en la estimación.

2.5.2. Medidas Objetivas de Calidad

Son valiosas porque brindan las herramientas para puntuar de forma automática, vía software o hardware un determinado video sin convocar a personas especializados que efectúen una evaluación subjetiva de calidad. La finalidad de la investigación objetiva de video es diseñar métricas de calidad que puedan predecir automáticamente la calidad de video percibida [36].

Clasificación de las medidas objetivas de calidad: Las medidas objetivas de calidad de video se pueden clasificar en función de la existencia de una única secuencia de video distorsionada, o si se dispone de un video de referencia original y sin distorsión, y de acuerdo a la cantidad de información usada o disponible de ambas secuencias. Así dichos métodos pueden ser: FR, RR y NR, para este proyecto se considera el método de medición de calidad de video FR:

1. **Método FR:** Este tipo de método basa su cálculo de calidad de video a partir de la comparación del video distorsionado y el video original capturado en la fuente, debido a ello suele proveer resultados con una confiabilidad alta, aunque requiere un procesamiento computacional elevado. La Figura 2.9 muestra el esquema de operación :

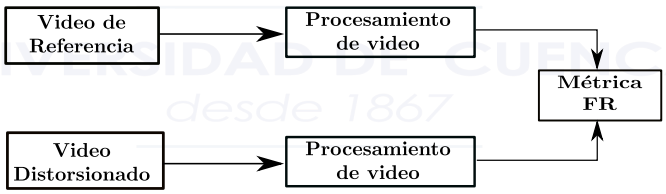


Figura 2.9: Diagrama de métodos Full Reference [6].

2. **Método RR:** Este tipo de métodos no efectúan una comparación directa de las dos señales de video (original y distorsionada), por el contrario, toman características o parámetros de ellas, y miden las diferencias entre éstas [6].
3. **Método NR:** Este método no hace uso del video original para estimar la calidad de la secuencia de video, toma únicamente la secuencia distorsionada y sobre ella aplica un conjunto de métricas estimadoras de calidad [6].

Una vez definidas las distintas medidas de calidad de video se describe a continuación cada uno de los métodos involucrados dentro de las mismas. Se clasifican en tres grupos en función de su estructura:

- Métricas tradicionales.
- Métricas orientadas a características naturales visuales.
- Métricas orientadas al HVS.

Se describen las medidas más utilizadas para evaluar la calidad de una secuencia de video en lo referente a medidas objetivas:

1. **PSNR**
   Es considerada como una de las medidas de calidad de video más populares y eficientes computacionalmente. Esta métrica mide la relación entre la potencia máxima posible de una señal y la potencia del ruido corrupto que afecta la fidelidad de su representación. En relación a la calidad de video, se define al ruido como el error cuadrático medio entre la señal original y la distorsionada, se expresa en unidades logarítmicas [6].

$$PSNR = 10 \cdot \log_{10}\left(\frac{L^2}{MSE}\right) \tag{2.1}$$

   Donde:

   - L es el máximo valor que puede tomar la señal.
   - MSE es el error cuadrático medio, y se define mediante:

$$MSE = \frac{1}{3MN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\sum_{k=1}^{3}[O(i,j,k) - D(i,j,k)]^2 \tag{2.2}$$

   - M y N son las dimensiones de cada uno de los cuadros de video.
   - O representa un frame de la secuencia original.
   - D representa un frame de la secuencia distorsionada.

   Las expresiones matemáticas antes descritas se aplican a cada cuadro del video, posteriormente se promedian los resultados con lo que se obtiene un único índice de calidad para la secuencia de video.

2. **SSIM**
   Tiene en consideración la relación que existe entre las señales de imagen y su estructura, lo cual implica que los píxeles de una imagen dependen unos de otros, en especial aquellos que se hallan próximos en la imagen.
   La diferencia con respecto a PSNR es que SSIM es un modelo basado en la percepción que considera la degradación de la imagen como un cambio percibido en la información estructural, al tiempo que incorpora importantes fenómenos perceptuales incluyendo términos de enmascaramiento de contraste. La información estructural hace referencia a la dependencia entre pixeles cuando están espacialmente cerca. Estas dependencias llevan información importante sobre la estructura de los objetos en la escena visual [37]. Se defi-

ne un conjunto de tres índices para comparar contraste, luminancia y la estructura tanto de la imagen original como distorsionada:

$$l(i,j) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{2.3}$$

$$c(i,j) = \frac{2\sigma_x\sigma_y + C_1}{\sigma_x^2 + \sigma_y^2 + C_2} \tag{2.4}$$

$$s(i,j) = \frac{\sigma_{xy} + C_3}{\sigma_{xy} + C_3} \tag{2.5}$$

Donde:
μ denota la media.
σ denota la desviación típica.
σ_{xy} denota la co-varianza.
Los sub-índices xy denotan tanto a las imágenes original y distorsionada.
C_1, C_2 y C_3 son constantes para evitar división por cero.
Mediante la multiplicación de estos tres índices se obtiene la medida SSIM para el pixel en la posición (i,j):

$$SSIM(i.j) = \frac{(2\mu_x\mu_y + C_1)(2\sigma xy + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2.6}$$

Esta ecuación por lo general es aplicada de forma individual a cada pixel de la imagen en cada plano de color, se hace uso de ventana de 8×8, la cual se desplaza píxel a píxel.

3. **MS-SSIM**
Se basa en la suposición de que el HVS está altamente adaptado para extraer información estructural de la escena, y, por lo tanto, una medida de similitud estructural puede proporcionar una buena aproximación a la calidad de imagen percibida [38].

2.6. Video Streaming

La transmisión multimedia a través de redes inalámbricas se está convirtiendo en un servicio cada vez más importante. Esta tendencia incluye el desarrollo de redes WLAN que permiten a los usuarios acceder a diversos servicios, incluidos aquellos que distribuyen contenido multimedia como video o audio [39].

El rendimiento de la transmisión de video a través de redes WLAN no solo está influenciado por el estado de la red, depende mucho de los parámetros del video como su codificación la cual puede o no usar alguna técnica de compresión, la calidad de las imágenes, el bitrate del video y la compresión del audio [39].

Hay muchos problemas relacionados con el rendimiento que se asocian a entrega de contenido multimedia en el estándar IEEE 802.11. Entre los más significativos se tienen:

- Bajas tasas de entrega (teóricamente hasta 11Mbps para IEEE 802.11b, en la práctica solo se puede lograr un rendimiento máximo de aproximadamente 6Mbps).
- Altas tasas de error debido a las características de los medios.
- Mecanismos de *backoff* en la transmisión.
- Colisión de paquetes.
- Atenuación de señal con distancia.
- Interferencia de señal.

Hay una gran cantidad y diversidad de variables que deben considerarse para la transmisión de video, cada una de las cuales tiene un impacto en el rendimiento y el comportamiento de la transmisión de video en un entorno WLAN. Entre las variables a considerar se destacan:

- El contenido y la complejidad del contenido.
- El esquema de compresión
- La configuración de codificación.
- El método de entrega del video.
- El servidor de transmisión utilizado
- El algoritmo de adaptación empleado por el servidor.

2.6.1. Codecs de Video

Un códec, o codificador/decodificador, es una herramienta que procesa video y lo almacena en una secuencia de bytes. Los codecs usan algoritmos para reducir el tamaño del archivo de audio o video, y luego descomprimirlo cuando sea necesario [40].

Hay diferentes tipos de codecs, y cada uno usa una tecnología diferente para codificar y reducir el tamaño del archivo de video para la aplicación deseada. Dependiendo del códec, la codificación se produce de dos maneras: con pérdida (también denominado lossy) o sin pérdida [40].

A continuación, se describen las características principales de los codecs de video más eficientes: H.264, VP8 y MJPEG, el cual es utilizado en este proyecto [41].

- H.264: es el estándar de codificación de video, desarrollado conjuntamente por la ITU-T, VCEG y la IEC, fue desarrollado con la intención de lograr una alta compresión de datos sin comprometer la calidad visual del video [42].
  En una compresión de video basada en bloques en H.264, cada uno de los cuadros de video se divide en bloques. Cada bloque se codifica utilizando una combinación MCTF a partir de la señal transformada, el tamaño de bloque recomendado es de 8×8 [42].
- MJPEG: Es un algoritmo con ciertas deficiencias en lo referente a almacenamiento de video. Cada fotograma es una imagen JPEG, la cual se almacena sola, independientemente

del fotograma precedente y siguiente [43].
Sin embargo, es simple de codificar y muy rápido. Su uso se recomienda para transmisiones de video en vivo en las que el contenido debe ser almacenado en el instante de la transmisión, de gran utilidad en sistemas que no disponen de suficiente potencia de CPU para codificar contenido en vivo a un mejor formato, o suficiente ancho de banda de disco duro para guardar el video sin procesar [43].
Ofrece un gran compatibilidad con los navegadores web y reproductores de video más populares,esto debido a que el hardware requerido es mínimo. Su principal desventaja es el uso de una mayor cantidad de bits para ofrecer una calidad similar, en comparación con los formatos más modernos.

- VP8: es un formato de compresión de video abierto y libre propiedad de Google y creado por On2 Technologies como sucesor de VP7. Solo es compatible con señales de video de escaneo progresivo con submuestreo cromático 4: 2: 0 y 8 bits por muestra, es un formato de codificación de transformación tradicional basado en bloques. Tiene mucho en común con H.264, por ejemplo, algunos modos de predicción [44].

2.7. Medidas de Calidad de una Llamada VoIP

Las llamadas VoIP como su acronimo lo dice significa voz a través de Internet, es una tecnología que nos proporciona comunicación de voz con elementos multimedia a través de una red IP.

Los dispositivos inteligentes como: routers, switches, etc, poseen herramientas que permiten verificar los datos, es decir comprueban que lo que entra por un puerto, sale por otro exactamente igual y en el menor tiempo posible [45].

Es de gran importancia medir la calidad de una llamada con la finalidad de garantizar que las conversaciones dispongan de la máxima calidad. Esa medición debe ser objetiva y comprobable, de ahí que existen diversos factores que permiten medir la calidad de una llamada VoIP, estos factores se describen a continuación:

- Latencia: Se define como el retraso existente desde el momento en que el sonido es enviado desde el teléfono del transmisor hasta el instante en el que llega al teléfono receptor. Por lo general en comunicaciones regulares VoIP la latencia es menor a 100ms y no mayor a 200ms [45].
- Jitter: conocido como la medida de la variabilidad en el tiempo de la latencia a través de una red, debido a que la información se divide en paquetes, cada paquete puede viajar por una ruta diferente desde el remitente hasta el receptor, provocando que lleguen a su destino en un orden diferente al que fueron enviados originalmente [45].
- Paquetes perdidos: en VoIP un paquete puede desaparecer por muchas razones entre las cuales se destacan: aumento en el tiempo de transmisión (cuando el TTL de cada paquete

es superado, el paquete se elimina por ser inútil), ruido en la señal digital que ocasiona variaciones que provocan que el paquete sea eliminado por haber sufrido cambios durante el viaje, etc [45].

- Ancho de banda: es una medida que nos sirve para conocer cuantos datos se pueden enviar por esa red.Este factor es de gran importancia, si se dispone de un ancho de banda reducido los paquetes demoraran en arribar a su destino, lo cual aumenta la latencia o produce la pérdida de paquetes [45].

Otro factor de gran importancia que permite medir la calidad de una llamada VoIP es el MOS. Descrito en la ITU-T P.800, MOS es la medida más conocida de calidad de voz. Es un método subjetivo de evaluación de calidad. Hay dos métodos de prueba: prueba de opinión de conversación y prueba de opinión de escucha.

Los sujetos de prueba de VoIP juzgan la calidad del sistema de transmisión de voz ya sea manteniendo una conversación o escuchando muestras de voz. Luego clasifican la calidad de la voz usando la siguiente escala: 5 - Excelente, 4 - Bueno, 3 - Feria, 2 - Pobre, 1 - Malo.

El MOS se calcula promediando las puntuaciones de los sujetos de prueba. Usando esta escala, se considera una puntuación promedio de 4 o superior como excelente y sin errores. Para medir cada uno de los parámetros antes descritos en una conversación VoIP se hace uso del Protocolo RTCP.

RTCP definido en el RFC 3550 [1] trabaja conjuntamente con RTP [2], este último se encarga de la entrega de los datos en tiempo real, en tanto que RTCP es usado para enviar paquetes de control a los participantes en una llamada. Su principal utilidad es proveer anotaciones acerca la calidad del servicio proporcionado por RTP.

2.8. Conclusiones

Las técnicas de reconocimiento facial han recibido mucha atención por parte de investigadores. Como se describió en este Capítulo se dispone de algoritmos de reconocimiento de rostros que tienen diferentes características. Entre los algoritmos más utilizados se destacan: Eigenfaces, Fisherfaces y LBPH, siendo este ultimo según la bibliografía el que mejor rendimiento nos ofrece frente a los cambios de luz.

Por otro lado, el protocolo SIP descrito en este capítulo permite iniciar, modificar y finalizar sesiones interactivas al usuario donde se da la intervención de elementos multimedia como la voz, video, mensajería instantánea, juegos on-line o realidad virtual. Estas características proporcionan un gran número de posibles aplicaciones en las que se puede hacer uso de SIP, ya

[1] RFC 3550: https://tools.ietf.org/html/rfc3550
[2] RTP : Real Time Transport Protocol define un formato de paquete estándar para el envío de audio y video sobre Internet

sean orientadas a la comunicación, seguridad, intercambio de menajes, control de acceso, entre otras.

La transmisión multimedia a través de redes inalámbricas permite el envío de video o voz con calidad ajustable. Es así como la calidad del contenido a recibir se puede modificar ajustando parámetros tales como: esquema de compresión, configuración de la codificación, método de entrega del contenido (video o voz), tasa de entrega de datos, entre otros.

De los diferentes codecs analizados, MJPEG es el que mejor rendimiento nos entregaría teniendo en cuenta el limitado rendimiento de hardware. Es así como se puede combinar SIP con este tipo de tecnologías de *videostreaming* con la finalidad de desarrollar sistemas de comunicación completos y eficientes (audio y video).

Capítulo 3

Estado del Arte

Este capítulo presenta un resumen de los principales trabajos realizados en el área de control de acceso al hogar y en el área de seguridad.

3.1. Introducción

El rostro humano tiene una forma particular que requiere cálculos complejos para reconocerlo. Podemos memorizar muchas caras durante nuestra vida y llegar a reconocerlas de inmediato, incluso después de años. El envejecimiento y las distracciones como los anteojos, la barba o el cambio de color de la piel pueden variar gradualmente las tasas de reconocimiento facial, por tal motivo la identificación de rostros representa uno de los tipos de biométrica más utilizados. El proceso de reconocimiento facial procede de la siguiente manera: se empieza con el cálculo y la sustracción de características específicas, luego se las verifica con la base de datos ya existente y se obtiene una correspondencia positiva entre las caras comparadas. Después de obtener los detalles de la forma de la cara, el sistema ajusta los datos y los guarda en un archivo al cual se le conoce como archivo de entrenamiento.

En los últimos años la automatización de las funciones del hogar ha tomado un gran impulso debido al ahorro energético, seguridad y *confort*. En este marco la identificación de características faciales ha tenido un gran auge gracias al avance de las tecnologías multimedia.

Dependiendo del algoritmo que se use para realizar el reconocimiento facial supondrá una carga muy elevada para el dispositivo, por ventaja en los últimos años el avance que se ha obtenido en el hardware a llevado a miniaturizar el mismo en dispositivos muy pequeños y potentes en términos computacionales.

3.2. Acceso al Hogar de Manera Inteligente

En la Tabla 3.1 se listan diversos proyectos relacionados con sistemas de acceso inteligente que utilizan reconocimiento facial. Se destacan dos sistemas comerciales de porteros inteligentes (SkyBell y Ring), los cuales carecen de la capacidad de reconocimiento de rostros.

Los trabajos presentados en la tabla usan un método para abrir la puerta automáticamente una vez detectado y contrastado el rostro con la base de datos, de la misma manera la mayoría no posee un botón como timbre, al contrario poseen un método que detecta el movimiento y realiza la detección si encuentra un rostro humano en las imágenes.

Como hardware preferido utilizan dispositivos basados en procesadores ARM los cuales ofrecen menos consumo de energía y son muy baratos comparados con lo precios de las computadoras de escritorio.

Para realizar el reconocimiento de rostros los autores utilizaron OpenCV la cual es una librería que se instala en el dispositivo, ninguno de los autores han ocupado servicios de reconocimiento facial de la nube como los que ofrece Amazon (AWS) o Google (Tensor Flow), esto se debe en gran medida a la latencia que puede existir al enviar las imágenes al servidor y a que estos servicios son pagados.

Tabla 3.1: Trabajos relacionados: acceso inteligente al hogar

	Tipo Adquisición	**Algoritmo Utilizado**	**Lenguaje Programación**	**Software Reconocimiento Usado**	**Tipo Notificación**	**SO Notificación**
Enhanced Smart Doorbell System Based On Face Recognition [46]	Cámara	Eigenface	Python, C, SQL	OpenCV	Registro de Entradas	Android
Design and Implementation of Smart Doorbell using IOT [47]	Cámara	No descrito	C#	OpenCV, Microsoft Project Oxford	Google Cloud Message, Fotos	Android, Microsoft UWP
Low Cost Smart Security Camera with Night Vision Capability Using Raspberry Pi and OpenCV [48]	Cámara modificada con infrarrojo	No descrito	Python	OpenCV	Foto con nombre usuario vía correo	Multiplataforma
IOT based Home Automation using Raspberry Pi with Doorbell Security [49]	Cámara	LBPH	No descrito	OpenCV	Ninguna	Ninguna
Automated Door Access Control System Using Face Recognition [50]	Cámara	LBPH	Python	OpenCV	Log vía correo electrónico	Ninguna
Ring [18]	Cámara con infrarrojo, audio dos direcciones	No Disponible	No Disponible	No Disponible	Notificación y video en tiempo real	Android, IOS, Windows Phone
SkyBell [51]	Cámara con infrarrojo, audio dos direcciones	No Disponible	No Disponible	No Disponible	Notificación y video en tiempo real	Android, IOS

3.3. Seguridad en Acceso al Hogar Inteligentes

Uno de los aspectos más importantes a considerar en los sistemas de acceso es la seguridad, ya que los atacantes en los sistemas de rostros tradicionales pueden conseguir una fotografía del rostro del usuario y entrar al hogar fácilmente.

De todos los sistemas analizados y presentados en la Tabla 3.1, ninguno toma en cuenta el aspecto seguridad, últimamente ha sido Apple quien ha llevado la bandera en seguridad biométrica desarrollando su sistema llamado ***FaceId***.

FaceId de Apple funciona con un sensor llamado ***TrueDepth***, el cual utiliza una combinación de emisor y sensor de infrarrojos para pintar 30.000 puntos de luz infrarroja sobre y alrededor de la cara. Eso permite crear una especie de huella digital en 3D de la cara que pueda ser usada para compararla posteriormente. Con este tipo de tecnología se llega a un nuevo nivel en cuanto a seguridad se refiere, eliminando el riesgo de posibles acceso a personas no autorizadas.

Otra empresa que ha desarrollado reconocimiento fácil es Microsoft con su propuesta llamada ***Windows Hello***, el cual permite a los usuarios de ordenadores o teléfonos inteligentes iniciar sesión en sus dispositivos con solo mirar al mismo [52].

Windows Hello no funciona con cámaras tradicionales, es un requisito poseer una cámara en 3D y sensores infrarrojos ya que al igual que el sistema desarrollado por Apple crea un modelo en 3D del rostro.

3.4. Conclusiones

Los trabajos realizados hasta el momento, demuestran que es posible la implementación de un sistema de reconocimiento facial, usando plataformas de código abierto como OpenCV, además se aprecia que es posible la implementación dentro de un dispositivo de hardware limitado como la Raspberry Pi 3.

Los trabajos que abordan el reconocimiento de rostros muestran que se puede obtener un buen rendimiento y sobretodo un alto nivel de precisión de cerca del 85 %. La mayoría de trabajos implementaron sus proyectos utilizando plataformas Linux y Python como lenguaje de programación, esto debido a su facilidad de programación y a la compatibilidad de OpenCV con el mismo, además se poseer librerías con las que se puede hacer uso de los pines GPIO.

En la mayoría de trabajos se aplica un pre-procesamiento a las imágenes tomadas, se alinean las caras, se aplican filtros y se convierte las imágenes a blanco y negro para que el procesamiento no sobrecargue el hardware utilizado.

En la Tabla 2.1 se aprecia la evaluación de los diversos algoritmos para el reconocimiento en los cuales se puede apreciar que el mejor resultado es provisto por LBPH debido a que no es tan

sensible a los cambios de iluminación, por esta cualidad se elegiría este como algoritmo para el reconocimiento de nuestro sistema.

Varios autores ven la necesidad de implementar un sistema de notificaciones hacia el dueño de la casa, la mayoría lo hace enviando una foto por e-mail o a través de aplicación para que reciba estas notificaciones y le permita abrir la puerta desde el dispositivo móvil.

Finalmente en el apartado de seguridad, ninguno de los autores tomaron medidas para implementar funciones que eviten que se pueda hackear el sistema con una foto del usuario, esto puede deberse a la limitada capacidad de procesamiento, pero es uno de los aspectos más importantes a considerar para así evitar la presencia de intrusos en el hogar.

Capítulo 4

Arquitectura y Especificación del proceso de diseño del sistema de acceso

En este capítulo se trata la metodología de manera técnica, mostrando las características de los dispositivos utilizados y la disposición física de los mismos, se describen las plataformas de software utilizadas en la programación de las distintas funciones de control tanto en el computador central como en el teléfono móvil, se describe la estructura física del sistema en general y de cada una de sus funciones de control.

4.1. Introducción

Uno de los ejes fundamentales en el área de la investigación científica lo constituye la replicabilidad, la misma brinda la posibilidad de obtener los mismos resultados de un estudio ya realizado siempre y cuando se replique de forma exacta al original. Para conseguir este objetivo, la metodología deber estar clara para el lector, es por esto que en este capítulo se tratan aspectos técnicos de los dispositivos utilizados, así como los parámetros utilizados para su configuración.

4.2. Especificación de Requerimientos del Prototipo

Para cumplir con los objetivos propuestos las funcionalidades que debe incorporar el prototipo de control de acceso inteligente son:

- Capturar imágenes en tiempo real.
- Procesar la información recolectada en tiempo real y controlar el sistema en general (imágenes).
- Controlar elementos de acceso al hogar (cerraduras).
- Comunicar al administrador del domicilio con el individuo que pretende acceder al hogar (llamada de voz y *videostreaming*).
- Proporcionar el acceso al hogar a personas previamente registradas.
- Registrar nuevos usuarios.

