



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS

Facultad de Ciencias Químicas e Ingeniería

**Armado y desarme de sistema de acceso empleando
reconocimiento facial**

TESIS PARA OBTENER EL GRADO DE:

Ingeniería Eléctrico-Electrónico

Jose Emmanuel Lagunas Buenos Aires

DIRECTOR: Dr. José Gerardo Vera Dinas

NOMBRE DE LOS SINODALES: Dr. Outmane Oubram, Dr. Luis Cisneros Villalobos, Dra. Alina Martínez Oropeza y Mtro. Francisco Javier Becerra Gonzales

CUERNAVACA, MORELOS

Marzo, 2023.

RESUMEN

En este trabajo se creó el prototipo de un sistema de apertura y cierre empleando reconocimiento facial.

La metodología empleada consistió en la recopilación de información, análisis de requerimientos, tanto de la parte física como del software requerido, así como los aspectos a tener en cuenta para un funcionamiento ideal del sistema, realizando sus respectivas pruebas de funcionamiento y la eficacia del sistema al reconocer a una persona con diferente indumentaria cubriendo su rostro.

Con este sistema no solo se pretende brindar seguridad, sino también incentivar el uso de herramientas tecnológicas como el reconocimiento facial, y dispositivos que están en el mercado, para implementar un control de acceso más seguro y económico.

Palabras clave: Visión artificial, acceso, inteligencia artificial, Raspberry Pi.

ABSTRACT

In this work, the prototype of an entry system that opens and closes using facial recognition was created.

The methodology used consisted in the collection of information, analysis of requirements, of both, the physical part and the software, as well as the aspects to be taken into account for an ideal functioning of the system, carrying out their respective performance tests and the effectiveness of the system when recognize a person with different hats and fase masks.

With this system, it is not only intended to provide security, but also to encourage the use of technological tools such as facial recognition, and devices that are on the market to implement a more secure and economical access control.

Keywords: Artificial vision, access, artificial intelligence, Raspberry Pi.

AGRADECIMIENTOS

Les agradezco a mis padres Laura y Rene, a mi hermana Monserrat y a mi perro Spike por apoyarme a lo largo de carrera universitaria, por ser mi fortaleza y motivo para poder seguir adelante.

Al poder superior por siempre ayudarme.

Le doy las gracias al Dr. Gerardo Vera Dimas p, por haberme asesorado durante el desarrollo de mi tesis, por creer en el proyecto y compartir su conocimiento conmigo, por dedicar su tiempo, su compromiso, apoyo y paciencia, gracias por formar parte fundamental de mi crecimiento académico.

Agradezco a los profesores que pusieron dedicación en sus cátedras, por su esfuerzo y por enseñarme las bases de mi profesión

De igual forma agradezco a la institución.

DEDICATORIA

Le dedico este trabajo a mis padres, por el amor, comprensión y el apoyo incondicional que siempre me han brindado.

Al portador de la luz por estar siempre a mi lado.

ÍNDICE

CAPÍTULO PRIMERO. 14

1. INTRODUCCIÓN. 14

1.1. OBJETIVOS..... 15

1.1.1 OBJETIVO GENERAL 15

1.1.2. OBJETIVOS ESPECÍFICOS..... 15

1.2. IMPACTO POSIBLE. 16

1.3. JUSTIFICACIÓN..... 16

CAPÍTULO SEGUNDO 17

2. MARCO TEÓRICO 17

2.1. RECONOCIMIENTO E IDENTIFICACIÓN FACIAL..... 17

BIOMETRÍA..... 17

2.1.1 INTELIGENCIA ARTIFICIAL..... 18

2.1.2 CÁMARA..... 19

2.1.3 PROCESAMIENTO DE IMÁGENES: LAS IMÁGENES TIENEN ETAPAS DE PROCESAMIENTO, LAS CUALES SE CLASIFICAN COMO SE MUESTRA EN LA FIGURA, DONDE EN PRIMERA INSTANCIA SE HACE LA CAPTURA, DESPUÉS PASA POR UN PROCESAMIENTO PARA LUEGO IDENTIFICAR LOS OBJETOS CAPTADOS..... 19