Estas funcionalidades se representan en el siguiente esquema operativo del sistema:

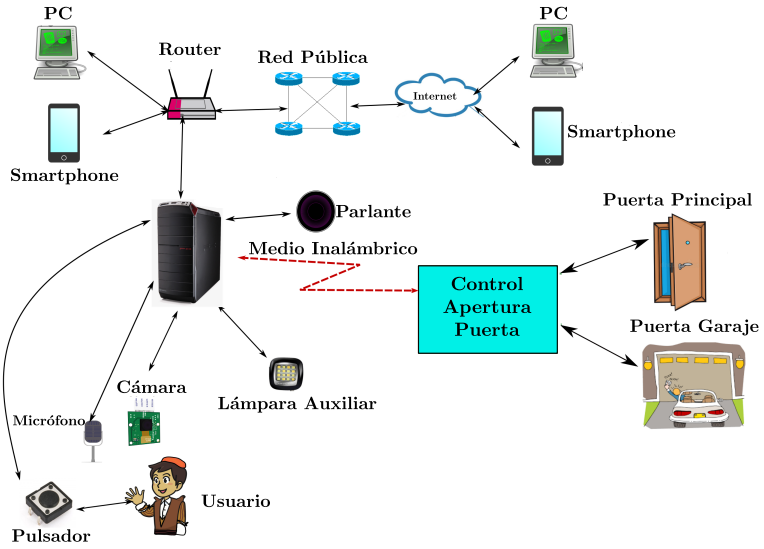


Figura 4.1: Esquema explicativo del sistema de acceso

Los requerimientos implícitos que deben satisfacer cada una de las funcionalidades del prototipo se describen a continuación:

- **Capturar imágenes en tiempo real.**
  Para satisfacer este requerimiento se utilizará una cámara estándar de alta definición, la cual garantice la calidad de la imagen y vídeo capturados.
- **Procesar la información recolectada en tiempo real y controlar el sistema en general.**
  Para ofrecer esta funcionalidad se hará uso de un miniordenador el cual sea capaz de proporcionar las funcionalidades de un ordenador común.
- **Controlar elementos de acceso al hogar.**
  En función del hardware a utilizar se implementará un sistema de control mediante el uso

de hilos y *sockets*, que permitan gestionar el acceso al hogar.

- **Comunicar al administrador del domicilio con el individuo que pretende acceder al hogar.**
  Para satisfacer este requerimiento se implementará una central telefónica IP y un sistema de *videostreaming* local, se utilizarán plataformas de código abierto para su implementación. El administrador del domicilio en función del vídeo y audio recibido estará en la capacidad de proporcionar o no acceso al hogar mediante el uso de un aplicativo *Android*, el cual recibirá la llamada y vídeo.
- **Proporcionar el acceso al hogar a personas previamente registradas.**
  Para satisfacer este requerimiento se propone implementar algoritmos de reconocimiento facial que permitan discriminar entre rostros de personas registradas y no registradas. Se manejará el control de acceso mediante la gestión de puertos y pines que permitan ejecutar las órdenes del procesador.
- **Registrar nuevos usuarios.**
  Mediante el uso de un aplicativo *Android* se gestionará el registro de nuevos usuarios, el cual mediante *sockets* enviará peticiones para habilitar el proceso de toma de imágenes (rostro del usuario) y posterior entrenamiento de las mismas.

4.3. Estudio y Elección del Hardware del Prototipo

Es necesario realizar la elección de hardware con las características técnicas antes descritas, la plataforma elegida debe disponer de entradas y salidas digitales que permitan enviar o recibir órdenes desde o hacia el computador. Una manera efectiva para efectuar la elección de hardware correcto es considerar la "cadena de valor tecnológica" del IoT, la cual recomienda considerar aspectos tales como [53]:

- Recolección de datos: La plataforma de hardware debe ser capaz de manejar cualquier tipo de sensor ya sea estos de temperatura, luminosidad, presión, movimiento, etc.
- Procesamiento de datos: La capacidad de procesar la gran cantidad de información capturada por los diferentes dispositivos es uno de los aspectos que debe cumplir el hardware considerado, en especial si se considera que se debe procesar audio y vídeo de manera simultánea.

Las plataformas de hardware consideradas se describen a continuación:

4.3.1. Raspberry Pi

Raspberry Pi es una computadora pequeña creada por la Raspberry Pi Foundation en Reino Unido. El principal componente de la Raspberry Pi en sus diferentes modelos consiste en un

chip ARM [1] modelo 28xx desarrollado por Broadcom corriendo desde los 700 MHz para el primer modelo llegando hasta el 1Ghz en el modelo 3B+.Para la parte de procesamiento de gráficos cuenta con un GPU VideoCore de Broadcom el cual le hace posible reproducir video HD [7].

Figura 4.2: Placa Raspberry Pi [7]

4.3.2. Banana Pi

La Banana Pi es un micro-ordenador muy parecido a Raspberry Pi, basa su funcionamiento en Allwinner A20 capaz de correr *Android*, Lubuntu o Raspbian. Algunas de sus características más relevantes son su CPU A20 ARM Cortex-A7 Dual-Core, GPU: ARM Mali400MP2Complies con OpenGL ES 2.0/1.1 y su Memoria SDRAM: 1GB DDR3 (compartida con GPU), lo cual lo convierte en una de los más potentes miniordenadores del mercado.

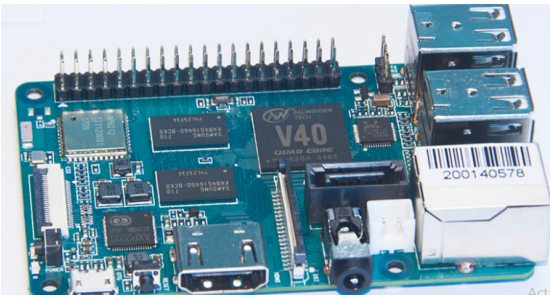


Figura 4.3: Placa Banana Pi [8]

[1] ARM consiste de una arquitectura RISC, Ordenador con Conjunto Reducido de Instrucciones de 32 bits

4.3.3. Orange Pi

Orange Pi es una micro-computadora conformada por una sola placa y diseñada por la compañía Shenzhen Xunlong Software Limited. Orange Pi es de código abierto y en su placa dispone de la capacidad de ejecutar sistemas operativos como: Ubuntu, Debian, *Android* y las imágenes de Rasberry Pi y Banana Pi. Hace uso de microprocesadores tipo A64 Quad-core Cortex-A53 64bit o ARM Cortex-A5 32bit y AllWinner H2, H3 ó H5, además dispone de versiones con capacidad en RAM que van desde los 256 MB hasta los 2 GB DDR3 SDRAM.

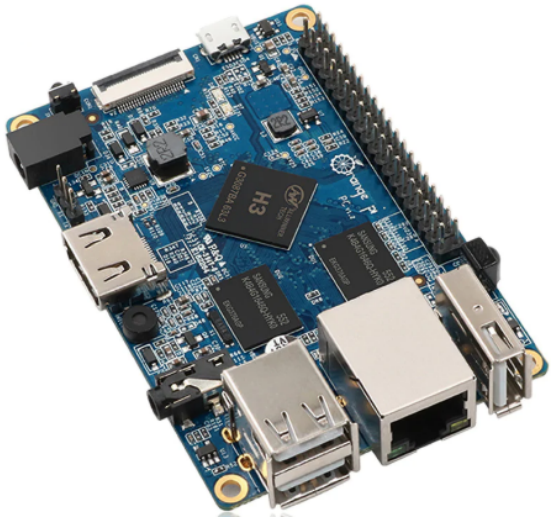


Figura 4.4: Placa Orange Pi [9]

4.3.4. Módulo WiFi ESP8266

El Módulo WiFi ESP8266 es un dispositivo que incorpora una pila de protocolos TCP/IP que le permite a cualquier microcontrolador acceder a su red WiFi. El ESP8266 es capaz de alojar una aplicación o descargar todas las funciones de red Wi-Fi de otro procesador de aplicaciones. Este módulo tiene una capacidad suficiente de procesamiento y almacenamiento a bordo que le permite integrarse con los sensores y otros dispositivos específicos de la aplicación a través de sus GPIO con un desarrollo mínimo inicial y carga mínima durante el tiempo de ejecución.

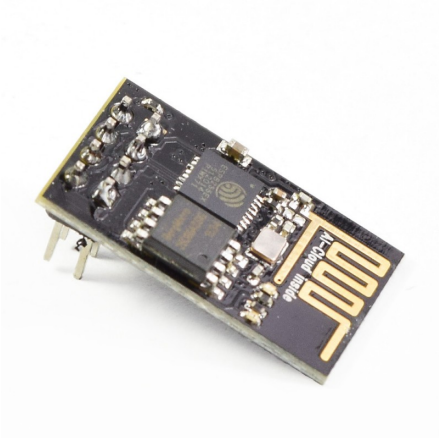


Figura 4.5: Módulo WiFi ESP8266 [10]

En función de las especificaciones de requerimientos del prototipo a diseñar se optó por utilizar como elemento de Hardware la Raspberry Pi 3 modelo B , la cual satisface las demandas planteadas en lo referente a captura y procesamiento de datos de forma fiable y rápida. Raspberry Pi posee soporte para una gran cantidad de sistemas operativos (Aros, *Android*, Raspbian, Ubuntu Mate, Lubuntu, Fedora, etc), otro aspecto de gran importancia es el costo en comparación a otras plataformas que satisfacen los requerimientos planteados esta es la más económica. En la sección 4.4 se detalla la estructura y características de la Raspberry Pi 3.
De igual manera se optó por hacer uso del módulo WiFi ESP-82266 para el manejo de actuadores (puerta y garaje) dado su versatilidad en conexiones inalámbricas con disintas plataformas de servcios y hardware.

4.4. Descripción de Hardware: Raspberry Pi 3 Modelo B

La Raspberry Pi en términos generales consiste en una computadora del tamaño de una tarjeta de crédito diseñada y fabricada por Raspberry Pi Foundation, una organización sin fines de lucro.

La ultima versión es la Raspberry Pi 3 Modelo B es un dispositivo muy versátil que empaqueta una gran cantidad de hardware en una tarjeta pequeña y económica. Es perfecta para la electrónica en sus diversos campos, útil para programar lecciones y estimular la enseñanza de ciencias de la computación.[54]. La Tabla 4.1 presenta las características técnicas de la Raspberry Pi B.

- **Pines GPIO:** En el borde del dispositivo nos encontramos los pines GPIO, que se pueden usar para interactuar con otras piezas de hardware. Son accedidos y programados vía software, estos pines se pueden definir como como entradas o salidas digitales. No todos

Tabla 4.1: Características Raspberry Pi 3

	Raspberry Pi Modelo 3
Precio	$ 35
Tipo CPU	ARM Cortex A53
Nucleo CPU	4
Bus CPU	64-bit
Velocidad CPU	1.2GHz
Tipo GPU	VideoCore IV
Nucleos GPU	1
Velocidad GPU	400MHz
Memoria	1GB LPDDR2
Flash	microSD(hasta 256GB)
Puertos USB	4
Red	10/100Mbps
WiFi/Bluethoot	802.11b/g/n, Bluethoot 4.1
HDMI	HDMI 1.4
Fecha Lanzamiento	Febrero 2016

los pines son configurables existen pines de voltaje 5v y 3.3v, así como pines de GND. Para cualquier referencia se puede abrir un terminal en el Raspberry Pi y corriendo el comando ***pinout*** se obtendrá una gráfica con los pines y su función [11].

- **Audio: Raspberry Pi 3 Modelo B:** La Raspberry PI 3 dispone de un conector de audio de 3.5mm que se puede utilizar como salida de audio, pero no tiene entrada de micrófono y no integra micrófono algo que en la actualidad le resta potencial en cuanto a soporte multimedia. Una posible solución a este problema es utilizar una tarjeta de sonido con adaptador USB como la que se muestra en la Figura 4.6.

Figura 4.6: Tarjeta de audio USB[Fuente: Autores]

- **USB: Raspberry Pi 3 Modelo B :** Cuenta con 4 puertos USB 2.0 que permiten conectar teclados, mouse, cámaras etc, que proporcionan a la Raspberry una funcionalidad adicional.
  Existen algunas diferencias entre el hardware USB en el Raspberry Pi y el hardware USB en computadoras de escritorio o dispositivos portátiles / tabletas. El puerto host USB dentro del Pi es un host OTG ya que el procesador de aplicaciones que alimenta el Pi, BCM2835, fue diseñado originalmente para ser utilizado en el mercado móvil, es decir,

como el puerto USB único en un teléfono para conexión a una PC, o a un solo dispositivo. En esencia, el hardware OTG es más simple que el hardware equivalente en una PC [55].

- **Bus SCI :** El bus CSI es un bus definido por la alianza MIPI, con este bus se puede utilizar una cámara que haga uso de esta interfaz como es el caso de la ***Raspberry Camera***. Este bus se encuentra entre la salida de audio y el conector HDMI de la Raspberry.
- **Ethernet Y WiFi :** El Raspberry Pi 3 integra un puerto Ethernet que permite conectarnos a una red con una conexión máxima de 100 Mbps, además cuenta con un módulo WiFi para conectarse a redes de manera inalámbrica.
  En la Figura 4.7 se aprecia los elementos que forman la Raspberry Pi:

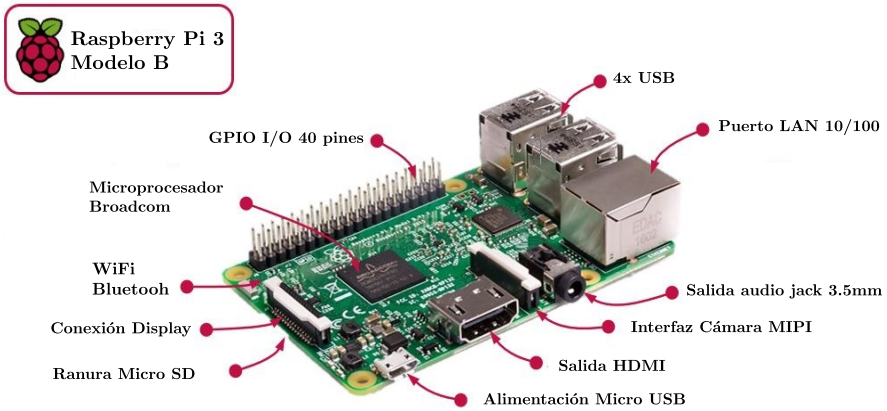


Figura 4.7: Ubicación elementos en la Raspberry Pi [11]

4.5. Plataformas de Servicios

Como se ha descrito en capítulos anteriores el “Sistema de control de acceso inteligente para hogares” debe brindar la funcionalidad de abrir la puerta del garaje del hogar a través de internet, para cumplir este objetivo es necesario hacer uso de plataformas que permitan conectar el dispositivo móvil a la red, en esta sección se describen las plataformas más conocidas hasta el desarrollo de este proyecto. Posteriormente en función de las necesidades se escoge una plataforma sobre cual se trabajará en el desarrollo del "Sistema de Acceso Inteligente", dicha plataforma se describe de forma detallada en la sección 4.6.

- **ThingSpeak:** ThingSpeak es una plataforma que ofrece diversos servicios destinados exclusivamente a la construcción de aplicaciones de IoT. Ofrece la capacidad de recopilación de datos en tiempo real, la visualización de los datos recopilados en forma de gráficos, la capacidad de crear complementos y aplicaciones para colaborar con servicios web, redes sociales y otras API [56].

- **Spacebrew**: Spacebrew es un conjunto de herramientas de software abierto, dinámicamente re-enrutable para conectar elementos interactivos entre sí. Cada elemento que se conecta al sistema se identifica como un suscriptor (leyendo datos) o un editor (eliminando datos). Los datos están en uno de tres formatos estandarizados: un booleano (verdadero / falso), un rango numérico (0-1023) o una cadena (texto); también se puede enviar como un formato personalizado que especifique. Basa su funcionamiento en un modelo cliente/-servidor y hace uso de WebSockets para realizar la comunicación entre los dos dispositivos [57].
- **Firebase:** Firebase es un servicio al que podemos llamarlo una base de datos en tiempo real, permite el desarrollo de aplicaciones web y aplicaciones móviles. Entre las características de Firebase destacan: proveer funciones estadísticas, bases de datos, informes de fallos y mensajería, almacena y sincroniza datos de app en milisegundos, permite autentificar usuarios y gestionar la seguridad del sistema [58].

4.6. Descripción de la Plataforma de Servicios: Firebase

Como se describió en la sub-sección 4.5 Firebase es una base de datos en tiempo real que permite el desarrollo aplicaciones móviles. A continuación, se describen sus funcionalidades más relevantes:

- **Realtime Database:** Esta función permite almacenar y sincronizar datos JSON en tiempo real. Las ventajas de sincronización en tiempo real brindan a los usuarios la capacidad de acceder a los datos desde cualquier dispositivo ya sea este web o móvil, y optimiza el trabajo en conjunto de los mismos. Si un usuario se desconecta del servicio el SDK de Realtime Database hace uso del caché local del dispositivo para efectuar la publicación y almacenamiento de los cambios efectuados, de esta manera cuando el dispositivo se conecta, los datos locales se sincronizan automáticamente [58].
- **Crashlytics:** Firebase Crashlytics brinda la funcionalidad para efectuar un seguimiento de fallas en lo referente a la estabilidad y calidad que afectan la aplicación, priorizarla y corregirla en tiempo real [58].
- **Cloud Storage:** Cloud Storage permite almacenar fotos y vídeos de los usuarios. La funcionalidad de almacenamiento ofrece rapidez en el procesado y facilidad en el manejo de contenido como fotos y vídeos [58].
- **Hosting:** Esta funcionalidad permite enviar contenido web de forma rápida y eficaz, brindando la capacidad de montar una aplicación web de una página, una página de destino para aplicaciones de dispositivos móviles o una aplicación web progresiva de forma rápida y sencilla [58].

En el desarrollo del Sistema de Acceso Inteligente planteado se hará uso de *Firebase Realtime Database* para controlar el estado de la variable de apertura y cierre de la puerta del garaje del

domicilio. La configuración de los parámetros de *Firebase Realtime Database* para efectuar el control de la apertura y cierre de la puerta del garaje se describen en el Anexo E.

4.7. Plataforma para Servicios de Comunicación Integral: Asterisk

Asterisk es un *framework* de código abierto para la construcción de aplicaciones de comunicación. Asterisk convierte un computador ordinario en un servidor de comunicaciones para sistemas IP PBX, puertas de enlace VoIP, servidores de conferencias y otras soluciones personalizadas [59].

La arquitectura Asterisk está constituida por:

- **Canales (conexiones de telefonía a la PBX):** Se encargan de controlar diversos tipos de conexiones (protocolos VoIP como SIP, IAX, MGCP y H.323). Los canales se registran para conexiones de salida a otro servidor VoIP con SIP a la red Free World Dialup u otros proveedores SIP.
- **Protocolos soportados por Asterisk:** Admite muchos protocolos para voz sobre IP, entre ellos se hallan protocolos de señalización como H.323 o SIP y protocolos de transporte de medios como RTP. Los flujos de medios, la voz en la red, se pueden codificar utilizando diversos códecs.
- **Plan de marcado Asterisk:** El plan de marcado se almacena en un archivo de texto, el archivo de configuración extensions.conf, en este archivo las acciones están conectadas a extensiones.
- **Núcleo:** El núcleo PBX actúa como el sistema de conmutación central para llamadas telefónicas dentro de una empresa. Los sistemas *IP* PBX manejan el tráfico interno entre las estaciones y actúan como el guardián de acceso al mundo exterior, es el componente esencial que provee gran parte de la infraestructura. Se encarga de la lectura de los archivos de configuración, la carga de módulos y distintos componentes que proporcionan otras funcionalidades de llamada [60].
- **Interfaz del Administrador** Asterisk se ejecuta en un sistema Linux, Unix FreeBSD [2] o OpenBSD [3]. La mayoría de las funcionalidades se basan en Linux. Como administrador, puede conectarse a una PBX de Asterisk en ejecución con una interfaz de línea de comando o una de varias interfaces gráficas [60].

La configuración de Asterisk se lleva a cabo en archivos de texto, los archivos de configuración de Asterisk se hallan en el directorio: /etc/asterisk. En el Apéndice C se detalla la configuración

[2] FreeBSD consiste en un sistema operativo de uso libre para ordenadores con CPU que hacen uso de arquitectura x86, Intel 80386, Intel 80486 (versiones SX y DX), y Pentium

[3] OpenBSD consiste en un sistema operativo de uso libre Unix multiplataforma, descendiente de 4.4BSD. Se enfoca en la seguridad y la criptografía

Asterisk empleada para efectuar llamadas entre terminales (Raspberry Pi y teléfono móvil)

4.8. Descripción del Proyecto CSipSimple para *Android* Studio

CSipSimple [4] consiste en una aplicación VoIP (Voz sobre Protocolo de Internet) para el sistema operativo *Android* que hace uso del Protocolo SIP 2.4.1. Es un software libre y de código abierto publicado bajo la Licencia Pública General de GNU.[61] [5]

Las principales características CSipSimple son:

- Soporte multi-codec: Speex (banda estrecha / banda ancha), G.711 (u-law/a-law), GSM, iLBC, G.729, G.722, AMR (banda estrecha), iSAC, SILK (banda estrecha/banda ancha/-banda ultra ancha).
- Agrega soporte para Códec, G.726, G.722.1 y Opus.
- Soporte para múltiples cuentas: se pueden activar hasta 10 cuentas al mismo tiempo.
- Hace uso de un controlador de audio nativo.
- Integración con el sistema operativo *Android* con filtros y reglas de reescritura.
- Seguridad y cifrado con SRTP, SIP sobre TLS 1.0 y ZRTP.
- SIP SIMPLE mensajería.
- Ocultamiento de pérdida de paquetes (PLC) usando PJSIP.
- Soporte para IPv6 : si el hardware, la versión de *Android*, el ISP y todas las demás partes de las conexiones involucradas pueden manejar IPv6, entonces CSipSimple se puede usar para hacer llamadas directas de extremo a extremo de IPv6 a IPv6.

A esta aplicación se le hará modificaciones para añadir nuevas características tales como: la capacidad de incluir vídeo unidireccional en las llamadas, se habilitará *sockets* de comunicación entre la aplicación y la Raspberry Pi para el manejo de las variables de apertura y cierre de puertas, se dotará a la aplicación de una función para identificar y registrar al usuario en el servidor Asterisk y de vídeo de forma automática, se añadirá la opción de registro de nuevos usuarios en lo referente al reconcomiendo de rostros para el control de acceso.

En el Apéndice F se detalla el proceso de compilación desde el repositorio a la IDE Android Studio, de igual manera se detallan las mejoras y modificaciones realizadas al proyecto CSip-Simple.

[4] CSipSimple: `https://github.com/tqcenglish/CSipSimple`
[5] GNU consiste en una licencia de software libre, que provee a desarrolladores de software y usuarios finales la libertad de ejecutar, estudiar, compartir y modificar software.

4.9. Descripción del Framework Multimedia GStreamer

GStreamer es un *framework* multimedia de código libre multiplataforma escrito en el lenguaje de programación C. GStreamer permite crear aplicaciones audiovisuales que facilitan el manejo de audio, vídeo o cualquier flujo de datos multimedia de una manera modular. Su característica más importante es la reducción del procesado de aplicaciones mediante su diseño de pipelines [12].

La idea básica de GStreamer es unificar varios elementos en pipelines, los cuales definen como se realizará el flujo de datos. GStreamer consta de una variedad de formatos disponibles además posee gran flexibilidad ya que se pueden instalar plugins (brinda nuevas funcionalidades a la aplicación) y códecs que facilitan la creación de la aplicación. En la Figura 4.8 se aprecia la arquitectura donde el núcleo del *framework* provee la infraestructura para el sistema (pipeline, manejo multimedia, clases bases, tipos de archivos soportados, plugins etc) [12]. GStreamer

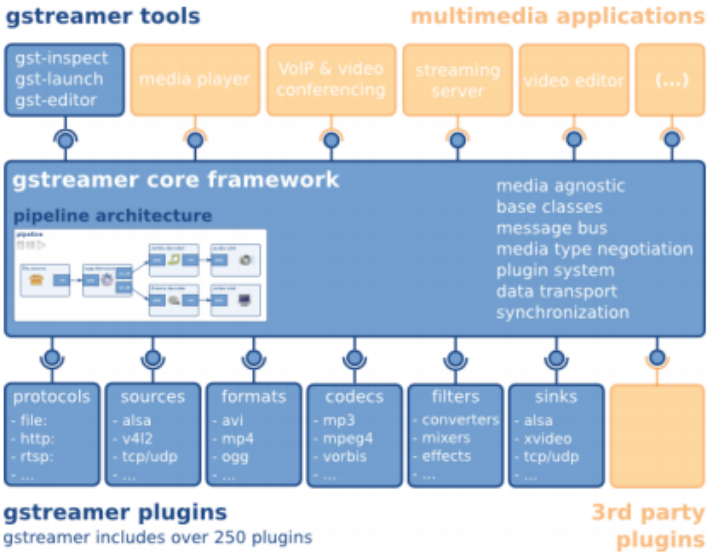


Figura 4.8: Visión general arquitectura GStreamer [12]

posee enlazadores con otros lenguajes de programación como: Java, Qt, o Python, garantizando la interoperabilidad entre los diversos lenguajes.

Una tubería en GStreamer es un conjunto de elementos que se conectan entre sí para obtener un resultado deseado (transmitir flujo de datos). Los elementos de una tubería realizan varias tareas como obtener una fuente de vídeo desde un archivo o desde la cámara, decodificar o codificar en un formato específico, sacar el flujo de datos por la tarjeta de sonido o simplemente crear un servidor TCP (tcpserversink) para compartir los datos a través de la red [12].

4.10. Casos de Uso y Diagrama de Flujo del Algoritmo de Control Principal

Para satisfacer los requerimientos del sistema y proveer un esquema de fácil comprensión que permita visualizar el prototipo, la Figura 4.9 presenta el diagrama de caso de uso general del Sistema de Acceso Inteligente. En función de los parámetros antes descritos y analizados se

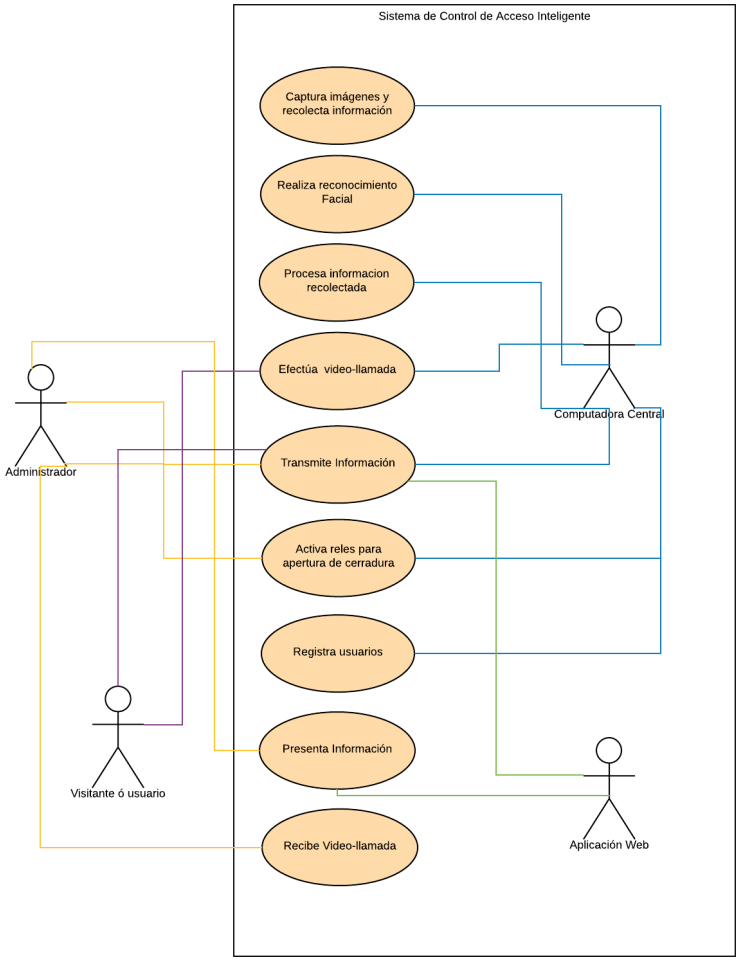


Figura 4.9: Diagrama: Caso de uso del sistema de acceso en general

tendrá un sistema con dos actores externos: Administrador del domicilio y usuario o invitado y dos actores internos: computadora central y la aplicación web (servidor web, servidor de aplicaciones y base de datos).

La Figura 4.10 describe mediante el flujograma general del Sistema de Acceso el conjunto

de pasos que se efectúan para gestionar el acceso al domicilio. Se distinguen tres ejecuciones principales: Control de Acceso puerta principal, Control de Acceso Puerta de Garaje y Control Remoto de Puerta Principal y *videostreaming*.

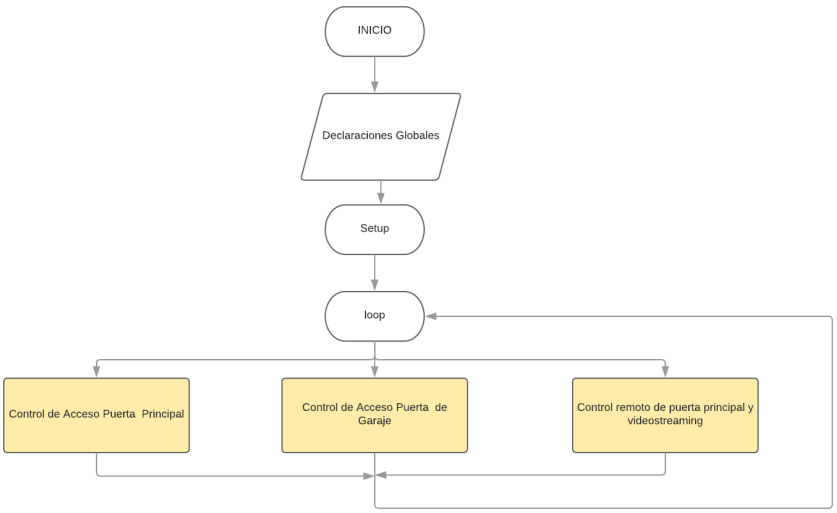


Figura 4.10: Flujograma del prototipo general.

1. **Control de Acceso Puerta de Garaje**
   El diagrama de bloques correspondiente se presenta en la Figura 4.11. Esta función hace uso de *Firebase Realtime Database*, la cual proporciona una base de datos alojada en la nube. El administrador dispondrá de un aplicativo móvil con la opción de abrir y consultar el estado de la puerta del garaje. Los datos se guardan en formato JSON y se sincronizan en tiempo real con cada cliente conectado a Firebase.

2. **Control Remoto de puerta Principal y *videostreaming***
   El diagrama de flujo se presenta en la Figura 4.12. Mediante esta función es posible acceder en tiempo real a la cámara de vídeo de la Raspberry Pi del portero de Acceso, con lo cual el usuario dispone de un sistema de vigilancia a través de su dispositivo móvil. De igual manera el usuario está en la capacidad de abrir la puerta principal mediante el aplicativo móvil, lo cual brinda comodidad en el acceso al hogar.

3. **Control de Acceso puerta principal**
   Como se muestra en la Figura 4.13 el bloque de ejecución lleva a cabo la siguiente secuencia: Cuando un usuario desea acceder al domicilio presionará el botón de ingreso (pulsante), el cual activará el hilo de proceso que permite ejecutar el algoritmo de reconocimiento facial mediante la toma de diversas fotografías. Si el rostro del usuario se

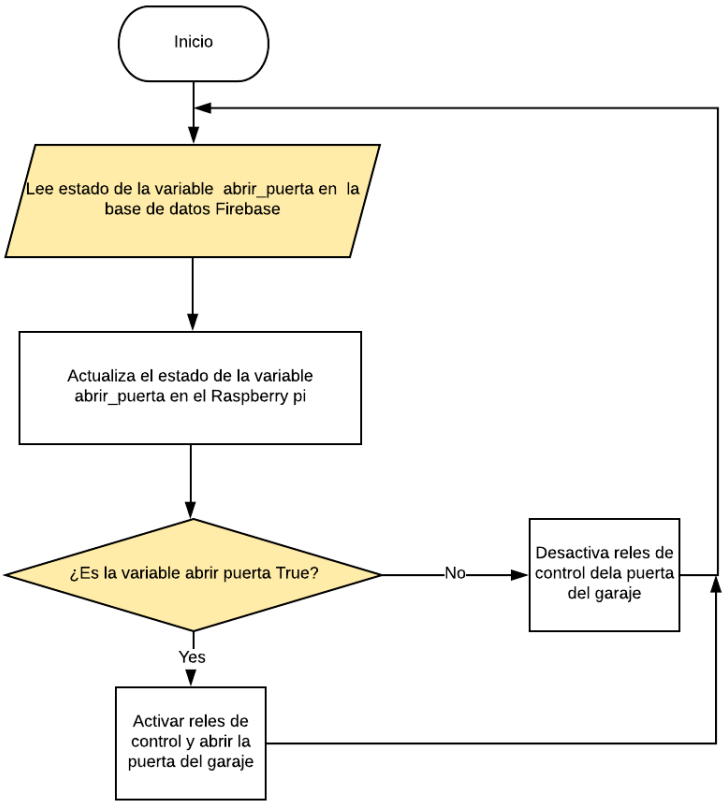


Figura 4.11: Flujograma de acceso puerta garaje

halla registrado y entrenado correctamente por el algoritmo de reconocimiento se procede a abrir la puerta principal, caso contrario se activará una llamada de voz y vídeo a través de la centralita telefónica Asterisk y GStreamer respectivamente. La vídeollamada será recibida en el teléfono móvil del administrador del domicilio el cual sin tener la necesidad de contestar ya sabrá por anticipado la identidad de la persona que desea ingresar al domicilio, con esta información el administrador decidirá rechazar la llamada y denegar el acceso o contestar la llamada para llevar a cabo un diálogo con el usuario y determinar si se habilita o no el acceso al hogar.

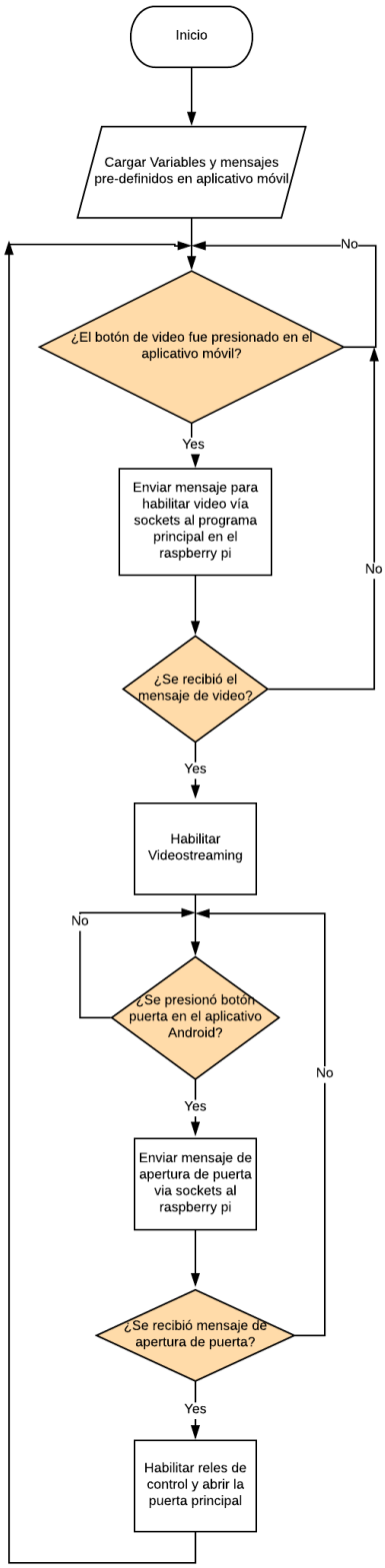


Figura 4.12: Flujograma de control puerta principal y *videostreaming*

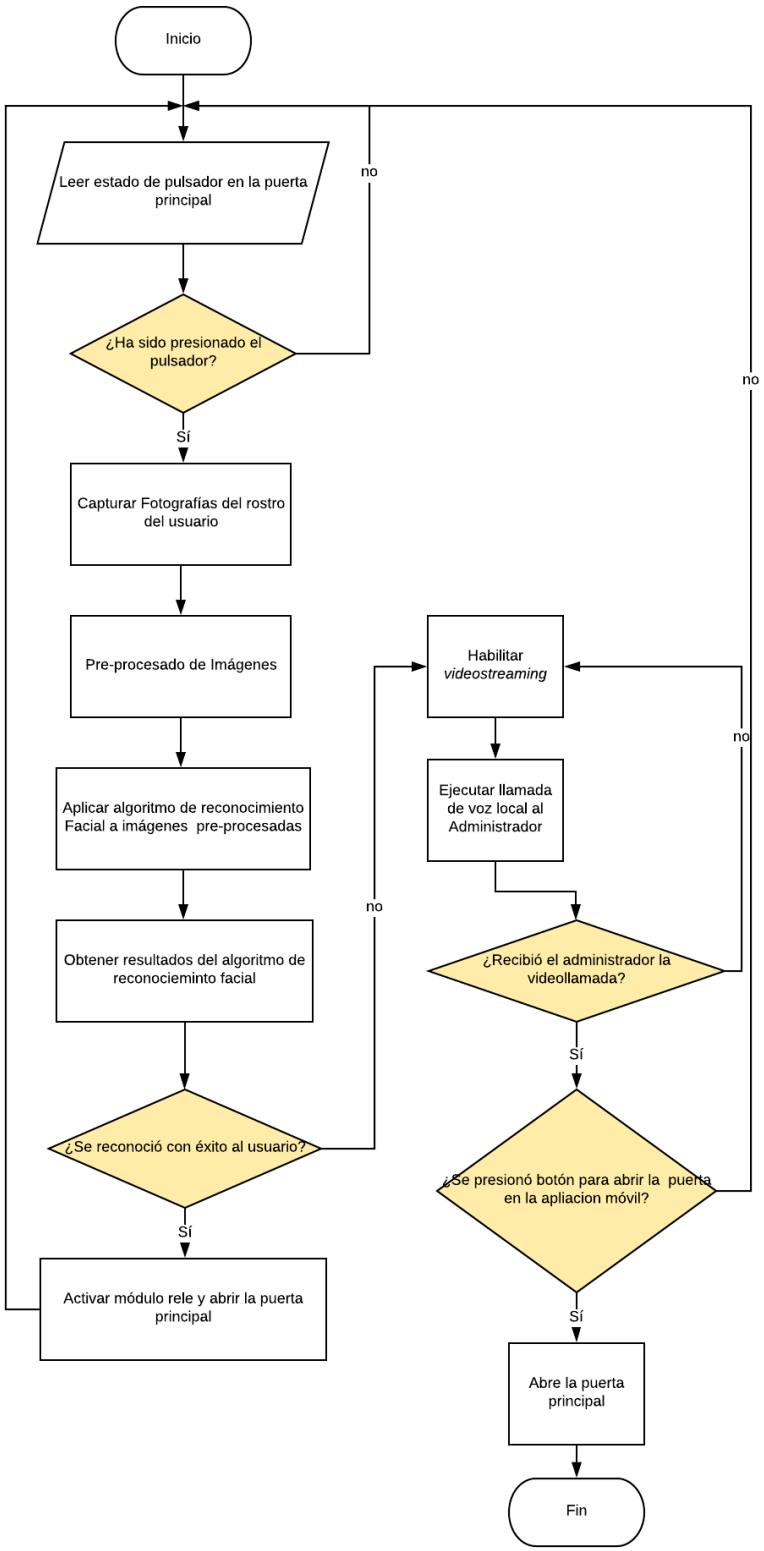


Figura 4.13: Flujograma de acceso puerta principal.

4.11. Arquitectura del Sistema de Acceso Inteligente

4.11.1. Topología de Red

Dada las necesidades, costo y condiciones de diseño se optó por una red centralizada con topología en estrella. La topología se describe en la Figura 4.14.

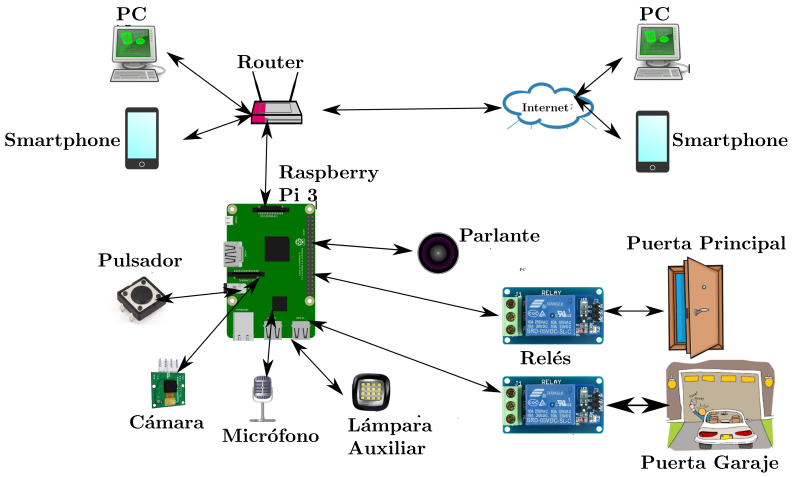


Figura 4.14: Topología de red del sistema de acceso inteligente.

El diseño propuesto consta de tres niveles: en el nivel más bajo se encuentran los actuadores y sensores (cámara de vídeo), en el nivel medio se halla el sistema de procesamiento de datos y a nivel superior se encuentra la conexión a internet. Se utiliza la Raspberry Pi 3 para manejar los datos recolectados por las entradas analógicas y digitales, este elemento se encuentra en un nivel medio de la topología, su función más importante es la de servir como puente entre el nivel más bajo y el nivel superior (nivel de acceso a internet). Se hace uso de la conexión inalámbrica WiFi 802.11 b/g/n para comunicarse con el enrutador, el cual permite conectarse a internet.

4.11.2. Topología de Control

En la Figura 4.15 se presenta la conexión física de cada uno de los actuadores y dispositivos de toma de datos a los pines del procesador de acuerdo a su tipo entrada/salida. La cámara para la captura de imágenes se conecta al puerto CSI (cámara Raspberry Pi), en tanto que para la conexión del micrófono y parlante se hace uso de un adaptador de audio USB 2.0.

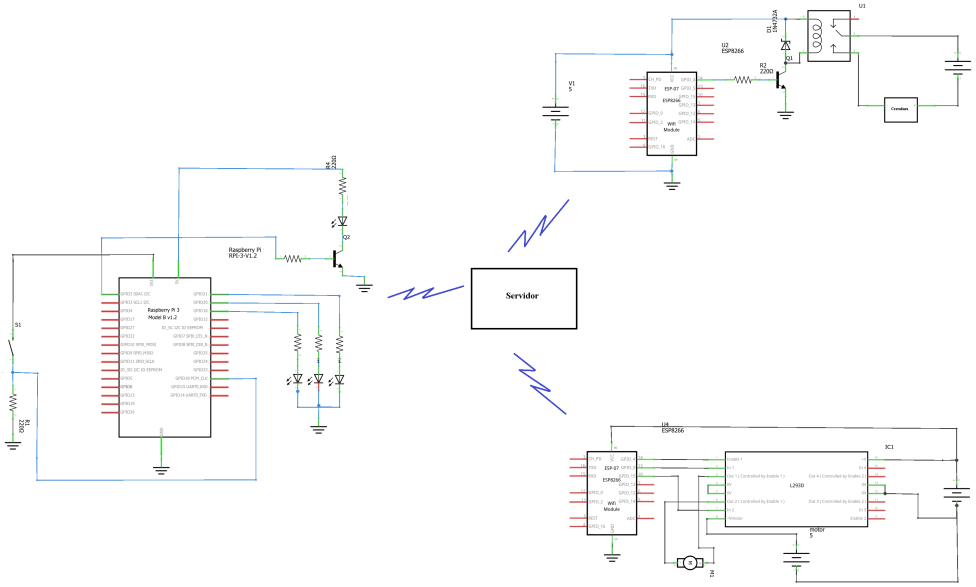


Figura 4.15: Conexión física de los dispositivos del sistema

4.12. Entorno de Programación y Software Utilizado

En esta sección se describen cada una de las funciones que intervienen en el código de control principal. Todas ellas se ejecutan en paralelo mediante el uso de hilos, los hilos permiten a las aplicaciones llevar a cabo diferentes operaciones de forma concurrente en un mismo espacio de proceso. Las funciones de control y proceso se detallan a continuación:

Función de Reconocimiento de Rostros: OpenCV y Python
Para ejecutar esta función se hace uso de la librería OpenCV. Esta función se encarga de capturar las imágenes del usuario que pretende ingresar al domicilio y las reduce al tamaño estándar de 250×250 pixeles. Con las imágenes en tamaño estándar se aplica la etapa de pre-procesamiento que se encarga de escalar el rango de la imagen de [0, 255] a [0, 1.0] usando una función gamma. La imagen corregida de salida se obtiene a partir de la ecuación 4.1:

$$O = I^{\frac{1}{G}} \tag{4.1}$$

Donde I es la imagen de entrada y G es el valor gamma. Los valores gamma<1 desplazarán la imagen hacia el extremo más oscuro del espectro, mientras que los valores gamma >1 harán que la imagen parezca más clara.
Posterior al pre-procesamiento gamma se efectúa una ecualización de histogramas para mejorar

el contraste las imágenes en función de la cantidad de luminosidad disponible, como se muestra en el listado 4.1.

```
clahe = cv2.createCLAHE (clipLimit = 2.0, tileGridSize = (8,8))
cl1 = clahe.apply (img)
```

Listado 4.1: Función de ajuste de histograma.

Posterior al pre-procesamiento de imágenes se procede a cargar el archivo de entrenamiento que guarda las características faciales de los usuarios registrados en el sistema de ingreso, la información de este archivo se coteja con cada una de las imágenes pre-procesadas con la finalidad de obtener una estimación de semejanza entre rostros. El diagrama de flujo de esta función se detallan en el Apéndice D.

Función de control: Activación de *videostreaming*, control de apertura y cierre de puertas

Esta función es la encargada de manejar el socket de *videostreaming*, control de apertura y cierre de puertas. Se encarga de recibir mensajes por parte del aplicativo móvil y en función del mensaje recibido efectúa diferentes tareas, tales como: apertura de la puerta principal, apertura de la puerta del garaje y habilitación/cierre de *videostreaming*. El funcionamiento del módulo socket se resume en la Figura 4.16. El modo de operación de la función de control de puertas y

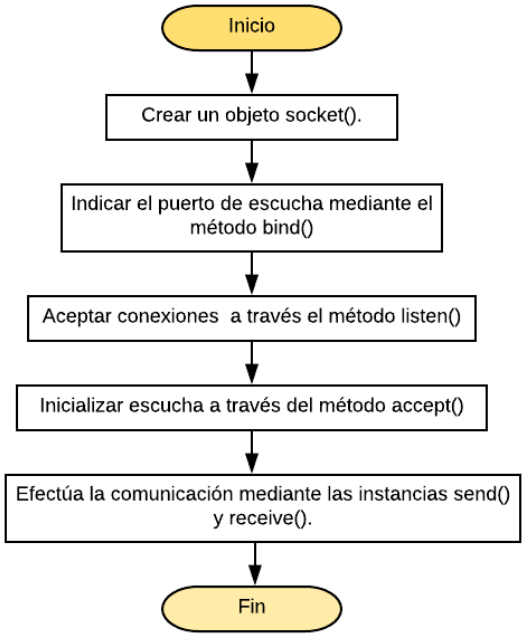


Figura 4.16: Funcionamiento del módulo socket.

videostreaming se detalla en el Apéndice D

Función de control: Registro y entrenamiento de usuarios en el Sistema de Acceso
Esta función permite registrar usuarios nuevos en el sistema de acceso al hogar mediante la aplicación móvil, se utilizan *sockets* para enviar la solicitud de creación al programa de control principal en la Raspberry Pi 3. Cuando la solicitud de creación de usuario es recibida y aceptada se procede a invocar el script creabase.py, el cual se encarga de capturar las imágenes del rostro del usuario a registrar, las pre-procesa (ajuste gamma y ecualización de histogramas), reduce las imágenes al tamaño estándar 250×250 píxeles y las almacena en el directorio de la base de datos del sistema de acceso.