2.1.4 PLACAS DE DESARROLLO 20

2.1.5 CONTROL DE ACCESO: EL CONTROL DE ACCESO ES UN TÉRMINO MUY USADO EN CUANTO SE REFIERE A LA SEGURIDAD DE UN LUGAR O ESTABLECIMIENTO, ESTE ES UN COMPONENTE MUY IMPORTANTE PORQUE PERMITE MONITOREAR Y CONTROLAR ELECTRÓNICAMENTE EL TRÁFICO DE PERSONAS A TRAVÉS DE OBJETOS COMO PUERTAS, ASCENSORES, TALANQUERAS, ENTRADAS PRINCIPALES, ETC..... 20

2.1.6 MACHINE LEARNING 21

2.1.7 APRENDIZAJE SUPERVISADO 21

2.1.8 APRENDIZAJE NO SUPERVISADO	22
2.1.9 APRENDIZAJE POR ESFUERZO	22
2.2 FASES DE RECONOCIMIENTO FACIAL	22
2.2.1 DETECCIÓN DEL ROSTRO	22
2.2.1.1 Plantilla.....	23
2.2.1.2 Bootstrap	23
2.2.1.3 AdaBoost	23
2.2.1.4 Rotación	24
2.2.1.5 Escalado.....	24
2.2.1.6 Recorte:.....	24
2.2.1.7 Ecuación del histograma:.....	24
2.2.3 RECONOCIMIENTO.....	27
<u>2.3 TECNOLOGÍAS ACTUALES DE RECONOCIMIENTO FACIAL.....</u>	29
2.3.1 DEEP VISION AI	29
2.3.2 SENSE TIME	30
2.3.3 AMAZON REKOGNITION	31
2.3.4 FACE++	31
2.3.5 GOOGLE CLOUD VISION.....	32
2.3.6 OPEN CV.....	32
2.4 HARDWARE	34
2.4.1 PLACAS DE DESARROLLO	34
2.5 SOFTWARE.....	34
2.5.1 RASPBAN.....	34
2.5.2 IDE PYTHON (THONNY).....	35
<u>CAPÍTULO TERCERO.....</u>	35
<u>3. METODOLOGÍA Y PRUEBAS</u>	35
3.1. EL PROYECTO CONSTARA DE CUATRO FASES:.....	35

3.1.1 RECOPIACIÓN DE LA INFORMACIÓN:.....	35
3.1.2 DEFINICIÓN DE LOS REQUERIMIENTOS:	35
3.1.3 SELECCIÓN DE LA TÉCNICA APROPIADA:	35
3.1.4 IMPLEMENTACIÓN DE PROTOTIPO:.....	36
3.2 PRUEBAS Y PROCEDIMIENTO.....	36
3.2.1 RECOPIACIÓN DE LA INFORMACIÓN.....	36
3.2.1.1 Características de la cámara:	37
3.2.1.2 Etnia:.....	38
3.2.1.3 Características individuales.....	38
3.2.1.4 Otros factores:	38
3.2.1.5 La variación de iluminación	38
3.2.2 DEFINICIÓN DE LOS REQUERIMIENTOS.....	39
3.2.2.1 REQUERIMIENTOS DE LA UNIDAD DE PROCESAMIENTO	39
3.2.3 SELECCIÓN DE LA TÉCNICA APROPIADA	42
3.2.4 IMPLEMENTACIÓN DE PROTOTIPO.....	44
PRUEBAS.....	49
CONFIGURACIÓN E INSTALACIÓN DE ENTORNO DE TRABAJO.....	49
CAPTURA DE ROSTRO.....	51
PRUEBAS DE EFICACIA	52
DISTINTOS TIPOS DE ILUMINACIÓN	58
<u>CAPÍTULO CUARTO.....</u>	<u>62</u>
<u>4. RESULTADOS FINALES.....</u>	<u>62</u>
<u>CAPITULO QUINTO.</u>	<u>65</u>
<u>5.1 CONCLUSIONES.</u>	<u>65</u>
<u>5.2. RECOMENDACIONES.</u>	<u>66</u>
<u>BIBLIOGRAFÍA.....</u>	<u>66</u>