Con las imágenes de los rostros a registrar en el directorio de la base de datos del sistema de acceso se procede a generar el archivo de carga de entrenamiento "trainer.yml", el cual permite registrar las características distintivas de cada usuario por nombre de registro. En el listado 4.2 se presentan las funciones de escritura y entrenamiento utilizadas.

```
1 recognizer.train(images, lables)
2 recognizer.write('trainer.yml')
```

Listado 4.2: trainer

El código completo de la implementación de registro de usuarios se muestra en el Apéndice D.

Función de control: Seguridad en el acceso por Reconocimiento Facial
Esta función , es la encargada de evitar el acceso a personas no autorizadas y detectar posibles falsos positivos que pretendan aprovecharse de las vulnerabilidades del sistema de captura de imágenes y posterior reconocimiento de rostros. Esta función pretende contrarrestar uno de los más grandes problemas de los sistemas de seguridad basados en reconocimiento de rostros como lo es el acceso no autorizado a través del uso de fotos o vídeos de un usuario previamente registrado.
Este problema lo han sufrido grandes compañías como: Microsoft Corporation [1] y Apple [2], cuyos sistemas de reconocimiento facial en computadoras y móviles fueron burlados por una simple foto a color [62] y una máscara impresa en 3D [63] con la apariencia del individuo registrado al sistema.
Considerando que el sistema de acceso utiliza una cámara de fotos convencional (Raspicam) y no una cámara 3D que permita capturar imágenes en profundidad de los usuarios, se propone hacer uso de un sistema de seguridad basado en el movimiento de facciones del rostro en instantes de tiempo predefinidos de los cuales el usuario registrado es consiente a la hora de acceder al sistema. El sistema descrito posee las siguientes ventajas:

- Obliga al usuario a realizar movimientos fáciles pre-definidos (movimiento de boca, ojos o cejas) durante la etapa de captura de imágenes, con lo cual se evita que el sistema pueda

[1] https://www.microsoft.com/es-ec/
[2] https://www.apple.com/la/

se engañado por fotos.

- Se agrega un nivel de seguridad extra puesto el movimiento facial que se deba realizar solo será conocido por las personas registradas en el sistema. Existe un tiempo límite para realizar el movimiento facial de acceso, fuera del cual el sistema se reinicia evitando el acceso del usuario.
- Se puede configurar desde uno hasta diez niveles de seguridad extra, considerando que por cada nivel de seguridad en el que se pida movimiento facial se aplica un reconocimiento rápido de rostros para filtrar de forma temprana posibles intrusos.

El sistema de seguridad descrito se basa en la propuesta de Vahid Kazemi and Josephine Sullivan [64], los cuales proponen técnicas efectivas en la detección de marcas faciales.

Para efectuar la implementación y detección de marcas faciales se hace uso de un conjunto de entrenamiento de puntos de referencia faciales etiquetados en una imagen. Estas imágenes se etiquetan manualmente, especificando puntos de coordenadas (x, y) de las regiones que rodean a cada estructura facial. Considera la probabilidad de distancia entre pares de píxeles de entrada. Teniendo en cuenta estos datos de entrenamiento, un conjunto de árboles de regresión está entrenado para estimar las posiciones del hito facial directamente a partir las intensidades de los píxeles (es decir, no se lleva a cabo la extracción de características).

Para detectar el movimiento de ojos, rostro o cejas en base al trabajo de Soukupová y Čech [13] en su artículo de 2016, Real-Time Eye Blink Detection using Facial Landmarks, los autores derivan una ecuación que refleja la relación del movimiento ocular, denominada relación de aspecto del ojo (EAR):

$$EAR = \frac{\|p2 - p6\| \, \|p3 - p5\|}{2 \, \|p1 - p4\|} \tag{4.2}$$

donde los puntos p1, p2, p3, p4, p5 y p6 son coordenadas del ojo humano dentro la imagen: Esta fórmula se puede generalizar para la detección del movimiento en los labios de la boca y

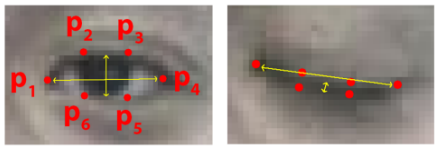


Figura 4.17: Coordenadas de los ojos dentro una imagen. [13]

cejas. La implementación de la función de seguridad se presenta en el Apéndice D.

4.13. Conclusiones

Como se describió en este capítulo se dispone de cuatro hilos de proceso los cuales se ejecutan en paralelo. Dentro de cada hilo se ejecutan una o más funciones de control las cuales fueron descritas en los diagramas de flujo del sistema.

Se considera utilizar Firebase para el control de la apertura y cierre de la puerta del garaje en tiempo real, como se describió en la sección 4.5 Firebase brinda versatilidad y se adapta a la programación en Android Studio puesto que está vinculado en sus librerías.

Se toma Asterisk como plataforma para la gestión de llamadas puesto que implementa el protocolo VoIP SIP, el cual se usa en el presente trabajo, además de que el Raspberry Pi tiene soporte para éste protocolo.

Capítulo 5

Análisis y Mediciones del Sistema de Control de Acceso

En este capítulo se cumple el último objetivo planteado en este trabajo, el cual consiste en evaluar diversos parámetros de calidad que afectan el funcionamiento del sistema en lo referente al control y acceso al hogar. Las mediciones de calidad consideradas son: evaluación de la calidad de audio y video utilizado en el proceso de video-llamada, medición de tiempos de procesamiento y ejecución de la Raspberry Pi, evaluación de la efectividad del algoritmo de reconocimiento facial, medición de tiempos de ejecución de las funciones de control y acceso del sistema en general.

5.1. Introducción

Como se describió en el Capítulo 2, se hace uso de medidas subjetivas y objetivas de calidad para determinar si el video recibido en el dispositivo móvil satisface los requerimientos técnicos y QoS demandado por las aplicaciones de video en tiempo real.

Tanto para la videollamada como para la llamada VoIP se utiliza la herramienta Wireshark para analizar el flujo de paquetes en la red con la finalidad de estimar el *jitter* y la pérdida de paquetes. Se efectuaron pruebas variando el parámetro distancia desde el enrutador central al dispositivo móvil y a la Raspberry Pi para determinar la calidad en el servicio.

OpenCV dispone de funciones de reconocimiento facial, las cuales proporcionan un umbral de exactitud. Se utilizó el umbral para determinar bajo distintas condiciones la efectividad de este algoritmo.

Se requiere velocidad de procesamiento en el sistema, por lo que se efectúan optimizaciones

tanto en hardware (Raspberry Pi 3) como software (OpenCV) con el propósito de acelerar el procesamiento de datos. En este capítulo se presentan los resultados obtenidos con las optimizaciones realizadas.

5.2. Mediciones de Efectividad del Algoritmo de Reconocimiento Facial

En esta sección, se dan a conocer los resultados de las pruebas realizadas para evaluar el rendimiento del algoritmo de reconocimiento de rostros implementado en este proyecto. Para las pruebas se utilizó más de una base de datos con características distintas que serán de utilidad para tener una idea del funcionamiento de los algoritmos frente a diferentes condiciones.

Estudio N°1

Se establece un conjunto de pruebas que corresponde a rostros de un individuo registrado en el sistema, el conjunto se halla conformado por una clase con 10 fotos de un mismo individuo, el tamaño estándar de las fotos son 250×250 píxeles. En este conjunto, no todos los rostros tienen la misma expresión, están situados de manera frontal, además no hay cambio de entorno. Los resultados se muestran en la Tabla 5.1:

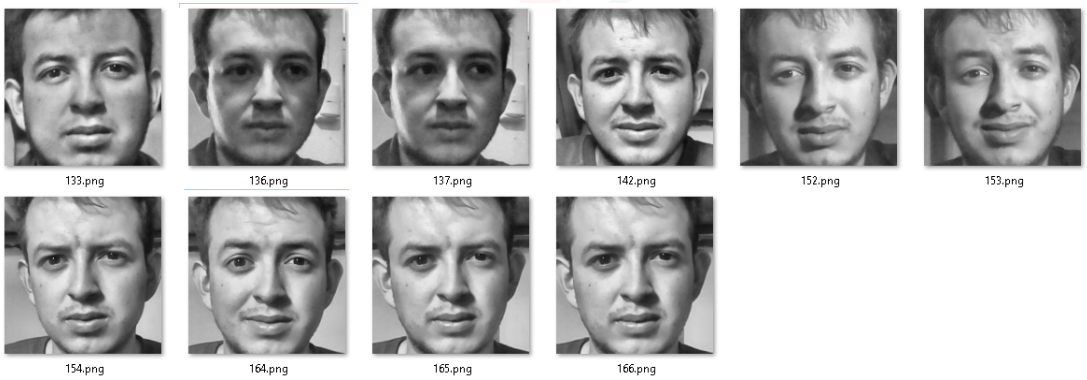


Figura 5.1: Conjunto de pruebas #1

Tabla 5.1: Resultados estudio N°1

# de imágenes de Prueba	Reconocimientos satisfactorios	Promedio de reconocimiento	Distancia (cm)	Efectividad (%)
10	10	23,7	40	97,5
10	9	28,6	70	58,3
10	9	23,5	30	95,8

Para el cálculo de la efectividad se utilizó como referencia la característica de la librería de

reconocimiento facial LBPH, la cual proporciona valores de umbral de reconocimiento por cada fotografía testeada, se tomó como referencia [65], el cual propone que un valor umbral de por debajo de 23 garantiza un 100 % de efectividad en el reconocimiento de rostros y un valor umbral por encima 35 no garantiza el reconocimiento facial (efectividad cercana al 1 %-10 %).

Para cada imagen capturada en tiempo real se evalúa la efectividad una sola vez, la prueba se efectúa con diez fotografías. Posterior al cálculo de la efectividad de cada imagen se promedian los resultados para obtener el valor de efectividad de la prueba. Dicho valor de efectividad se utiliza como referencia para determinar si se concede o no el acceso al usuario.

En la Tabla 5.1 se observa un porcentaje de efectividad alto a pesar de que existen variaciones en la luminosidad de las imágenes, este efecto es compensado con el pre-procesamiento efectuado a cada fotografía. Las fotografías de entrenamiento de cada individuo en el sistema se crearon con imágenes tomadas a distancias inferiores a 45cm, por este motivo la efectividad en el reconcomiendo se reduce conforme la distancia de la cara del individuo se aleja de la camera (mayores a 45cm). Este efecto se da puesto las imágenes capturadas pierden ciertas características distintivas que son detectadas con fotografías tomadas a proximidad.

Estudio N°2

Se efectúa un procedimiento similar al realizado en el estudio 1, se ejecutan pruebas considerando un conjunto de fotos de un individuo no registrado en el sistema, se consideran 10 fotos con tamaño estándar de 250 × 250 pixeles, no todos los rostros tienen la misma expresión, y están situados de manera frontal, además no hay cambio de entorno. Los resultados se muestran en la Tabla 5.2.

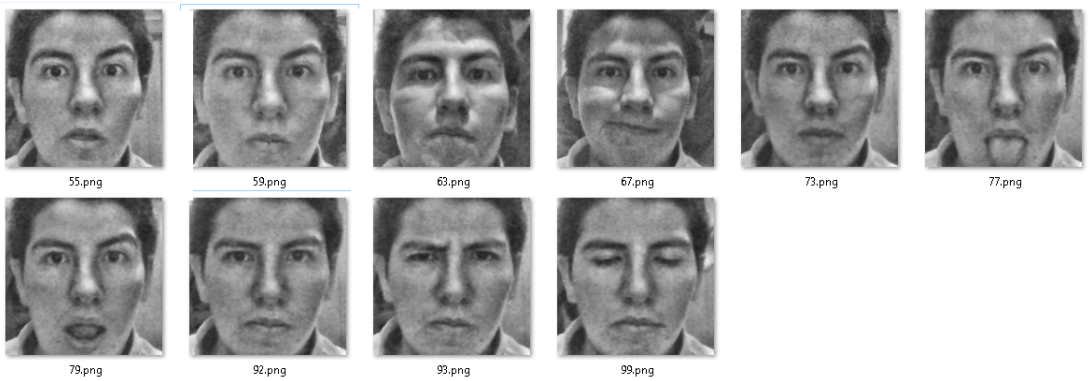


Figura 5.2: Conjunto de pruebas #2

En la Tabla 5.2 se observan porcentajes de efectividad bajos, lo cual comprueba que los rostros no fueron reconocidos con éxito.

Para cuantificar la fiabilidad diagnóstica de las pruebas, se determinan los valores: VP (verda-

Tabla 5.2: Resultados estudio N°2

# de imágenes de Prueba	Reconocimientos satisfactorios	Prom. de reconocimiento	Distancia (cm)	Efectividad (%)
10	9	32,5	40	20,8
10	8	34,8	70	1,73
10	8	33,7	30	10,8

deros positivos), FP (falsos positivos), FN (falsos negativos) y VN (verdaderos negativos), en un conjunto de 7 pruebas realizadas con rostros de usuarios registrados y no registrados en el sistema, los resultados se presentan en la Tabla 5.3

Tabla 5.3: Valores predictivos

	VP	FP	FN	VN	# de caras evaluadas
Prueba 1	10	0	4	2	16
Prueba 2	10	1	1	2	14
Prueba 3	9	2	1	1	13
Prueba 4	9	1	1	1	12
Prueba 5	10	1	1	0	12
Prueba 6	12	1	1	4	18
Prueba 7	11	1	1	5	18

La sensibilidad da una idea de la capacidad del algoritmo para permitir el ingreso a usuarios registrados, en tanto la especificidad indica la capacidad del algoritmo para denegar el acceso a usuarios no registrados; es decir la proporción de usuarios no registrados correctamente identificados. Los valores de sensibilidad y especificidad se detallan en la Tabla 5.4.

La curva ROC de la Figura 5.3 permite disponer de un método estadístico para determinar la

Tabla 5.4: Sensibilidad y especificidad

1 − Especificidad	Sensibilidad	Especificidad
0	0,71	1
0,33	0,90	0,66
0,66	0,9	0,33
0,5	0,9	0,5
1	0,90	0
0,2	0,92	0,8
0,16	0,91	0,83

exactitud diagnóstica de las pruebas realizadas, se observa que el punto de corte en la escala continua en la que se alcanza la sensibilidad y especificidad más alta es (0.2,0.923076923), de igual manera el área bajo la curva alcanza un valor de: 0,90, este valor es cercano a 1, lo cual

garantiza que las pruebas realizadas proporcionan resultados confiables y el algoritmo permite discriminar usuarios registrados y no registrados a lo largo de todo el rango de puntos de corte posibles.

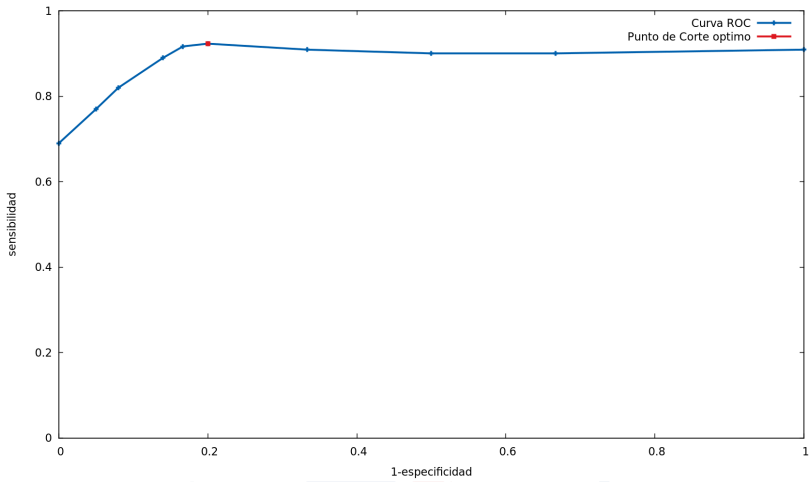


Figura 5.3: Curva ROC del algoritmo utilizado

5.3. Mediciones de la Calidad de Servicio en Llamadas de Voz (VoIP)

Como se describió en el Capitulo 4 se hace uso de Asterisk para el manejo de llamadas VoIP. En el Capítulo 2 se mencionaron algunos conceptos clave que permiten medir la calidad de una llamada, entre los más importantes están: jitter, latencia, cantidad de paquetes perdidos, ancho de banda utilizado en la comunicación y el MOS. En esta sección se calcula cada uno de los factores para diferentes escenarios en una llamada VoIP. Se captura el tráfico VoIP de una llamada considerando distancias diferentes desde la Raspberry Pi y el móvil hacia el enrutador, los resultados se muestran en la Tabla 5.5.

A medida que aumenta la distancia tanto desde el Raspberry Pi como del móvil hacia el enrutador central el *jitter* medio y máximo aumentan de forma gradual, de igual manera la cantidad de paquetes perdidos se ve afectada con la distancia y en especial cuando esta incrementa por encima de los 10m y existe algún tipo de obstáculo que atenúe la señal, en el peor de los casos el número de paquetes perdidos no excede el 2 %.

Parámetros como el Expected value y RTP packets nos indican el número de paquetes RTP que se espera recibir en el lado del receptor y el número de paquetes RTP que llegaron, estos

Tabla 5.5: Mediciones de calidad VoIP

	Distancia entre Raspberry a Router y móvil (4m)	**Distancia entre Raspberry a Router y móvil (6m)**	**Distancia entre Raspberry a Router y móvil (8m) pared**	**Distancia entre Raspberry a Router y móvil (10m) pared**
Max Jitter	30.42 ms	51.54 ms	52.35 ms	62.59 ms
Mean Jitter	26.00 ms	26.13 ms	27.11 ms	27.65 ms
RTP Packets	1190	944	719	1369
Expected	1195	946	727	1378
Lost	5 (0.42 %)	2 (0.21 %)	8 (1.10 %)	9 (0.65 %)

parámetros son de utilidad para estimar las pérdidas en el receptor. Las mediciones MOS se

Tabla 5.6: Mediciones MOS

Distancia entre Raspberry a Router	**Distancia entre Móvil a Router**	**MOS**
4m	4m	4,6
6m	6m	4,2
8m	8m	4
10m	10m	3,7

efectuaron utilizando la herramienta "MOS medition" provista por el softphone Linphone [1], la cual estima la calidad de audio percibida en un rango de 1 a 5, los resultados son verificados con la escucha de la conversación. En la Tabla 5.6 se observa que para distancias mayores a 10m el sonido es degradado por completo haciendo difícil distinguir las palabras en la llamada.

En la Figura 5.4 se observa el tráfico de voz en verde, el tráfico de datos en azul y el tráfico total en rojo en cada instante de tiempo durante una llamada VoIP. Se observa que el ancho de banda demandado por la llamada es relativamente bajo en comparación al usado por los datos.

En la Figura 5.5 se observa que en promedio se requiere alrededor de 900 Kbps de ancho de banda para la transmisión de video desde la raspberry Pi al movil lo cual es un consumo elevado, pero se compensa con la calidad de video percibida por parte del usuario final.

[1] Linphone es un teléfono SIP de código abierto, disponible en entornos móviles y de escritorio (iOS, Android, GNU / Linux, MAC OSX, Windows Desktop, Windows 10 UWP).

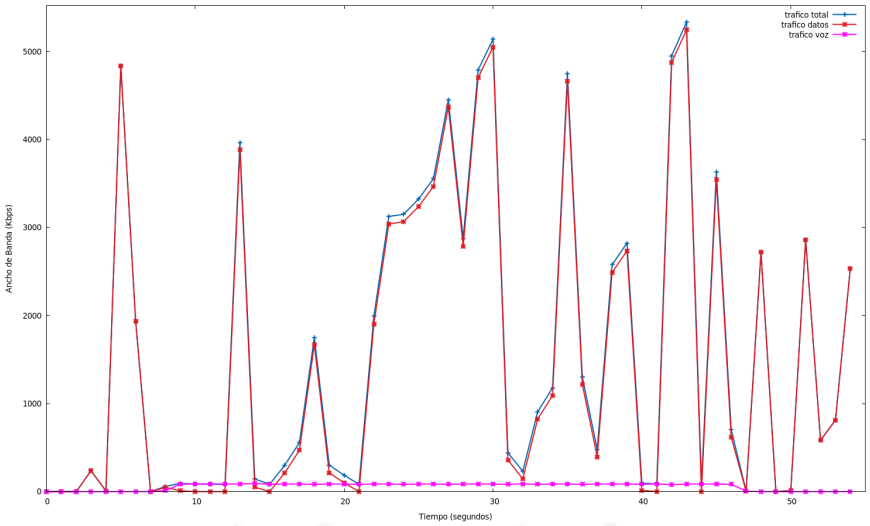


Figura 5.4: Uso de ancho banda: voz y datos.

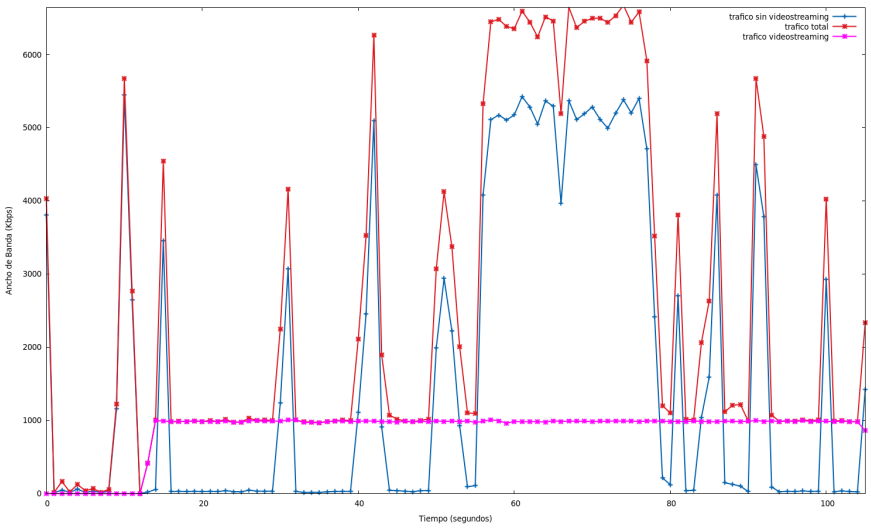
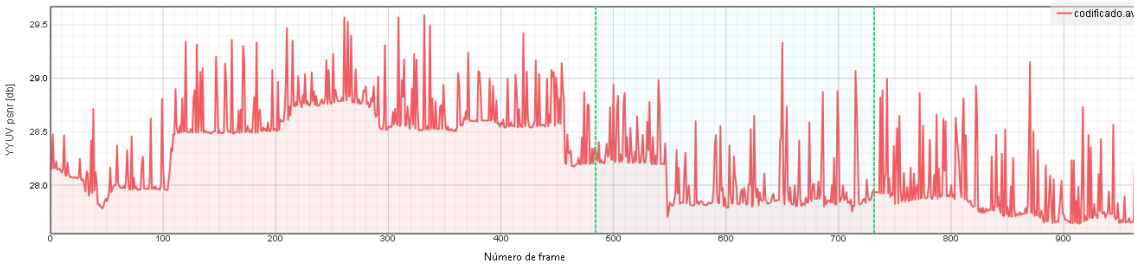


Figura 5.5: Uso de ancho banda: video y datos.