Índice de figuras

<i>Figura 1. Raspberry Pi 4 Diagrama</i>	<i>48</i>
<i>Figura 2. SD Card Formatter</i>	<i>49</i>
<i>Figura 3. Raspberry Pi instalación.....</i>	<i>49</i>
<i>Figura 4. Advanced IP Scanner.....</i>	<i>49</i>
<i>Figura 5. PuTTY.....</i>	<i>50</i>
<i>Figura 6. VNC Viewer.....</i>	<i>50</i>
<i>Figura 7. Raspberry Pi 4, Terminal.....</i>	<i>50</i>
<i>Figura 8. Línea de comando haarscascade.....</i>	<i>51</i>
<i>Figura 9. Captura de rostro de usuario.....</i>	<i>52</i>
<i>Figura 10. Carpetas de usuarios</i>	<i>52</i>
<i>Figura 11. Usuario.....</i>	<i>53</i>
<i>Figura 12. Usuario con gorra.....</i>	<i>53</i>
<i>Figura 13. Usuario con cubrebocas.....</i>	<i>53</i>
<i>Figura 14. Usuario con lentes de sol.....</i>	<i>54</i>
<i>Figura 15. Usuario con lentes.....</i>	<i>54</i>
<i>Figura 16. Usuario con cubrebocas tricapa.....</i>	<i>54</i>
<i>Figura 17. Usuario cámara de mala calidad.....</i>	<i>55</i>
<i>Figura 18. Usuario cámara de mala calidad rostro inclinado.....</i>	<i>55</i>
<i>Figura 19. Usuario luz de noche y rostro ángulo contrapicado.....</i>	<i>55</i>
<i>Figura 20. Usuario rostro de lado.....</i>	<i>56</i>
<i>Figura 21. Usuario no registrado.....</i>	<i>56</i>
<i>Figura 22. Usuario no registrado segunda prueba</i>	<i>56</i>
<i>Figura 23. Usuario contrapicado</i>	<i>57</i>
<i>Figura 24. Usuario perfil.....</i>	<i>57</i>
<i>Figura 25. Usuario rostro 3/4.....</i>	<i>57</i>
<i>Figura 26. Usuario rostro en casi ¾</i>	<i>58</i>
<i>Figura 27. Usuario en ángulo picado.....</i>	<i>58</i>
<i>Figura 28. Usuario de noche, buena iluminación</i>	<i>59</i>
<i>Figura 29. Usuario noche, mala iluminación.....</i>	<i>59</i>
<i>Figura 30. Usuario noche, mala iluminación 2.....</i>	<i>59</i>

<i>Figura 31. Usuario noche, mala iluminación 3.....</i>	<i>60</i>
<i>Figura 32. Usuario noche, mínima iluminación requerida.....</i>	<i>60</i>
<i>Figura 33. Software ya no reconoce rostro.....</i>	<i>60</i>
<i>Figura 34. Usuario iluminado por luz de cámara</i>	<i>61</i>
<i>Figura 35. Diagrama de conexiones del sistema.....</i>	<i>61</i>
<i>Figura 36. El sistema abriendo y cerrando dependiendo del rostro que se reconozca.....</i>	<i>62</i>
<i>Figura 37. Sistema conectado y energizado</i>	<i>63</i>
<i>Figura 38. Solenoide cerrado</i>	<i>63</i>
<i>Figura 39. Solenoide abierto</i>	<i>64</i>

Índice de anexos.

<i>Anexo 1. Modulo Rele</i>	<i>69</i>
<i>Anexo 2. Porta pilas</i>	<i>69</i>
<i>Anexo 3. Jumpers</i>	<i>69</i>
<i>Anexo 4. Raspberry Pi 4.....</i>	<i>70</i>
<i>Anexo 5. Fuente de alimentación para Raspberry Pi 4</i>	<i>70</i>
<i>Anexo 6. Batería de litio 18650</i>	<i>70</i>
<i>Anexo 7. Cámara web.....</i>	<i>71</i>
<i>Anexo 8. Módulo de cámara para Raspberry Pi.....</i>	<i>71</i>
<i>Anexo 9. Diagrama de código para captura de rostro</i>	<i>72</i>
<i>Anexo 10. Diagrama modelo de entrenamiento</i>	<i>73</i>
<i>Anexo 11. Diagrama de modelo de entrenamiento, primeros dos bloques</i>	<i>74</i>
<i>Anexo 12. Diagrama de modelo entrenamiento, segundos bloques.</i>	<i>75</i>
<i>Anexo 13. Diagrama reconocimiento facial y candado.....</i>	<i>76</i>
<i>Anexo 14. Diagrama modelo de entrenamiento, desglose de importación de paquetes.....</i>	<i>77</i>
<i>Anexo 15. Diagrama reconocimiento facial, desglose de bloques de rasgos faciales.</i>	<i>78</i>
<i>Anexo 16. Diagrama reconocimiento facial, bucle de reconocimiento facial.</i>	<i>79</i>
<i>Anexo 17 Reconocimiento facial, desglose de solenoide.....</i>	<i>80</i>
<i>Anexo 18. Código de captura de rostro</i>	<i>81</i>
<i>Anexo 19. Código de entrenamiento de reconocimiento de rostros</i>	<i>82</i>
<i>Anexo 20. Código de apertura y cierre con reconocimiento facial.....</i>	<i>84</i>

Índice de tablas

<i>Tabla 1. Resultados de pruebas.....</i>	<i>64</i>
--	-----------

CAPÍTULO PRIMERO.

1. INTRODUCCIÓN.

En la actualidad existen diversas tecnologías que ayudan a las personas a mejorar tanto calidad de vida, como tiempos y/o recursos en cierta tarea específica. Estas tecnologías se encuentran en entornos industriales y de investigación, muy apenas este desarrollo tecnológico se está implementando en las viviendas, se busca que la implementación de estos sistemas brinde cierto tipo de confort y seguridad dentro del hogar. Para esto se aplican ciertas técnicas orientadas a automatización en una vivienda, que integran la tecnología en los sistemas de seguridad, bienestar y comunicaciones. Esto se conoce comúnmente como Domótica, la cual pretende realizar un ahorro energético y una mejor accesibilidad para el propietario.

Con el desarrollo tecnológico dentro del hogar se puede referir a aquellas aplicaciones y servicios que permiten mejorar la calidad de vida de los usuarios al aportar soluciones que facilitan la realización de tareas domésticas rutinarias, que suponen una comodidad añadida y que simultáneamente optimizan el consumo energético, seguridad y facilidad de uso.

Una de las características más relevantes de la automatización en el hogar es la seguridad, un factor bastante importante, el cual brinda un tipo confort que incluye la satisfacción del usuario cuando entra y sale de su hogar, causando la sensación de seguridad al saber que sus puertas principales y demás objetos tecnológicos están dentro de un sistema seguro dado por el acceso al reconocimiento facial.

De acuerdo a lo anterior, este proyecto se realizará en 4 etapas:

Primera parte se explica que tipos de tecnologías hay para un control de acceso, y los principios de uso del reconocimiento facial.

La segunda parte, para entender un poco más el aspecto donde se llevará a cabo el control de acceso, se investigaron algunos de los requerimientos que debe tener un control de acceso, el cual implemente el uso de procesamiento de imágenes.

La tercera parte hace una selección de los aspectos y herramientas de hardware y de software ideales que cumplen cada uno de los requerimientos previstos, para poder implementarlo en el hogar.

Y, por último, se implementará el sistema teniendo en cuenta las pautas y requerimientos escogidos anteriormente para el uso y funcionamiento ideal, cabe aclarar que los dispositivos utilizados se encuentran en el mercado y las herramientas de software son Open Source, algunos de estos objetos son la placa de desarrollo Raspberry Pi 4, una cámara digital en cuanto al software se usó Linux distribución debían (Raspbian) para Raspberry Pi, Python como lenguaje de programación junto con Open CV para la manipulación de las imágenes.

1.1. Objetivos.

1.1.1 Objetivo General

Crear un sistema de apertura activado o desactivado por reconocimiento facial, utilizando el algoritmo Haar cascade debido a su bajo coste de poder computacional, además de permitir que se revisen las imágenes en tiempo real, todo esto implementando en una Raspberry Pi 4 con el sistema operativo Raspberry Pi OS 5.15..

1.1.2. Objetivos Específicos.

1. 1. 2. 1. Recopilar e implementar técnicas de reconocimiento facial
1. 1. 2. 2. Implementar un prototipo de prueba.

1.2. Impacto posible.

Ayudar a reducir el error humano y la intrusión por clonación de radiofrecuencia en sistemas de alarma.

1.3. Justificación.

La cuarta revolución de la tecnología exige nuevas competencias en el desarrollo de sistemas inteligentes y automáticos tanto en la vida real como en la virtual, sobre todo con la aparición de nuevas tecnologías en el proceso de sistemas capaces de simular al ser humano.