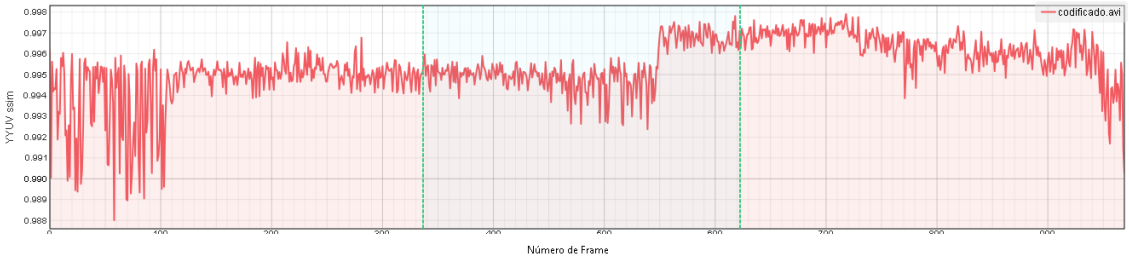
5.4. Mediciones de Calidad en el Servicio de Videostreaming

En el Capítulo 2 se describieron las medidas objetivas de calidad, para determinar si una secuencia de video satisface o no los requerimientos mínimos para efectuar una transmisión de video con calidad aceptable. Se consideran los parámetros PSNR y SSIM, los cuales son los más utilizados para evaluar tanto la cantidad de ruido como la calidad de la imagen en cada

fotograma de una secuencia de video. En la Figura 5.6 se presenta las gráficas de PSNR y SSIM de la secuencia de video original codificada usando el códec MJPEG:



(a) Métrica de calidad: PSNR video codificado



(b) Métrica de calidad: SSIM video codificado

Figura 5.6: Medidas objetivas de calidad para la secuencia de video codificada original

Efectuando la codificación del video original con MJPEG los valores de PSNR para cada trama se hallan por encima de los 20dB, lo cual garantiza fidelidad en la representación de la imagen para transmisiones de *videostreaming*. En lo referente a la métrica de calidad SSIM, los resultados obtenidos son favorables dado dicho índice para cada fotograma del video es un valor decimal cercano a 1, considerando que un valor de 1 solo es alcanzable en el caso de dos conjuntos idénticos de datos.

Se efectúa el mismo estudio en el lado del receptor, para lo cual se almacena y decodifica la secuencia de video en el dispositivo móvil. Con la finalidad de profundizar en el análisis se consideran diferentes escenarios de recepción del video, los resultados se resumen en la Tabla 5.7.

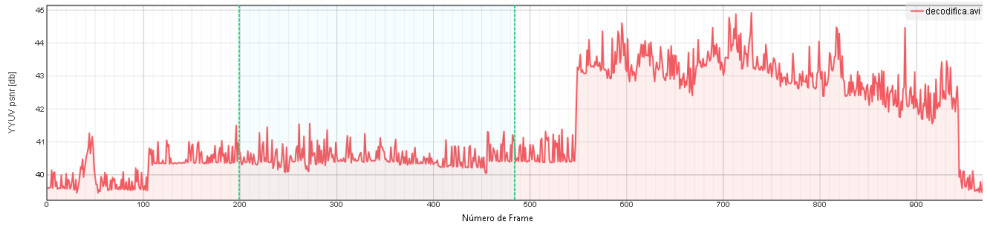
En la Tabla 5.7 se distingue que a medida que incrementa la distancia y el número de obstáculos entre el enrutador central y el dispositivo móvil el ruido crece considerablemente en la señal (PSNR). El índice SSIM para cada caso indica que la estructura y la forma en la que la imagen es percibida por el ojo humano a pesar del ruido presente no afecta en gran medida la transmisión, esto se debe al tipo de codificador utilizado el cual efectúa una compresión intracuadro.

En la Tabla 5.7 se describió un valor promedio de las medidas objetivas de calidad de la secuencia

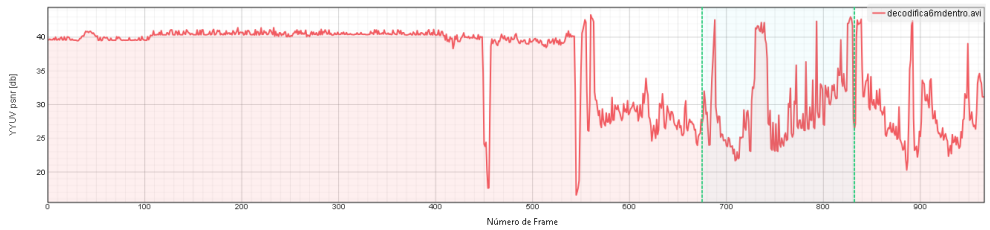
Tabla 5.7: Medidas objetivas de calidad promedio para una secuencia de video en el Receptor

Distancia	PSNR (dB)	SSIM
2m	45,3	0,997
4m	41,2	0,996
6m *	30,4	0,991
8m*	41,2	0,99
10m**	16,45	0,921
Nota: *(una pared); **(dos paredes) entre el enrutador y el móvil.		

de video en el receptor. En las Figuras 5.7 y 5.8 se presentan los valores de PSNR y SSIM para cada fotograma en función del tiempo en los diferentes escenarios de evaluación.



(a) PSNR a d=4m.



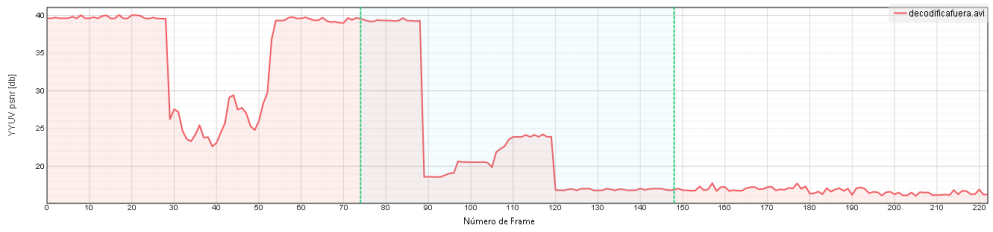
(b) PSNR a d=6m

Figura 5.7: PSNR de la señal decodificada a distancias diferentes desde el enrutador.

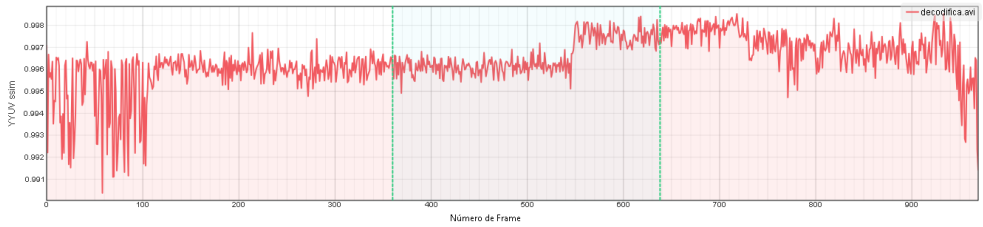
Los resultados del análisis de calidad de video para los diferentes casos considerados permiten concluir que para distancias mayores a 10m y con un número de obstáculos (paredes) mayores a 2 la calidad del video se degrada considerablemente debido a que el PSNR se degrada en estas zonas, para corregir este problema y aumentar el área de cobertura en función del tamaño de la vivienda se puede utilizar enrutadores de mayor alcance (con dos o tres antenas).

El cálculo del PSNR y SSIM se efectuó mediante el uso del software MSU Quality Measurement Tool: Metrics information [2], la cual implementa algoritmos para el cálculo de distintas métricas de calidad de video tales como: PSNR, MSAD, Delta, MSU Blurring Metric, MSU Blocking

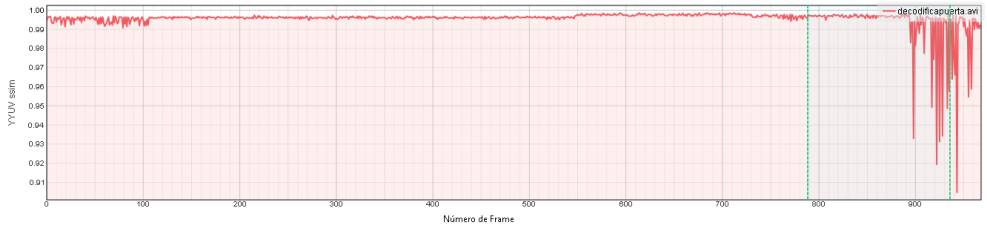
[2] http://www.compression.ru/



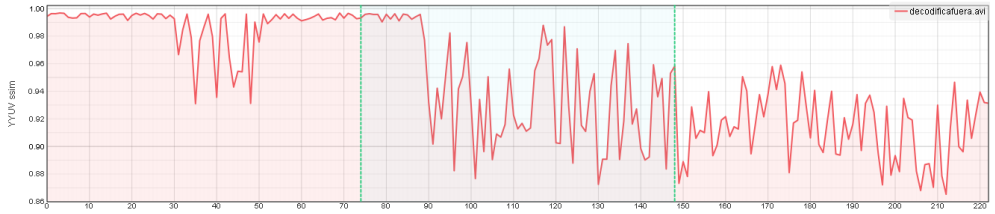
(a) PSNR a d=10m



(b) SSIM a d=4m.



(c) SSIM a d=8m



(d) SSIM a d=10m

Figura 5.8: SSIM y PSNR de la señal decodificada a distancias diferentes desde el enrutador.

Metric, SSIM, MultiScale SSIM, 3-Component SSIM, Spatio-Temporal SSIM, VQM, entre otras. MSU Quality Measurement Tool dispone de una versión gratuita con las funciones de medición de calidad PSNR y SSIM disponibles.

Al tratarse de videostreaming el cálculo del PSNR y SSIM en el receptor se efectuó en dos pasos: se emitió la señal de video desde la fuente hasta el receptor, posteriormente se decodifica la señal y se graba el contenido en un archivo de video sin formato, este archivo es pasado como ingreso al software MSU Quality Measurement Tool.

5.5. Medición: Tiempos de Ejecución y Procesado de las Funciones de Control del Sistema de Acceso Inteligente

En el Capítulo 4 se describieron las funciones de control más importantes en la parte de software del sistema de acceso, es de vital importancia que los tiempos de ejecución sean reducidos y optimizados con la finalidad de proveer calidad en el servicio al usuario. Para satisfacer este requerimiento se puede modificar la velocidad de procesamiento estándar de la Raspberry pi mediante *overclock*, de igual manera la instalación de OpenCV se optimiza a través de los módulos de procesamiento NEON [3] y VFPV3 [4]. La Tabla 5.8 presenta la comparación de velocidad y rendimiento con el uso de *overclock* y NEON/VFPV3 y sin el uso de los mismos:

Tabla 5.8: Tiempos de procesamiento: funciones de control del sistema de acceso

Función de Control	**Sin overclock (seg) ****	**Con overclock(seg) ***
Carga de entrenamiento del módulo LBPHFaceRecognizer	3,35	2,34
Carga de entrenamiento del módulo facial landmark predictor	10,13	9,12
Función llama() (para llamada VoIP)	0,56	0,12
Función seguridad()	5,65	4,69
Función cámara() (directorios y rutas de usuarios registrados)	0,4567	0,3229
Carga de módulos faceCascade() y eyeCascade()	0,56	0,18
Cierre de sockets	0,005	0,005
Apertura de sockets	0,41	0,31
Función Creación de usuario()	26,18	24,23
Función lista de usuarios()	12,34	10,56
Función videostreaming()	3,43	2,49
Función Apertura de puerta principal()	0,03	0,04
Función Apertura de puerta garaje ()	0,05	0,05
Función de entrenamiento ()	13,12	11,23
Procesamiento de imágenes (todas las imágenes)	10,23	8,64
Tiempo total de ingreso al domicilio para usuarios registrados	11,70	9,47
Tiempo total de ingreso al domicilio para usuarios no registrados	15,74	12,09
Nota: * con overclock y NEON/VFPV3 y ** sin overclock y NEON/VFPV3		

En la Tabla 5.8 se analiza el tiempo total de ingreso al domicilio para usuarios registrados y no

[3] ARM NEON: es una extensión de arquitectura optimizada para procesadores ARM. Fue diseñado por los ingenieros de ARM específicamente para un procesamiento de video más rápido, procesamiento de imágenes, reconocimiento de voz y aprendizaje automático.

[4] VFPV3: Es una optimización de punto flotante incluida en el chip de la Raspberry Pi 3

registrados. Es importante analizar el tiempo para usuarios no registrados puesto en caso de que sea denegado el acceso al domicilio se procede a efectuar una videollamada al administrador, se establece la comunicación y se abre la puerta si el administrador así lo dispone, si bien este tiempo puede variar en función del dialogo entre usuario y administrador para efectos de análisis se estima una duración de diálogo de alrededor de 5 segundos. En la tabla se evidencia que la optimización tanto en hardware como software reduce considerablemente el tiempo de acceso final al usuario se encuentre o no registrado en el sistema. El proceso de *overclock* e instalación de OpenCV optimizado se detalla en el Apéndice B.

5.6. Conclusiones

En este estudio se demostró que el algoritmo de reconocimiento facial implementado provee mejores resultados en función de la base de datos con la que se entrena, es decir, si se dispone de fotografías con diferentes niveles de luminosidad de un mismo individuo la probabilidad de reconocimiento exitoso incrementa notablemente.

Las mediciones de calidad de audio para llamadas VoIP brindan un rango de operación dentro del cual se garantiza calidad en el servicio brindado al usuario final. La Tabla 5.5 indica que incluso a más de 10m de distancia y con la presencia de obstáculos(paredes) desde el enrutador central el máximo *jitter* (62,59ms) no sobrepasa los 200 ms valor bajo el cual se considera la calidad de una llamada como deficiente.

Los resultados provistos por la Tabla 5.7 advierten que para distancias mayores a 10m con la existencia de obstáculos (paredes) la calidad de video se degrada de forma considerable, este hecho se verificó en la práctica. Se debe considerar el caso en el que se efectúa la transmisión de video conjuntamente con la llamada VoIP, en dicho caso se debe operar en un rango dentro del cual se garantice tanto calidad en la transmisión de video como de audio.

Capítulo 6

Conclusiones

En este capítulo se presentan las conclusiones finales del trabajo realizado, así como la interpretación de los resultados, las limitaciones que se encontraron en el proceso, y finalmente, se proponen maneras de ampliar la investigación a futuro.

6.1. Conclusiones

En este proyecto se ha implementado un sistema de control inteligente de acceso al hogar que permite al administrador del domicilio autorizar o denegar el acceso a los usuarios no registrados en el sistema.

Las mediciones de calidad realizadas permitieron establecer un rango de operación bajo el cual el sistema provee calidad en el servicio, sin embargo, los resultados pueden variar en función del escenario de operación del sistema. Otro aspecto fue el tipo de hardware utilizado el cual en dependencia de sus características provee una determinada velocidad de procesamiento de datos, se demostró que mediante *overclock* es posible acelerar la Raspberry Pi obteniendo resultados favorables en lo referente a velocidad de procesamiento.

En cuanto a las mediciones de *jitter* y pérdida paquetes en las llamadas es claro que a medida que aumenta la distancia entre el enrutador central, la Raspberry Pi y el móvil tienden a degradarse de forma considerable, en la práctica se comprobó que en tanto el *jitter* sea inferior a 200ms es posible llevar a cabo una llamada VoIP con cierto grado de calidad.

Con las mediciones realizadas en este proyecto sirven como una recomendación al usuario para saber donde colocar el punto de acceso dentro del hogar y así obtener el máximo rendimiento. Si se tratase de implementarlo en edificios una alternativa a la red inalámbrica seria una red

cableada la cual ofrece menos perdidas y se obtendría un mejor rendimiento, debido a que la red inalámbrica podría no tener el alcanze necesario y presentar mucha latencia para las llamadas.

De las pruebas de reconocimiento facial efectuadas, se concluye que para obtener resultados óptimos que permitan asegurar el acceso únicamente a usuarios registrados se debe generar una base de datos que posea fotografías del individuo con diferentes niveles de luminosidad. Como se demostró en las pruebas realizadas el rostro del individuo debe estar situado de manera frontal, y con los ojos abiertos de preferencia.

6.2. Recomendaciones

Existen diferentes variables que pueden ser mejoradas en estudios e implementaciones posteriores y que no fueron tomadas en cuenta en este proyecto por ser una primera implementación del sistema de acceso inteligente. A continuación se presentan las principales:

- En este proyecto se hizo uso de la Raspberry Pi modelo 3 B, la cual brinda gran capacidad de procesamiento y es accesible en costo a cualquier usuario, sin embargo, como se describió en el Capítulo 4 existen alternativas que proporcionan ciertas ventajas tanto en hardware como software, las cuales se deberían considerar para efectuar pruebas de rendimiento.
- La arquitectura implementada en este proyecto es de tipo centralizada, el sistema es controlado por la Raspberry Pi, sin embargo, se puede implementar una infraestructura distribuida para manejar el acceso al hogar y gestionar el proceso de reconocimiento de rostros, este tipo de arquitectura puede mejorar los tiempos de procesamiento y acceso al domicilio.
- El tipo de codec utilizado en la transmisión de *videostreaming* influye en gran medida en la calidad del contenido que se recibe en el lado del receptor, se deben utilizar codecs que gestionen la calidad de cada fotograma por separado y no aquellos que consideran el flujo de fotogramas completo en la secuencia de video.
- En este tipo de sistemas se hace uso de *sockets* para comunicar el dispositivo móvil con la Raspberry Pi, por tal motivo en su gestión se debe considerar los tiempos de retardo que se generan entre dispositivos. Es importante reducir el número de *sockets* al máximo para optimizar el tiempo en el que el usuario requiera los diversas características del sistema.

6.3. Trabajos Futuros

En esta sección, se explica cómo se podría mejorar los resultados obtenidos en el proyecto implementado, se proponen líneas de investigación para futuras mejoras prácticas.

- Como se describió en el Capítulo 4, la seguridad es un parámetro importante. Uno de los aspectos en que se puede trabajar a futuro es el cambio de la cámara de la Raspberry Pi 3 (cámara 2D) por una cámara 3D, con este tipo de cámaras se evitarían falsos positivos, puesto que brindan la capacidad de capturar muchos aspectos faciales, desde la estructura ósea hasta las curvas alrededor de la cuenca del ojo, la nariz y la pera.
- El proyecto realizado funciona en un entorno de red local para la comunicación entre el usuario y el administrador del domicilio, se propone para implementaciones futuras extender la configuración del sistema a internet con la finalidad de efectuar videollamadas y controlar la apertura/cierre de la puerta principal desde fuera del domicilio.
- Para trabajos futuros se puede implementar el servicio de llamada VoIP haciendo uso de IAX en lugar de SIP, IAX entre algunas de las ventajas en comparación a SIP no requiere una configuración tediosa y consume menos ancho de banda que SIP.

Anexos

UNIVERSIDAD DE CUENCA

desde 1867

Apéndice A

Configuración de Dispositivos

En el presente Apéndice se detalla la instalación de Ubuntu Mate en el Raspberry Pi, se aclara que este sistema operativo no tiene soporte oficial por parte de la fundación Raspberry. La imagen de Ubuntu Mate se puede obtener desde el siguiente enlace: `http://ubuntu-mate.org/raspberry-pi/`.

Un requisito importante a tener en cuenta es que la tarjeta de memoria a utilizar en la Raspberry debe ser al menos de clase 8 y como mínimo 16GB de memoria, ya que de lo contrario se tendrá una experiencia muy pobre en rendimiento y velocidad. Como requisito previo se debe formatear la tarjeta de memoria esto se realiza con el software SD FORMATER [66], se conecta la SD con un lector de tarjetas y se selecciona la letra perteneciente a la misma y se pulsa en *format*, el entorno gráfico se muestra en la Figura A.1.

Una vez realizado el formateo se ingresa a:`https://etcher.io/` , se descarga e instala la utilidad Etcher SD Card image. Ejecutamos Etcher y elegimos el archivo imagen de Ubuntu Mate previamente descargado, se elige la unidad de la tarjeta microSD, como se muestra en la Figura A.2.

Finalmente, se hace clic en Flash para transferir la imagen a la tarjeta microSD. El progreso se muestra mediante una barra que te dice cuánto lleva realizado el proceso. Una vez que se acabe, la utilidad automáticamente desmontará la tarjeta microSD de manera que se pueda retirar de forma segura del ordenador.

En la primera puesta en marcha se iniciará el asistente donde se deberá configurar el usuario y la contraseña, para esta tesis se ha colocado como usuario el nombre ***domótica***

Una vez configurado se procede con la instalación, después de reiniciarse al sistema será posible acceder al escritorio donde se mostrará una interfaz como en la Figura A.4

Realizado el proceso antes descrito se ha instalado satisfactoriamente Ubuntu Mate en la Rasp-

Figura A.1: Entorno gráfico SD Card Formatter

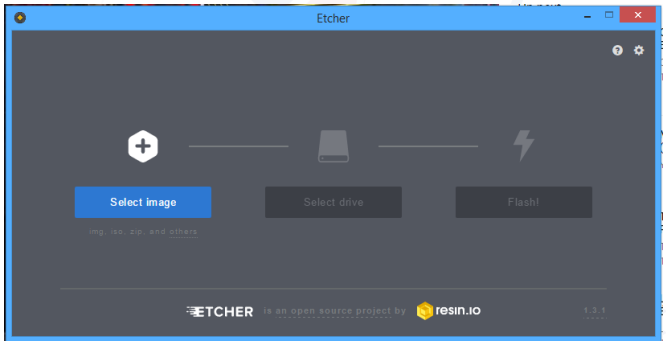


Figura A.2: Entorno gráfico Etcher

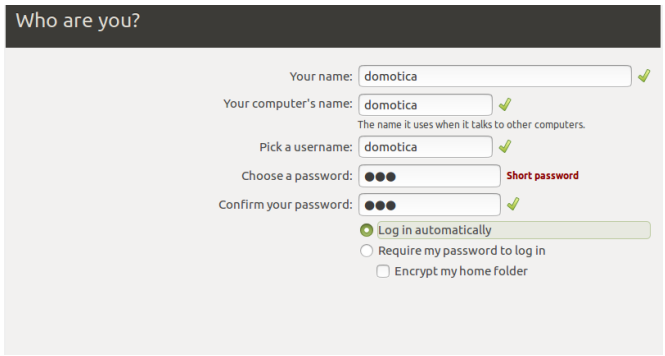


Figura A.3: Configuración de usuario Ubuntu Mate

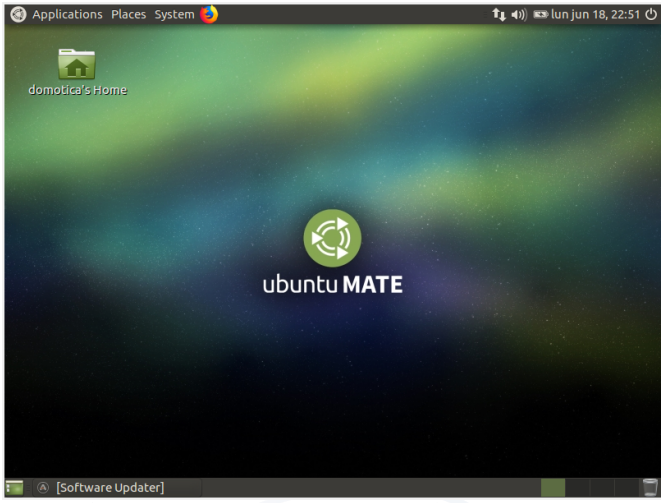


Figura A.4: Escritorio Ubuntu Mate

berry Pi.

Apéndice B

Instalación OpenCV, dlib y overclock en la Raspberry Pi 3

B.1. Instalación OpenCV

B.1.1. Expandir Sistema de Archivos

Como primer paso se debe expandir el sistema de archivos de la Raspberry Pi para utilizar todo el espacio disponible en la tarjeta micro SD:

```
1 $ sudo raspi-config
```

Listado B.1: Configuración de Raspberry Pi 3.

Se selecciona “Opciones Avanzadas ” en el menú como muestra la Figura B.1, luego seleccionar “Expandir Sistema de Archivos ” (Figura B.2) y finalmente hacer click en “Finalizar” y reiniciar la Raspberry Pi.

B.1.2. Instalación de Dependencias

- Como primer paso se debe actualizar los paquetes ya instalados en la Raspberry Pi, refierase a la línea 1 del Listado B.2.
- El siguiente paso es instalar “CMake” la herramienta que nos permite configurar la instalación de OpenCV (Linea 2/ Listado B.2).
- Se instalan los paquetes que permiten cargar diferentes tipos de imágenes (Linea 3/ Listado B.2).

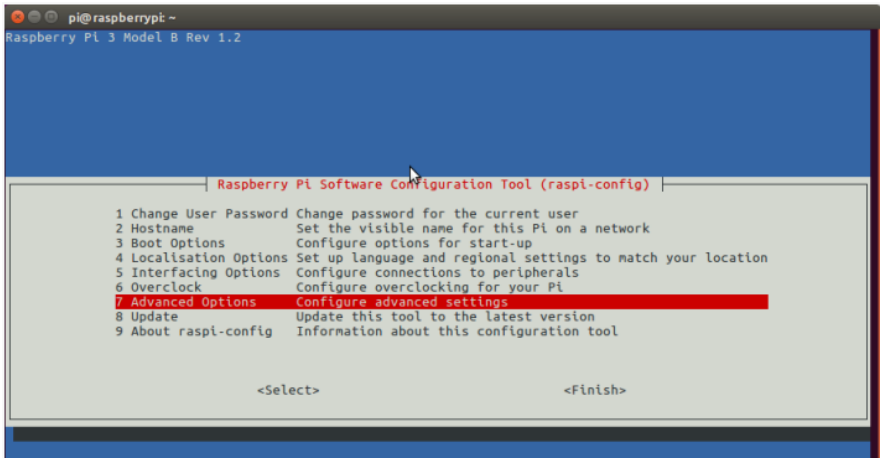


Figura B.1: Selección opciones avanzadas en el menú Raspi-config

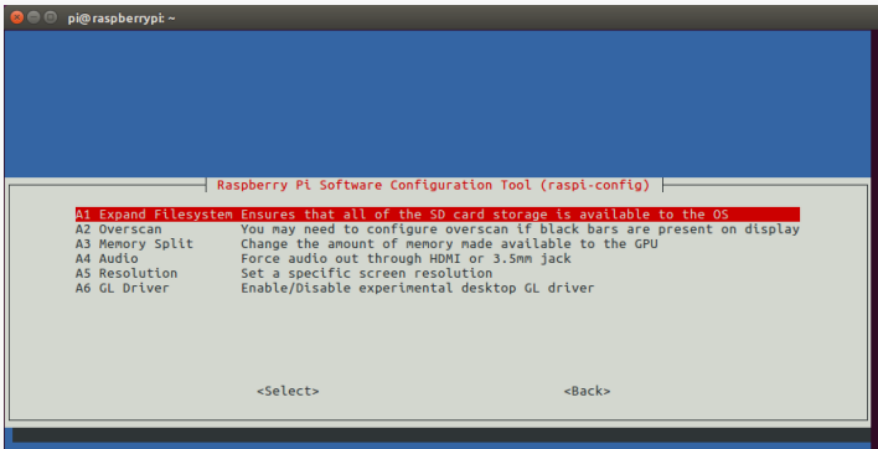


Figura B.2: Menú Expansión sistemas de archivos en la Raspberry Pi 3

- Se instalan los paquetes de vídeo, que permiten leer varios formatos y trabajar con flujos de vídeo directamente (Linea 4 y 5/ Listado B.2).
- Instalar la librería GTK para poder mostrar imágenes generadas por OpenCV (Linea 6/ Listado B.2).
- Para optimizar las operaciones(matrices) dentro de OpenCV se instala gfortran (Linea 7/ Listado B.2).
- Finalmente se instalan las cabeceras de Python 3 para configurar OpenCV y vincularlo con Python (Linea 8/ Listado B.2).

```
1 sudo apt-get update && sudo apt-get upgrade
2 sudo apt-get install build-essential cmake pkg-config
3 sudo apt-get install libjpeg-dev libtiff5-dev libjasper-dev libpng12-dev
4 sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
```

```
sudo apt-get install libxvidcore-dev libx264-dev
sudo apt-get install libgtk2.0-dev libgtk-3-dev
sudo apt-get install libatlas-base-dev gfortran
sudo apt-get install python3-dev
```

Listado B.2: Instalación de dependencias

B.1.3. Descarga e Instalación de OpenCV

Una vez instalado las dependencias, se procede a obtener el código fuente de OpenCV 3.3.0 de su repositorio oficial, para lo cual referirse a las líneas 1-5 del Listado B.3. El siguiente paso es instalar "pip" un gestor de paquetes de Python y "NumPy" un paquete que permite hacer procesos numéricos (líneas 6-8 del Listado B.3).

```
$ cd ~
$ wget -O opencv.zip https://github.com/Itseez/opencv/archive/3.3.0.zip
$ unzip opencv.zip
$ wget -O opencv_contrib.zip https://github.com/Itseez/opencv_contrib
/archive/3.3.0.zip
$ unzip opencv_contrib.zip
$ wget https://bootstrap.pypa.io/get-pip.py
$ sudo python3 get-pip.py
$ pip install numpy
```

Listado B.3: Descarga de OpenCV y Numpy

Para compilar OpenCV se configura el instalador usando CMake, como se muestra en el Listado B.4

```
$ cd ~/opencv-3.3.0/
$ mkdir build
$ cd build
$ cmake -D CMAKE_BUILD_TYPE=RELEASE \
-D CMAKE_INSTALL_PREFIX=/usr/local \
-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib-3.3.0/modules \
-D ENABLE_NEON=ON \
-D ENABLE_VFPV3=ON \
-D BUILD_TESTS=OFF \
-D INSTALL_PYTHON_EXAMPLES=OFF \
-D BUILD_EXAMPLES=OFF ..
```

Listado B.4: Configuración para compilar OpenCV

Se debe verificar que la compilación incluya rutas validas como el Interprete, librerías, NumPy, etc.

Se procede a compilar e instalar OpenCV en el Raspberry Pi mediante las líneas de código del Listado B.5.

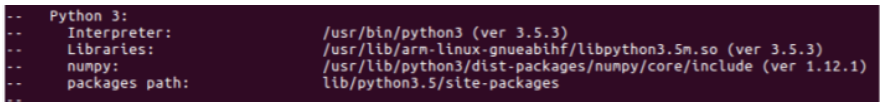


Figura B.3: Revisa que Python 3 será usado para compilar OpenCV

```
$  make -j2
$ sudo make install
$ sudo ldconfig
```

Listado B.5: Compilación de OpenCV

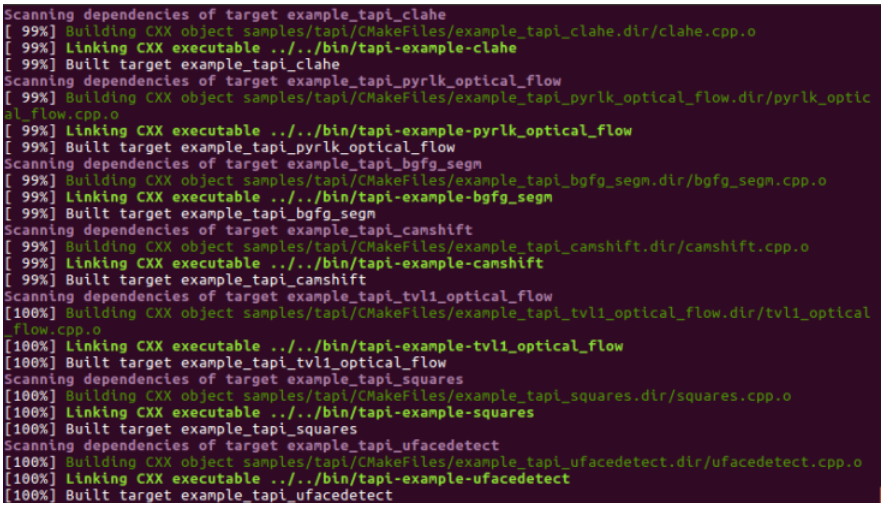


Figura B.4: Compilación OpenCV 3 satisfactoria

Una vez que ejecutados los comandos anteriores, OpenCV + enlaces de Python se instalaran en `/usr/local/lib/python3.5/site-packages`, cuando se termine de compilar se generara un archivo ***.so*** llamado "cv2.cpython-35m-arm-linux-gnueabihf.so".

Para hacer fácil la importación se renombra el archivo a "cv2.so" :

```
$ cd /usr/local/lib/python3.5/site-packages/
$ sudo mv cv2.cpython-35m-arm-linux-gnueabihf.so cv2.so
```

Listado B.6: Configuración del archivo cv2.so

B.1.4. Test de Instalación OpenCV

En el terminal de Ubuntu Mate ejecutamos los comandos del Listado B.7.

```
$ python
>>> import cv2
>>> cv2.__version__
```

Listado B.7: Test OpenCV

Nos devuelve la versión de OpenCV instalada, como se aprecia en la imagen B.5:

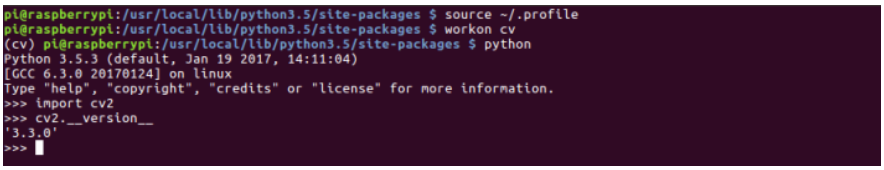


Figura B.5: Confirmación de la instalación de OpenCV.

B.2. Instalación dlib

La librería necesita los siguientes prerrequisitos para su correcto funcionamiento: Boost, Boost.Python, CMake, X11. Estos paquetes se instalan con los comandos del Listado B.8.

```
$ sudo apt-get install build-essential cmake libgtk-3-dev libboost-all-dev
```

Listado B.8: Configuración para compilar OpenCV

El siguiente paso es instalar la librería “SciPy” que trabaja en conjunto con dlib (Listado B.9).

```
$ pip install numpy
$ pip install scipy
$ pip install scikit-image
```

Listado B.9: Instalación de Scipy

Se procede a instalar “dlib”, con el siguiente comando:

```
$ pip install dlib
```

Listado B.10: Instalación dlib

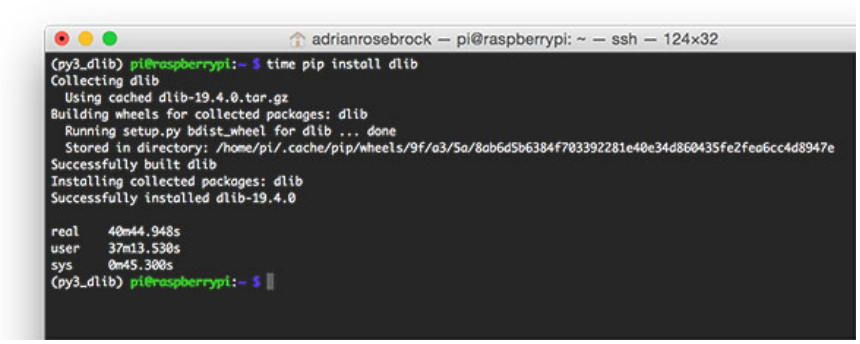


Figura B.6: Compilación exitosa dlib con integración en Python

B.3. Configuración de Overclock en la Raspberry Pi 3

El overclocking, es la forma de aumentar el rendimiento del hardware de Raspberry Pi ajustando varios parámetros del dispositivo. Para este propósito, se requiere hardware adicional (sistemas de enfriamiento y disipadores) y cierto tipo de conocimientos en el manejo de procesadores.

Hardware Adicional para Overclocking

Para hacer overclock a la Raspberry Pi, se debe equipar con tres accesorios esenciales de hardware:

- **Raspberry Pi:** Se utiliza la Raspberry Pi 3 Modelo B V1.2 con Ubuntu Mate ejecutándose.
- **Fuente de alimentación:** Se recomienda una fuente de alimentación confiable, un Pi 3 overclockeado requiere un mínimo de 1.5A o más.
- **Equipo de enfriamiento:** Para evitar el sobrecalentamiento del dispositivo Raspberry Pi, se debe equipar a los chips más importantes con un disipador de calor y con un ventilador,

El proceso de overclock se describe a continuación:

Se actualizan los paquetes instalados a sus últimas versiones y se verifica al principio y final del proceso el rendimiento del sistema mediante la herramienta “sysbench", para lo cual se utilizan los comandos del Listado B.11.

```
$ sudo apt-get update && sudo apt-get dist-upgrade
$ sudo apt-get install sysbench
```

Listado B.11: Proceso: overclock Raspberry Pi

Antes de comenzar a hacer *overclock* a la Raspberry Pi, se preparan y verifican algunas condiciones.

Frecuencia de la CPU

Para conocer la frecuencia en la que la CPU está configurada y ejecutándose, se lee los archivos de proceso cpuinfo_min_freq , cpuinfo_max_freq y cpuinfo_cur_freq del directorio /sys/devices/system/cpu/cpuf/cpufreq/, donde:

- cpuinfo_min_freq - es la frecuencia mínima para el modo ïnactivo".
- cpuinfo_max_freq - es la frecuencia máxima.
- cpuinfo_cur_freq - es la frecuencia de ejecución actual de Raspberry Pi.

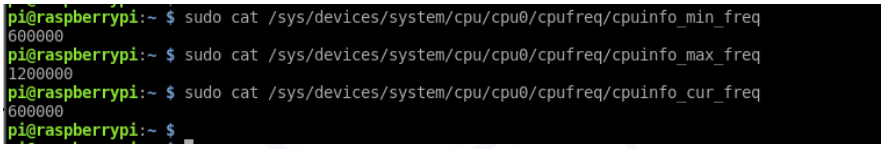


Figura B.7: Monitoreo de frecuencia de la CPU del Raspberry Pi

Temperatura de la CPU

Para conocer la temperatura actual de la CPU, se ejecuta el comando vcgencmd measure_temp.

```
$ while true ; do vcgencmd measure_temp ; sleep 1 ; done
```

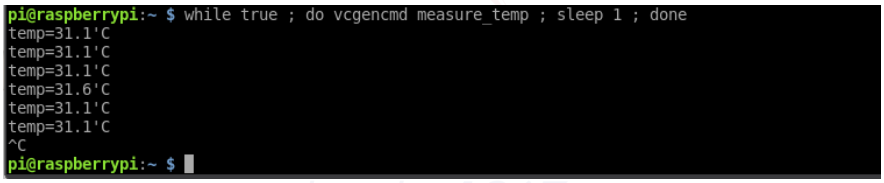


Figura B.8: Monitoreo de temperatura de la CPU del Raspberry Pi

Una vez verificadas las condiciones de funcionamiento del Raspberry Pi 3 se procede a modificar los valores de trabajo por defecto del mismo, de acuerdo a la configuración del Listado B.13.

Configuración por defecto del Raspberry Pi 3:

```
arm_freq=1200
gpu_freq=400
core_freq=400
sdram_freq=450
over_voltage_sdram=0
```

Listado B.12: Configuración inicial Raspberry Pi 3

Se procede a cambiar estos valores por defecto modificando el archivo: /boot/config.txt de la siguiente manera:

```
arm_freq=1300
gpu_freq=500
sdram_freq=500
over_voltage_sdram=0
```

Listado B.13: Ajuste de configuración Raspberry Pi 3

Apéndice C

Instalación Servidor Asterisk y Linphone

C.1. Instalación de Asterisk

Previo a la instalación de Asterisk es necesario instalar algunos paquetes complementarios los cuales se proveen en la Lista C.1.

```
$ apt-get install build-essential
$ apt-get install openssl libxml2-dev libncurses5-dev uuid-dev sqlite3
libsqlite3-dev pkg-config libjansson-dev subversion libiksemel-dev
libspeex-dev libssl-dev libmyodbc unixodbc-dev libsrtp0-dev
```

Listado C.1: Pre-requisitos Asterik

Se procede a instalar Asterisk mediante el comando:

```
$ apt-get -y install asterisk
```

Listado C.2: Instalación de Asterik

Configuración de Extensiones

Una vez instalado Asterisk se debe configurar las extensiones y el puerto de escucha que se traducen en las cuentas para los usuarios de nuestra central, los archivos a configurar son los siguientes:

- sip.conf
- extensions.conf

Modificación Sip.conf

Se debe modificar el archivo "sip.conf" ubicado en la ruta `/etc/asterisk/sip.conf` para lo cual nos referimos a Lista C.3.

```
[general]
udpbindaddr=0.0.0.0:5060
context=default
srvlookup=yes
allowguest=no
alwaysauthreject=yes

[7001]
type=friend
host=dynamic
username=7001
secret=123
callerid="Servidor" <7001>
context=extensiones-internas
canreinvite=no

[7002]
type=friend
host=dynamic
username=7002
secret=123
callerid="Usuario1" <7002>
context=extensiones-internas
canreinvite=no
```

Listado C.3: Configuración de la extensión sip.conf

Modificación Extensions.conf

Este archivo ubicado en la ruta `/etc/asterisk/extensions.conf` maneja las acciones de nuestra central, su configuración se muestra en la Lista C.4.

```
[general]
static=yes
writeprotect=yes
autofallthrough=yes
clearglobalvars=no
priorityjumping=no

[default]
; Recibe lo que no tiene un contexto propio definido.
; Rechaza todo por seguridad.
exten => _X.,1,Hangup(21)
exten => s,1,Hangup(21)

```

```
[extensiones-internas]
; Extensiones internas SIP
exten => _7XXX,1,Dial(SIP/${EXTEN})
same => n,Hangup(16)
```

Listado C.4: Configuración de la extensión extensions.conf

Para que todas las configuraciones tengan efecto se reinicia Asterisk con el siguiente comando:

```
$ /etc/init.d/asterisk restart
```

Listado C.5: Reinicio de Asterisk

C.2. Instalación Linphone

Se procede a instalar el teléfono VoIP ***Linphone***, su instalación se realiza desde con el siguiente comando:

```
$ sudo apt-get install linphone-nogtk
```

Listado C.6: Instalación de Linphone

Linphone contiene interfaz gráfica, pero solo necesitamos la versión disponible desde línea de comandos la cual se instala con la línea de comando anteriormente detallada, como primer paso se debe configurar el puerto de comunicación puesto su puerto por defecto 5060 está ocupado por el servidor Asterisk:

```
linphone
ports sip 5055
exit
```

Listado C.7: Configuración de Linphone

La instalación ha sido completada satisfactoriamente y ya es posible configurar el cliente SIP.

Apéndice D

Funciones de Control en Python

D.1. Control de Videostreaming y Apertura/Cierre de Puertas

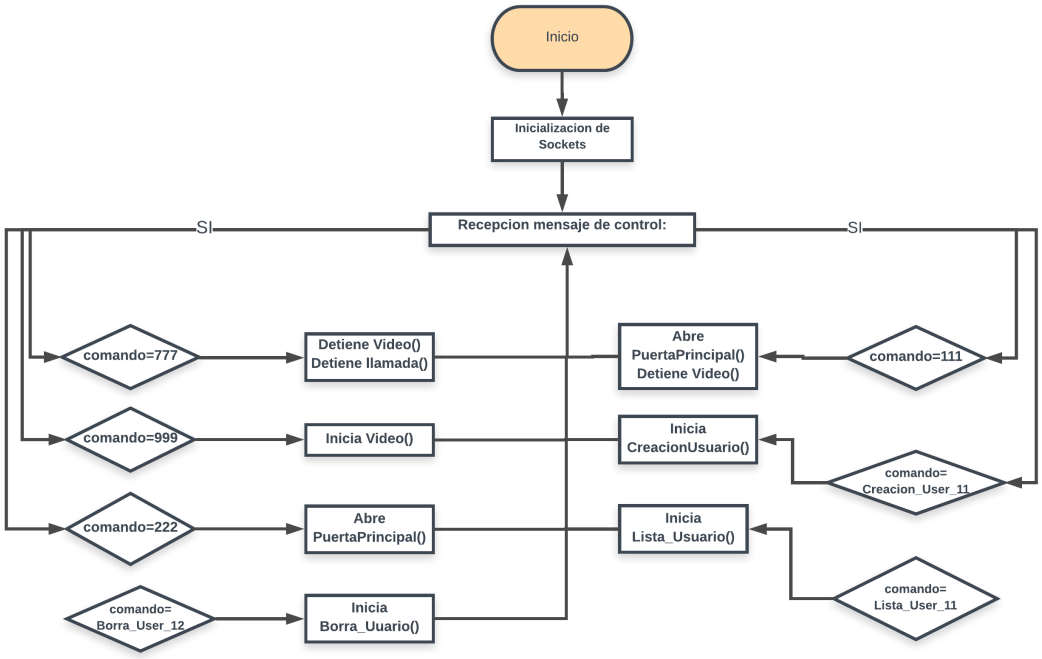


Figura D.1: Diagrama de flujo: función videostreaming y apertura/cierre de puertas

El diagrama de flujo de la función se presenta en la Figura D.1. Esta función se encarga de recibir una solicitud por parte del usuario y ejecuta un determinado proceso en función de la orden recibida. Se dispone de tres funciones básicas: inicializar o terminar videostreaming o en caso de recibir una llamada por parte de un usuario externo se puede rechazar dicha petición.

D.2. Reconocimiento de Rostros

Esta sección explica el código implementado para efectuar el reconocimiento de rostros utilizando la librería OpenCV. Para una mejor representación en la Figura D.2 se hace uso del diagrama de flujo para explicar como funciona el algoritmo de reconocimiento de rostros implementado.

D.3. Registro y Entrenamiento de usuarios en el Sistema de Acceso

Este conjunto de funciones creará la base de datos de los usuarios, para luego ser usada en la autentificación de rostros, en la Figura D.3 se detalla el diagrama de flujo de este proceso.