Por esta razón, la elaboración de este proyecto permitirá favorecer el estado de confort y seguridad de una persona integrando en un sistema de control de acceso para el hogar, un método inteligente como lo es la visión artificial, ya que es un proceso bastante eficiente para el reconocimiento de los propietarios, dado su bajo índice de error para disminuir la inseguridad de hurto en las viviendas.

CAPÍTULO SEGUNDO

2. MARCO TEÓRICO

2.1. Reconocimiento e identificación facial

Dado que el proyecto se centra en la implementación de un sistema de control de acceso a través de visión artificial, resulta fundamental definir algunos conceptos básicos aplicables que están dirigidos hacia el proyecto.

Biometría.

La biometría es la aplicación de técnicas estadísticas y matemáticas sobre las características físicas estáticas de los individuos para determinar o verificar su autenticidad.

Cuando se habla de una persona es importante distinguir entre características físicas estáticas o dinámicas. Estas últimas incluyen factores de comportamiento, como bien pueden ser una firma, la forma de teclear o la forma de caminar. Por su propia naturaleza, son fáciles de suplantar, ya que no se tratan de características intrínsecas al cuerpo humano, propias de su fisiología. Es por este motivo que cuando se habla de biometría se utilizan las características físicas estáticas, aquellas únicas, inamovibles y propias de cada ser vivo.

En función de la característica que se evalúa, se pueden distinguir diferentes métodos o procesos de autenticación biométricos:

- Reconocimiento de iris
- Reconocimiento facial

- Reconocimiento vascular (geometría de las venas del dedo)
- Reconocimiento por huella dactilar
- Reconocimiento de retina
- Reconocimiento de la geometría de la mano

Es importante mencionar que no todos los métodos presentan el mismo grado de fiabilidad, algunos son mejores que otros.

Aclarando eso, es imprescindible denotar que todo sistema de reconocimiento biométrico debe poseer, por lo menos, cuatro propiedades referidas a la característica física propia de cada método:

- Mensurable: Se puede medir.
- Única: Debe ser una característica única en cada persona.
- Universal: Todos los individuos deben tenerla.
- Estática: Dicha característica no experimenta cambios notables a lo largo del tiempo.

También existen otras propiedades que no son comunes a todos los métodos, como por ejemplo puede ser el contacto físico para medir la característica; en el caso del reconocimiento facial o el reconocimiento a través del iris no hay contacto, mientras que en el reconocimiento vascular o en el reconocimiento por huella dactilar sí existe contacto.

Por lo que respecta a su uso, actualmente la biometría es ya utilizada a nivel global y presenta una gran cantidad de beneficios para el sector doméstico, comercial y corporativo, como pueden ser: permite realizar trámites de forma remota, es fácil de usar y no precisa de recordar claves o patrones, es accesible para todos, se puede combinar con otras tecnologías, permite un control de presencia, aumenta la seguridad del control de acceso a lugares críticos, etc. (Domingo Jaramillo, 2021).

2.1.1 Inteligencia artificial: La IA es la capacidad de las máquinas para usar algoritmos, aprender de los datos y utilizar lo aprendido en la toma de decisiones tal y como lo haría un ser humano. Sin embargo, a diferencia de las personas, los

dispositivos basados en IA no necesitan descansar y pueden analizar grandes volúmenes de información a la vez. Asimismo, la proporción de errores es significativamente menor en las máquinas que realizan las mismas tareas que sus contrapartes humanas. (Rouhiainen Lasse, 2018).

2.1.2 Cámara: La cámara de vídeo, videocámara o cámara filmadora es un dispositivo que captura imágenes convirtiéndolas en señales eléctricas, en la mayoría de los casos a señal de vídeo. (Castaño Saavedra y Alonso Sierra, 2019)

2.1.3 Procesamiento de imágenes: Las imágenes tienen etapas de procesamiento, las cuales se clasifican como se muestra en la figura, donde en primera instancia se hace la captura, después pasa por un procesamiento para luego identificar los objetos captados.

La primera etapa llamada captura, es donde la cámara capta el objeto, donde depende de distintas variables como lo es la distancia del objeto o persona, la resolución de la cámara (Mpx) y las propiedades que se aplican a la cámara para esta captura de imagen.