D.4. Seguridad en el Sistema de Acceso

La funciones de seguridad se ejecuta para comprobar que la foto del usuario capturado no es una imagen fija o falsa. Se identifica si se ha efectuado movimiento de labios, en caso de cumplir con el requisito continua tomando fotos para realizar el reconocimiento, caso contrario se obliga a presionar el botón de ingreso nuevamente.El diagrama de flujo de la función se presenta en la Figura D.4.

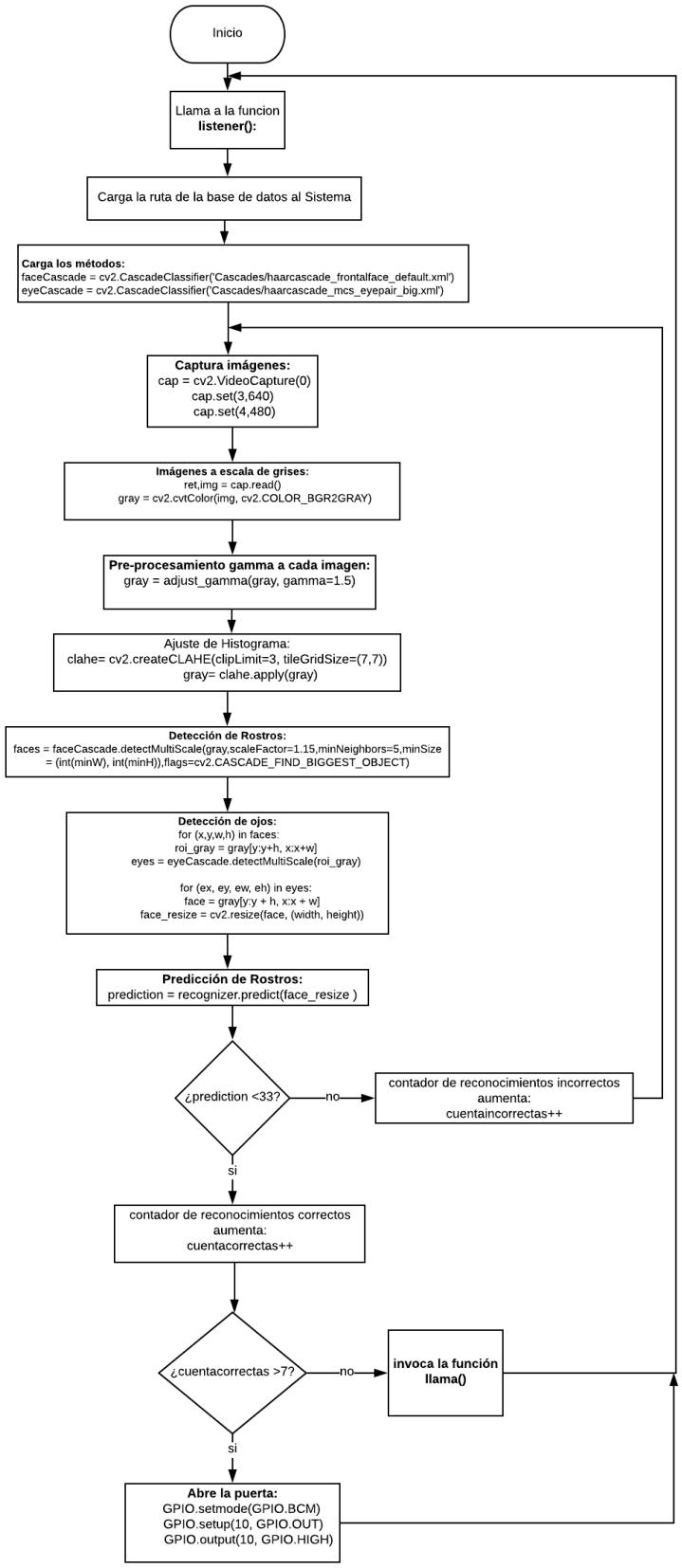


Figura D.2: Diagrama de flujo: función camara()

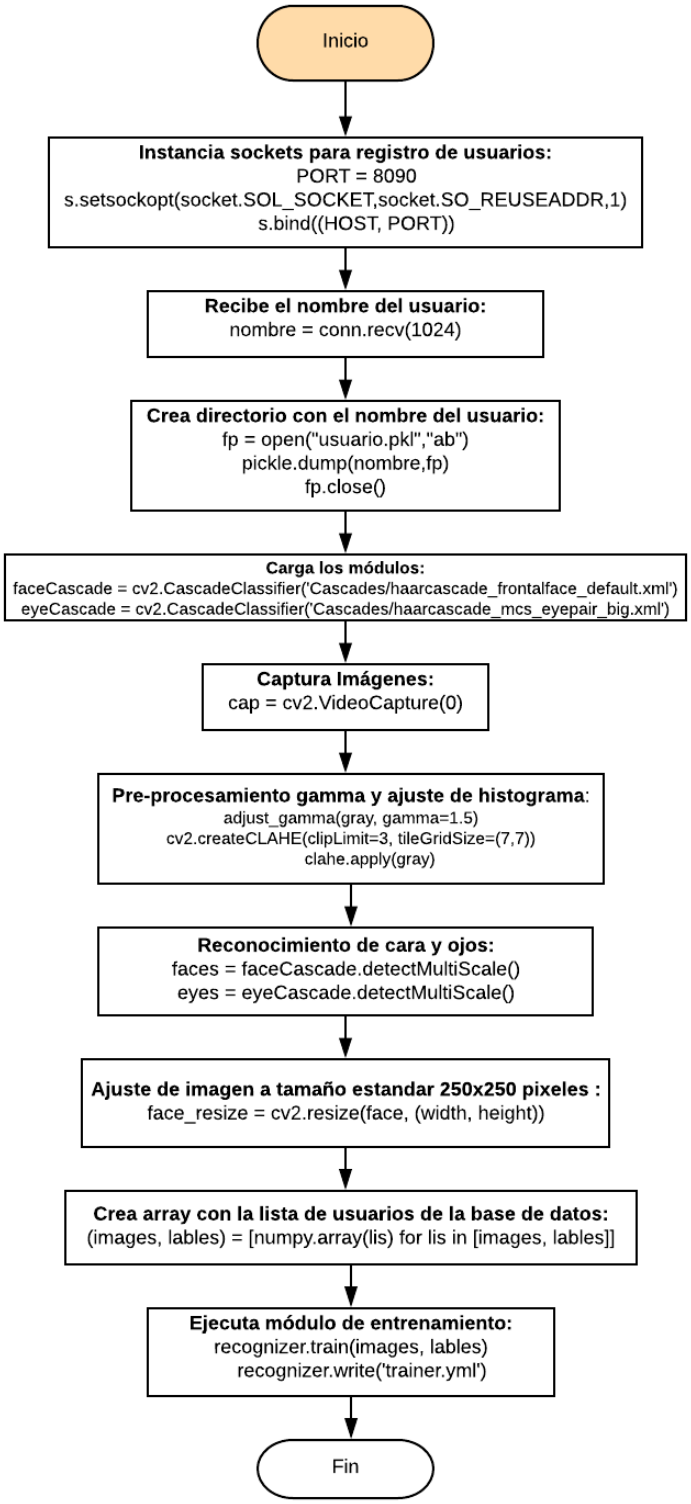


Figura D.3: Diagrama de flujo: registro y entrenamiento de usuarios.

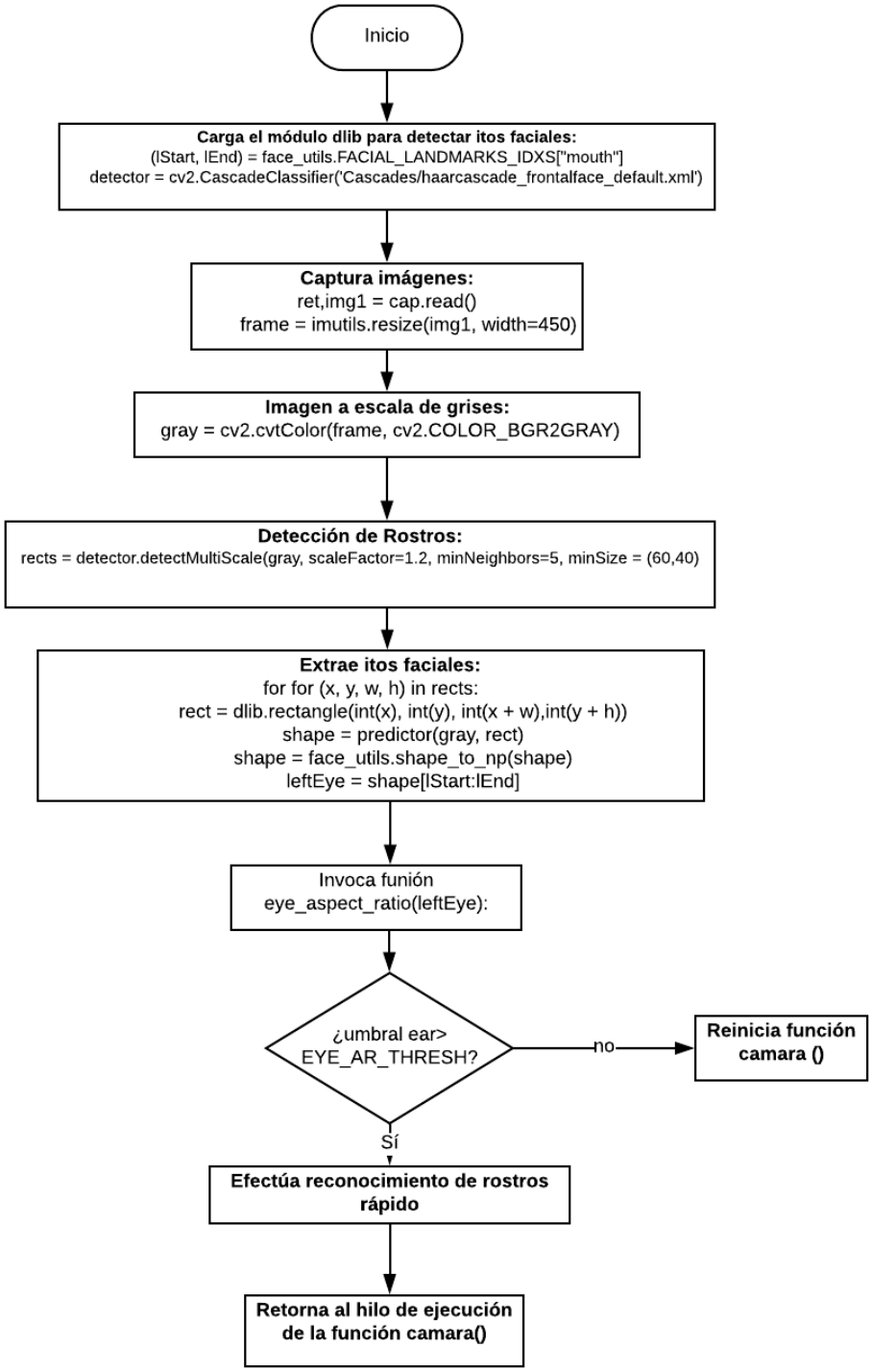


Figura D.4: Diagrama de flujo: función seguridad

Apéndice E

Configuración Firebase

E.1. Registro de Usuarios en Firebase y Creación de Nuevos Proyectos

Para crear una cuenta en Firebase, nos dirigimos al enlace: https://firebase.google.com/, se nos pide identificarnos con una cuenta de Google con la cual automáticamente Firebase crea un proyecto, se brinda la opción de crear nuevos proyectos para lo cual se escribe un nombre único para cada proyecto, posterior a este paso se genera una URL, que es la que se utilizará para hacer peticiones y consultas en tiempo real.

1. Acceso a https://firebase.google.com/:

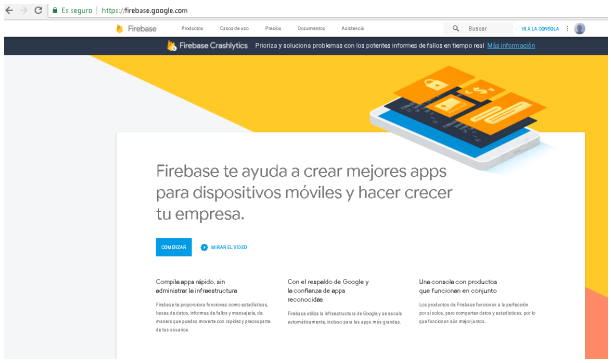


Figura E.1: Página de registro de Firebase

2. Registro en Firebase con cuenta de Google:
3. Creación de un nuevo proyecto en Firebase:

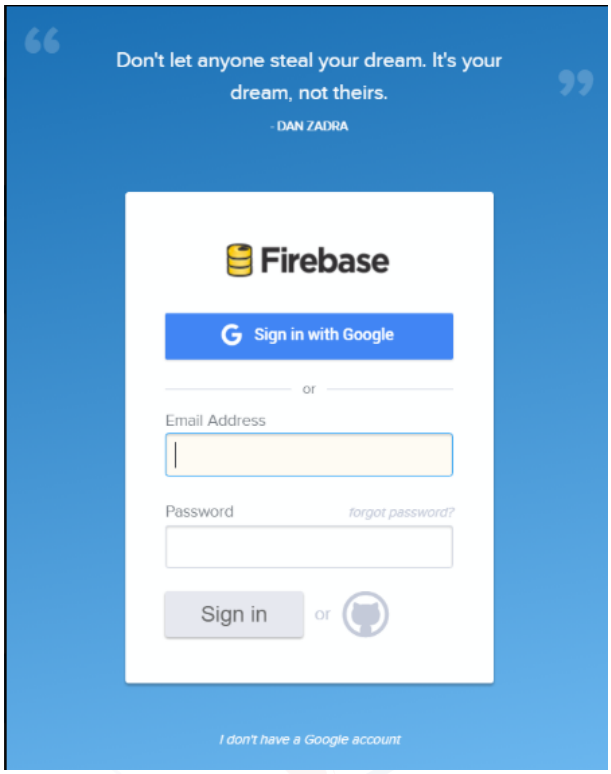


Figura E.2: Registro en Firebase

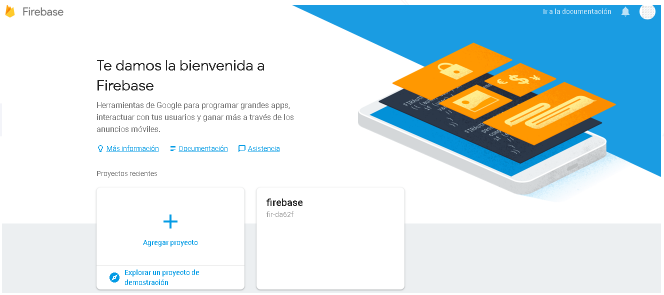


Figura E.3: Nuevo proyecto Firebase

4. Escritura de variables de control y obtención del URL para el manejo de peticiones en tiempo real:

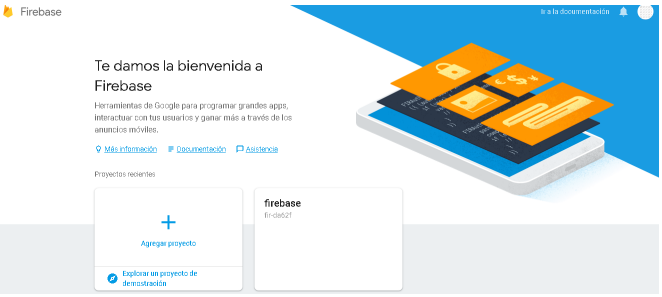


Figura E.4: Creación de variables de control

E.2. Instalación de Librerías Necesarias para el Uso de Firebase.

Para utilizar Firebase en la Raspberry Pi se requiere actualizar e instalar algunas dependencias, se utilizan los comandos de la Lista E.1.

```
sudo apt-get update
sudo apt-get install python-dev
sudo apt-get install python-gpiozero
sudo wget https://bootstrap.pypa.io/get-pip.py
sudo python get-pip.py
sudo pip install requests==1.1.0
sudo pip install python-firebase
```

Listado E.1: Librerías necesarias Firebase

Configuración de Firebase en python

Para el manejo de la variable de apertura y cierre de la puerta del garaje en Firebase, se implementa un script en python que permiten visualizar y modificar el estado de dicha variable, en la Figura E.5 se aprecia el diagrama de flujo de este proceso, cabe destacar que la lectura se la variable se realiza al solicitar abrir la puerta.

Configuración de Firebase en Android Studio

Para configurar y vincular Firebase al aplicativo móvil se efectúan los siguientes pasos:

- Abrir un proyecto existente en Android Studio, en este caso el proyecto CSipSimple ya modificado para este proyecto:
- Hacer clic en Herramientas >Firebase para abrir la ventana de Assistant.
- Hacer clic y expandir la función Real Time Database, posteriormente seleccionar el vínculo del instructivo proporcionado como se muestra en la Figura E.8:
- Hacer clic en conectarte a Firebase y seguir los pasos para agregar código a la app como se muestra en la Figura E.9. El código que se visualiza en las figuras E.10 y E.9 se modifica para manejar la variable de apertura y cierre de la puerta del garaje de la misma manera

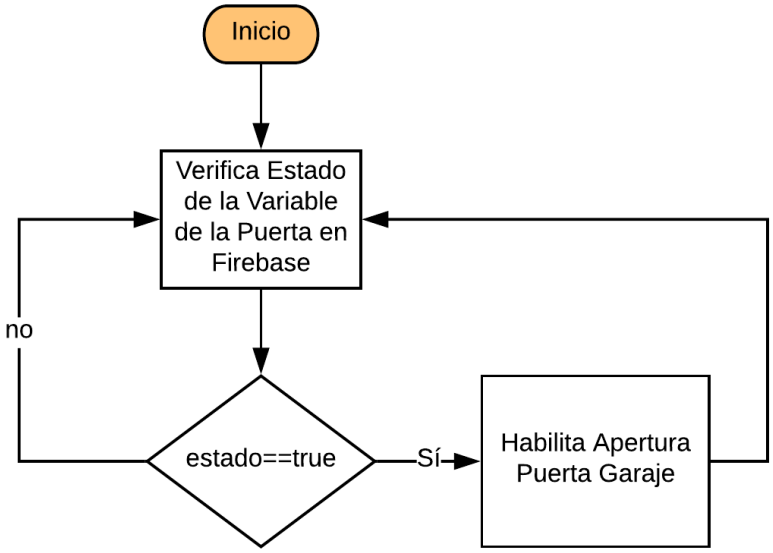


Figura E.5: Diagrama de flujo apertura de puerta

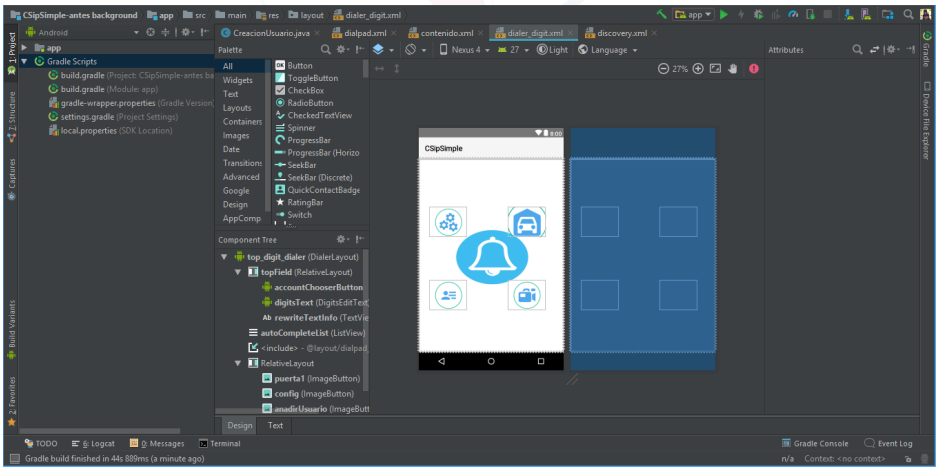


Figura E.6: Proyecto CSipSimple Android Studio

efectuada con python.

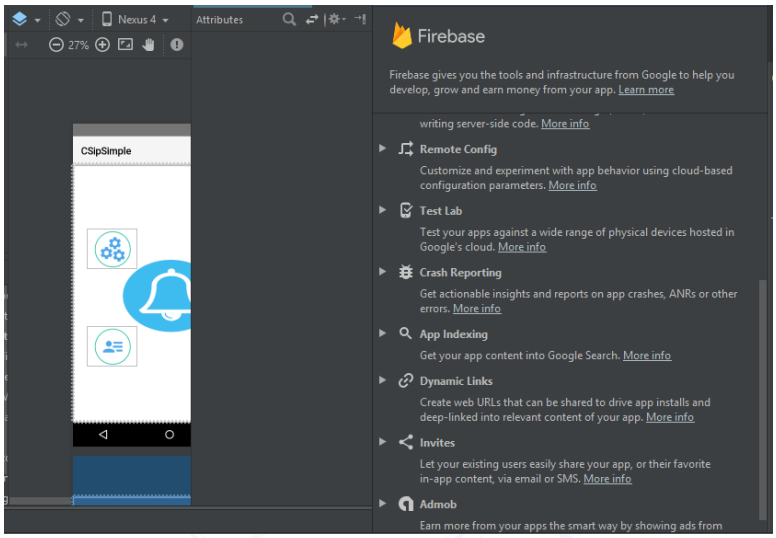
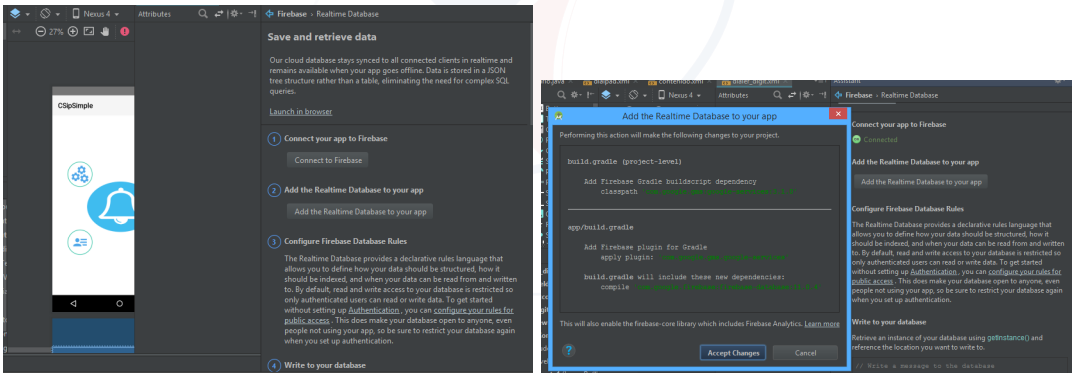


Figura E.7: Asistente de Firebase



(a) Configuración de la función: Real Time Database (b) Configuración de Android Studio para la función Real Time Database

Figura E.8: Configuración de Real time Database

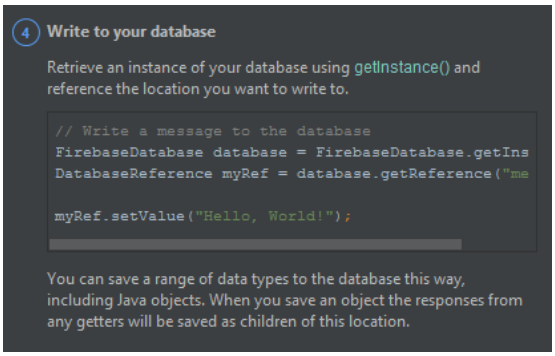


Figura E.9: Generación de código estándar: Firebase Database

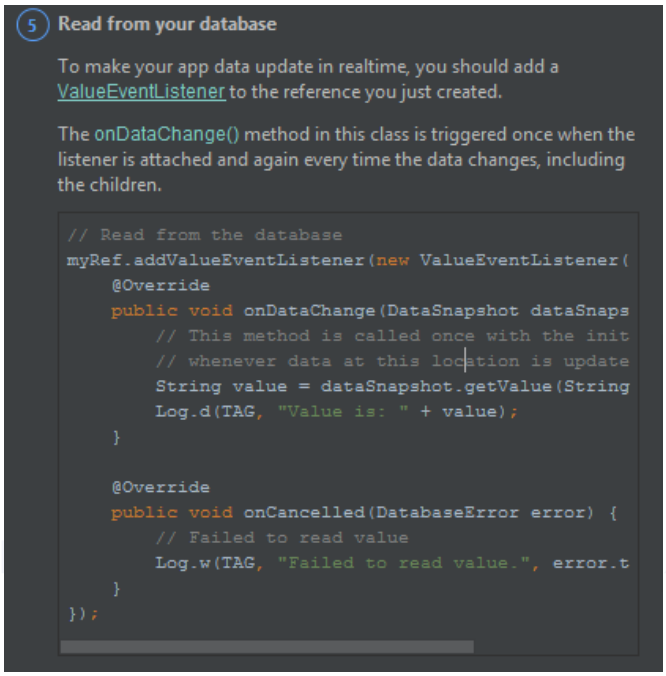


Figura E.10: Generación de código estándar: Firebase Database

Apéndice F

CSipSimple

Como primer paso se debe obtener los archivos necesarios para cargar el proyecto en Android Studio, nos dirigimos al repositorio del proyecto ubicada en la siguiente dirección:
https://github.com/tqcenglish/CSipSimple.git

Una vez ahí descargamos el proyecto, haciendo clic ***Download Zip*** como muestra la siguiente gráfica:

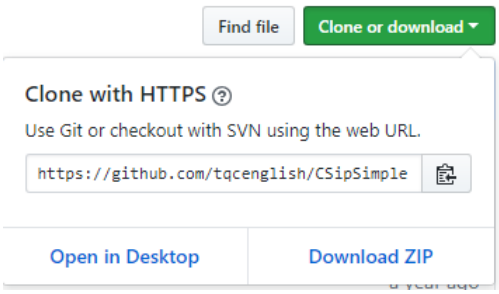


Figura F.1: Descarga código desde repositorio

Realizada la descarga se desconprime en un directorio de trabajo desde el cual cargamos el proyecto al IDE haciendo en clic en **Importar Proyecto**:

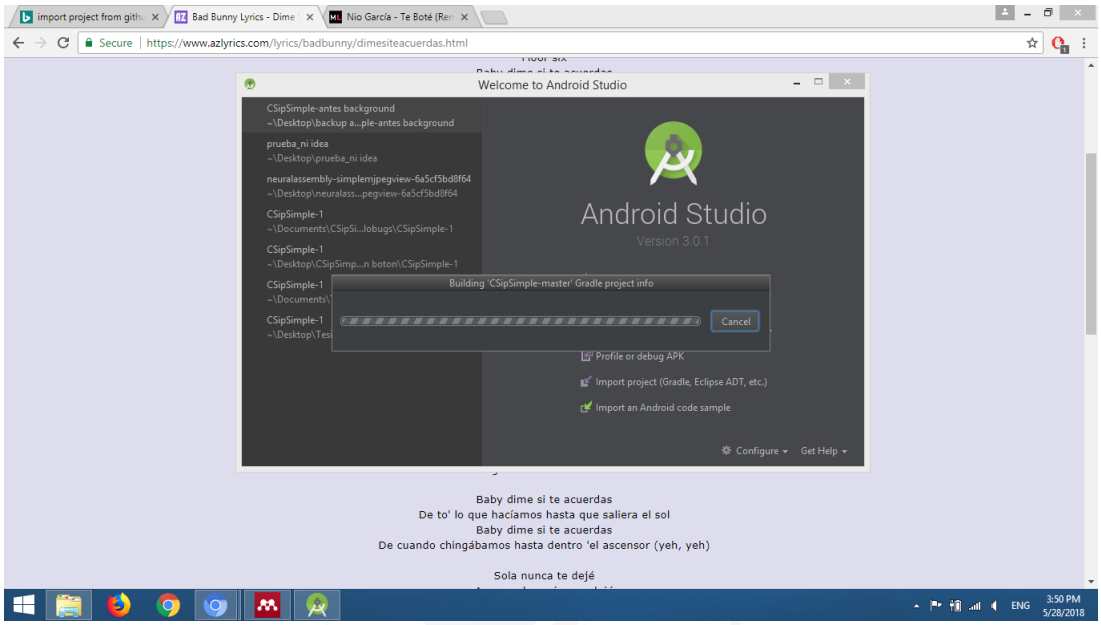


Figura F.2: Importación proyecto en Android Studio

Al cargar el proyecto se presentará un error de versión de compilación, se debe hacer clic en el mismo y automáticamente el IDE instalará los paquetes necesarios:

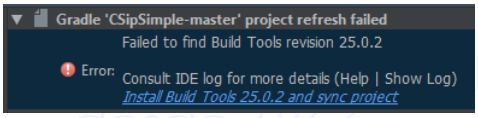


Figura F.3: Error Build Tools

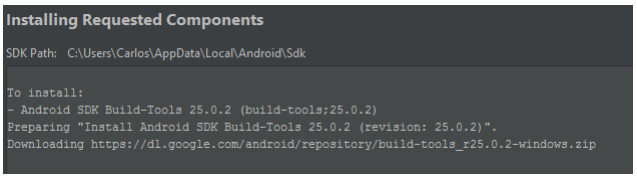


Figura F.4: Instalación componentes necesarios

En la pestaña `build.gradle:Module app` se debe configurar como se muestra en la Figura F.5, de esta manera se asegura el correcto funcionamiento de la misma ya que se agrega soporte para ***firebase*** y a otras librerías necesarias en el proyecto:

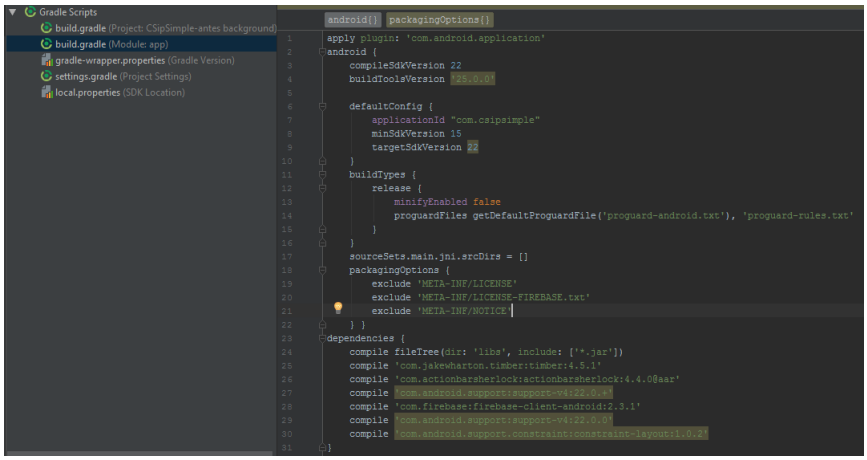


Figura F.5: Configuración build.gradle: Module app.

F.1. Soporte MJPEG

Se añade soporte para decodificar el vídeo que se envía desde la Raspberry Pi, el cual se halla en formato MJPEG, para ello se importan dos clases llamadas ***MjpegView.java*** y ***MjpegInputStream.java*** del siguiente repositorio https://bitbucket.org/neuralassembly/simplemjpegview.

La configuración más importante se resume en las líneas de código de la Lista F.1.

```
requestWindowFeature(Window.FEATURE_NO_TITLE);
getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
WindowManager.LayoutParams.FLAG_FULLSCREEN);
mv = (MjpegView) findViewById(R.id.mv);
mv.startPlayback();
```

Listado F.1: Configuración para soporte MJPEG

Esta configuración corresponde a la decodificación de vídeo dentro del activity, donde ***mv*** corresponde al cuadro donde se visualizará el vídeo, ***MjpegView*** es la llamada a la clase principal, la última línea inicia el procesado de vídeo para mostrarlo al usuario, el resto de código se puede revisar en la clase **in_Call_Card**.

F.2. Soporte Sockets

Para la comunicación con el servidor se utilizan códigos que permiten el envio y recepción de mensajes para cada uno de los servicios del sistema, la manera más fácil de enviarlos es usando

sockets, en las siguientes líneas se muestra una porción del código utilizado en la que se detalla la *dirección IP* y el ***puerto***, finalmente con el método ***execute*** se especifica el mensaje a enviar.

```
MyClientTask myClientTask = new MyClientTask(hostname,
Integer.parseInt(puerta));
myClientTask.execute(mensaje);
```

Listado F.2: Configuración de sockets

F.3. Soporte Configuración Automática de IP

La aplicación necesita la conocer la dirección *IP* del servidor para conectarse y poder obtener los diversos servicios, para ello se utilizó una herramienta de redes la cual se encarga de enviar pings a todas las direcciones posibles de la red, para de esta manera obtener su dirección *MAC* y por ende la *IP*. Acontinuacion se muestra una porción del código perteneciente a la comparación de la *MAC*, el código se puede obtener del siguiente repositorio https://github.com/rorist/android-network-discovery, cabe destacar que tiene licencia *GLP* por lo tanto se puede modificar y usar como se ha lo hecho en esta tesis:

```
String fabricante=host.hardwareAddress.toString();
     boolean h=  new String(fabricante).equals("b8:27:eb:3e:a6:6a");
            if(h== true) {
Toast.makeText(ActivityDiscovery.this, "Raspberry Encontrado ",
Toast.LENGTH_LONG).show();
holder.texto = (TextView) convertView.findViewById(R.id.ipmia);
holder.texto.setText(host.nicVendor);
holder.texto1 = (TextView) convertView.findViewById(R.id.macmia);
holder.texto1.setText(host.ipAddress);}
```

Listado F.3: Configuración de ip automática

Bibliografía

[1] M. F. T. Domínguez, "Estudio Y Diseño De Domótica Para El Conjunto Villa Navarra," Master's thesis, Pontificia Universidad Católica Del Ecuador, 2016.

[2] P. J. Phillips, P. Grother, R. Micheals, D. M. Blackburn, E. Tabassi, y M. Bone, "Face recognition vendor test 2002," in *2003 IEEE International SOI Conference. Proceedings (Cat. No.03CH37443)*, 2003. [En línea]. Disponible: http://www.face-rec.org/vendors/FRVT{_}2002{_}Evaluation{_}Report.pdf

[3] M. L. GUEVARA, J. D. ECHEVERRY, y W. ARDILA URUEÑA, "Detección De Rostros En Imágenes Digitales Usando Clasificadores En Cascada," *Scientia et Technica*, num. 38, pp. 1–6, 2008.

[4] Javier Vázquez Míguez, "Autenticación biométrica en redes sociales: Diseño e implementación de reconocimiento facial mediante EmguCV para autenticación en redes sociales." Ph.D. dissertation, Universidad Carlos III de Madrid, 2014. [En línea]. Disponible: https://e-archivo.uc3m.es/bitstream/handle/10016/22496/PFC{_}Javier{_}Perez{_}Miguez{_}2014.pdf?sequence=1{&}isAllowed=y

[5] A. Johnston, *SIP: Understanding the Session Initiation Protocol*, ser. Artech House telecommunications library. Artech House, 2009. [En línea]. Disponible: https://books.google.com.ec/books?id=AKDgVrDz9mYC

[6] I. G. Francisco, "Desarrollo de una herramienta para la medida de calidad de vídeo," , Dep. Teoría de la Señal y Comunicaciones. Escuela Técnica Superior de Ingeniería. Universidad de Sevilla, Mayo 2017.

[7] RASPBERRY PI FOUNDATION, "RASPBERRY PI 3 MODEL B+," 2018. [En línea]. Disponible: https://www.raspberrypi.org/

[8] Company Bio, "Banana Pi Single Board Computers," 2018. [En línea]. Disponible: http://www.banana-pi.org/

[9] Shenzhen Xunlong Software CO.,Limited, "Orange Pi Pc Plus," 2018. [En línea]. Disponible: http://www.orangepi.org/

[10] Naylamp, "Esp-8266," 2018. [En línea]. Disponible: https://naylampmechatronics.com/inalambrico/48-modulo-wifi-serial-esp8266.html

[11] F. Mocq, *Raspberry Pi 2 : utilice todo el potencial de su nano-ordenador.* Ediciones ENI, 2016.

[12] Montalvo Omar;Vicente Byron;, "DISEÑO E IMPLEMENTACIÓN DE UNA APLICACIÓN DISTRIBUIDA DE VIDEO BAJO DEMANDA BASADA EN LA ARQUITECTURA CLIENTE-SERVIDOR," Ph.D. dissertation, Escuela Politecnica Nacional, 2012.

[13] T. Soukupová y J. Cech, "Real-time eye blink detection using facial landmarks," in *21st Computer Vision Winter Workshop*, 2016.

[14] "Encuesta de Victimización y Percepción de Inseguridad 2011," Tech. Rep. [En línea]. Disponible: http://www.ecuadorencifras.gob.ec/documentos/web-inec/Victimizacion/Presentacion{_}principales{_}resultados.pdf

[15] F. Calvo, "Anàlisis Y Diseño De Una Red Domòtica para Viviendas Sociales," Ph.D. dissertation, Universidad Austral De Chile, 2014. [En línea]. Disponible: http://cybertesis.uach.cl/tesis/uach/2014/bmfcic169a/doc/bmfcic169a.pdf

[16] Raspberry Colaborators, "Raspberry Pi - Teach, Learn, and Make with Raspberry Pi," 2018. [En línea]. Disponible: https://www.raspberrypi.org/

[17] Google I/O Keynote, "Android Music, Movies, Home Automation," 2011. [En línea]. Disponible: http://www.businessinsider.com/le-io-kelive-googynote-2011-5

[18] Ring, "HD Video Doorbells," 2018. [En línea]. Disponible: https://ring.com/videodoorbells

[19] Vivint Smart Home, "Doorbell Camera," 2018. [En línea]. Disponible: https://www.vivint.com/products/doorbell-camera

[20] Linphone Colaborators, "Linphone and Raspberry Pi." [En línea]. Disponible: https://wiki.linphone.org/xwiki/wiki/public/view/Linphone/LinphoneandRaspberryPi/

[21] WebRTC Colaborators, "WebRTC Home | WebRTC." [En línea]. Disponible: https://webrtc.org/

[22] J. Joskowicz y R. Sotelo, "Medida de la calidad de voz en redes IP," *Memoria de Trabajos de Difusión Científica y Técnica, ISSN 1510-7450, Nº. 5, 2007, pags. 12-23*, 2018. [En línea]. Disponible: http://www.um.edu.uy/{_}upload/{_}investigacion/web{_}investigacion{_}62{_}MedidadelacalidaddevozenredesIP-JJoskowiczyRSotelo-revistaUM.pdf

[23] G. Aguirre, M. Antonio, V. Alfaro, J. Pablo, M. Venegas Cesar, P. Ortega, J. Carlos, G. Hurtado Efrén, G. Gutiérrez, y C. Alberto, "Control de Acceso del Hogar por Reconocimiento Facial," in *Congreso Nacional de Mecatrónica*, 2009. [En línea]. Disponible: http://www.mecamex.net/anterior/cong08/articulos/29.pdf

[24] easydom, "What is domotics," 2018. [En línea]. Disponible: http://www.easydom.com/en/discover/what-is-domotics

[25] S. Z. Li y A. K. Jain, *Handbook of Face recognition*. Springer, 2011. [En línea]. Disponible: https://books.google.com.ec/books?id=KSXwPmoqGWYC{&}dq=face+recognition{&}hl=es{&}source=gbs{_}navlinks{_}s

[26] A. K. Datta, M. Datta, y P. K. Banerjee, *Face detection and recognition : theory and practice*. Chapman and Hall/CRC, November 2, 2015. [En línea]. Disponible: https://books.google.com.ec/books?id=oyfSCgAAQBAJ{&}dq=face+recognition{&}hl=es{&}source=gbs{_}navlinks{_}s

[27] OpenCV Dev Team, "Face Recognition with OpenCV — OpenCV documentation," 2018. [En línea]. Disponible: https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec{_}tutorial.html

[28] OpenCV Collaborators, "OpenCV library," 2018. [En línea]. Disponible: https://opencv.org/

[29] A. Ozdil y M. M. Ozbilen, "A survey on comparison of face recognition algorithms," in *2014 IEEE 8th International Conference on Application of Information and Communication Technologies (AICT)*. IEEE, oct 2014, pp. 1–3. [En línea]. Disponible: http://ieeexplore.ieee.org/document/7035956/

[30] ITU-T, "Recommendation H.323," 2018. [En línea]. Disponible: https://www.itu.int/rec/T-REC-H.323/e

[31] IEFT, "SIP: Session Initiation Protocol," 2018. [En línea]. Disponible: https://www.ietf.org/rfc/rfc3261.txt

[32] ITU-T, "Recomendación H.248.1," 2018. [En línea]. Disponible: https://www.itu.int/rec/T-REC-H.248.1/es

[33] NiclasOlofsson, "minet," 2018. [En línea]. Disponible: https://github.com/NiclasOlofsson/MiNET/blob/master/src/MiNET/MiNET/Net/MCPE%20Protocol%20Documentation.md

[34] R. Pepper, "Sip uri," urlhttps://getvoip.com/library/what-is-a-sip-uri/, 2012.

[35] ITU-T Rec, "Subjective video quality assessment methods for multimedia applications," 2008. [En línea]. Disponible: https://www.itu.int/rec/T-REC-P.910/en

[36] J. L. Martínez, P. Cuenca, F. Delicado, y F. Quiles, "Objective video quality metrics: A performance analysis," in *2006 14th European Signal Processing Conference*, Sept 2006, pp. 1–5.

[37] A. Hore y D. Ziou, "Image quality metrics: Psnr vs. ssim," in *2010 20th International Conference on Pattern Recognition*, Aug 2010, pp. 2366–2369.

[38] Z. Wang, E. P. Simoncelli, y A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, vol. 2, Nov 2003, pp. 1398–1402 Vol.2.

[39] N. Cranley y M. Davis, "Study of the behaviour of video streaming over ieee 802.11b wlan networks," in *2006 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, June 2006, pp. 349–355.

[40] Bryan Clark , "¿Qué es un códec?" 2018. [En línea]. Disponible: https://www.makeuseof.com/tag/all-you-need-to-know-about-video-codecs-containers-and-compression/

[41] P. Cika, D. Kovac, V. Skorpil, y T. Srnec, "Subjective comparison of modern video codecs," in *2017 Progress In Electromagnetics Research Symposium - Spring (PIERS)*, May 2017, pp. 776–779.

[42] M. A. Ansari y I. U. Khan, "Performance analysis and evaluation of proposed algorithm for advance options of h.263 and h.264 video codec," in *2015 International Conference on Recent Developments in Control, Automation and Power Engineering (RDCAPE)*, March 2015, pp. 371–376.

[43] Dennnis Lark, "MJPEG video encoding in gstreamer," 2018. [En línea]. Disponible: https://www.technomancy.org/gstreamer/mjpeg-gstreamer-encoding/

[44] S. Basavaraju, C. R. Geetha, y H. D. GiriPrakash, "A novel method of post processing algorithms for image and vp8 video codec's," in *2013 International Conference on Signal Processing , Image Processing Pattern Recognition*, Feb 2013, pp. 214–219.

[45] SinoLogic, "Cómo medir la calidad de una llamada," 2018. [En línea]. Disponible: https://www.sinologic.net/2016-06/como-medir-la-calidad-de-una-llamada.html

[46] A. B. Thabet y N. B. Amor, "Enhanced smart doorbell system based on face recognition," in *2015 16th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, 2015, pp. 373–377.

[47] I. Deepika M, Hithashree C V y I. V. N, "Design and Implementation of Smart Doorbell using IOT," *International Conference on Emerging Trends in Science & Engineering*, vol. 9359, num. 5, pp. 630–633, 2017.

[48] W. F. Abaya, J. Basa, M. Sy, A. C. Abad, y E. P. Dadios, "Low cost smart security camera with night vision capability using Raspberry Pi and OpenCV," *2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, num. November, 2014.

[49] Y. Gandhi, S. Vasu, M. Katale, K. Gavhane, y A. Shinde, "IOT based Home Automation using Raspberry Pi with Doorbell Security," *International Engineering Research Journal(IRJET), ISSN 2395-1621*, pp. 1–4, 2015.

[50] T. Saraf, K. Shukla, y H. Balkhande, "Automated Door Acess Control System Using Face Recognition," *International Engineering Research Journal(IRJET), Volume: 05 Issue: 04, e-ISSN: 2395-0056*, pp. 3036–3040, 2018.

[51] SkyBell Technologies Inc, "SkyBell WiFi Doorbell," 2018. [En línea]. Disponible: http://www.skybell.com/

[52] Microsoft, "Windows 10 IoT Core Official Website | Developer Resource | Windows IoT," 2018. [En línea]. Disponible: https://developer.microsoft.com/es-es/windows/iot

[53] G. E. S. C. Janneth Liliana Peña Merizalde, "Estudio del modelo de referencia del internet de las cosas (iot), con la implementaciÓn de un prototipo domÓtico," Tesis, Escuela Politécnica Nacional, Quito, nov 2014.

[54] J. Fitzpatrick, "Todo lo que necesita saber sobre cómo comenzar con la raspberry pi," 2018. [En línea]. Disponible: https://www.howtogeek.com/138281/the-htg-guide-to-getting-started-with-raspberry-pi/

[55] R. P. Foundation, "Usb," 2018. [En línea]. Disponible: https://www.raspberrypi.org/documentation/hardware/raspberrypi/usb/README.md

[56] ioBridge, "ThingSpeak," 2018. [En línea]. Disponible: https://thingspeak.com/

[57] SPACEBREW, "What is Spacebrew?" 2018. [En línea]. Disponible: http://docs.spacebrew.cc/

[58] Firebase, "Firebase Crashlytics," 2018. [En línea]. Disponible: https://firebase.google.com/

[59] Asterisk, *Getting Started with Asterisk*, Ene 2017.

[60] I. Digium, "Asterisk introduction," 2018. [En línea]. Disponible: https://www.voip-info.org/asterisk-introduction/

[61] Google, "Csipsimple," 2018. [En línea]. Disponible: https://en.wikipedia.org/wiki/CSipSimple

[62] S. Arteaga, "El reconocimiento facial de windows 10, engañado por una foto," 2018. [En línea]. Disponible: https://computerhoy.com/noticias/software/reconocimiento-facial-windows-10-enganado-foto-73299

[63] M. Segovia, "El reconocimiento facial del iphone x falla con gemelos o familiares de usuarios," 2018. [En línea]. Disponible: https://www.pulzo.com/tecnologia/fallas-reconocimiento-facial-iphone-x-PP382226

[64] V. Kazemi y J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 1867–1874.

[65] J. H. Joe Minichino, *Learning OpenCV 3 Computer Vision with Python*, 3ra ed. O'Reilly Media, 2015, joe Minichino, Joseph Howse.

[66] SD Association, "SD Memory Card Formatter," 2018. [En línea]. Disponible: https://www.sdcard.org/downloads/formatter{_}4/