El preprocesamiento visualiza y procesa todo el entorno de la imagen que no es de interés, como lo son las interferencias de otras personas, objetos, entre otros, que no son de interés para el procesamiento de la imagen.

La segmentación se encarga de fraccionar, reconocer y extraer las características, objetos o interferencias que tiene la imagen en el momento que se realiza la captura.

En la extracción de características, como lo indica, es seleccionar las características que se desean para el procesamiento de la imagen, ya sea para un software o un algoritmo.

Al momento de identificar el objeto o la persona de la que se quiere extraer las características, se debe analizar mediante un algoritmo que tenga características de decisión para que tome el camino correcto para seleccionar el objeto o la persona a procesar, para luego reconocerla, este algoritmo debe estar previamente definido. (Cadena Moreano et al, 2017)

2.1.4 Placas de desarrollo: En general, las placas de desarrollo son dispositivos compuestos por circuitos impresos que funciona mediante un microprocesador, el cual posee una lógica específica, sea cual sea el modelo, la cual permite a desarrolladores, ingenieros y demás comunidad tener la capacidad de programar a un bajo nivel un dispositivo para el cualquier uso electrónico haciendo uso de entradas (Inputs) y salidas (Outputs). Son ampliamente utilizadas para la realización de proyectos emprendedores y de innovación debido a sus bajos costos y de fácil acceso.

Existe bastantes tipos de placas de desarrollo y entre ellas se distinguen algunas características que la hacen ideales para algunos proyectos, están diferencias parten de su potencial y función principal, algunas orientadas más a IoT (Internet de las cosas), otros para inteligencia artificial, y otras en poder de procesamiento etc. (Castaño Saavedra y Alonso Sierra, 2019)

2.1.5 Control de acceso: El control de acceso es un término muy usado en cuanto se refiere a la seguridad de un lugar o establecimiento, este es un componente muy importante porque permite monitorear y controlar electrónicamente el tráfico de personas a través de objetos como puertas, ascensores, talanqueras, entradas principales, etc.

El control de acceso se encuentra en la mayoría de industrias y hogares donde se ve un tipo de automatización en el hogar y esto surge a través de la necesidad antigua de resguardar y proteger bien recursos u objetos o inclusive la propia seguridad del usuario. Basándonos en esto se desarrolló una serie de soluciones basándose en la tecnología disponible en el mercado utilizando estándares y arquitecturas abiertas, lo cual posiciona como referencia dentro del sector de seguridad el control de acceso.

- Algunas de las características más esenciales del control de acceso son:
- Restringir el paso a sitios por horarios, puertas y usuarios.
- Controlar flujo de personas por las instalaciones de una Compañía.
- Trazabilidad (Generación de Reportes).

- Registro de tiempos y asistencia.
- Control automático de entradas y salidas de vehículos o personas.
- Control de personas dentro de una edificación.
- Permite un control exacto del personal sin necesidad de presencia de supervisores para el registro de los ingresos y salida de los empleados.
- Protección de activos, reemplazo de llaves.
- Reemplazo de guardas y personal de seguridad. (Castaño Saavedra, 2019)

2.1.6 Machine learning

El *machine learning* o aprendizaje automatizado es un área de la inteligencia artificial que puede definirse como la capacidad que posee una máquina o un software para aprender a partir de una base de datos, con el objetivo de tomar una decisión correcta sin la intervención de una persona. Este proceso se realiza principalmente a través de la programación de algoritmos y funciones matemáticas, aunque también toma una gran importancia el uso de la estadística.

Existen tres tipos de *machine learning*: el aprendizaje supervisado, el aprendizaje no supervisado y el aprendizaje por refuerzo. (Gutierrez Juan et al, 2019)

2.1.7 Aprendizaje supervisado

Este tipo de aprendizaje se basa en proporcionar a la máquina una base de datos con entradas y sus salidas correspondientes, o, dicho de otra forma, se proporciona la relación entre las entradas y las salidas para que aprenda de ella.

Si se quiere entrenar al algoritmo para que reconozca, por ejemplo, a animales, entonces se proporciona una base de datos con imágenes de cada animal, y cada imagen estará etiquetada con el nombre de su respectivo animal. De esta manera, el objetivo será que la máquina sea capaz de determinar qué animal es a partir de su imagen. (Gutierrez Juan et al, 2019)

2.1.8 Aprendizaje no supervisado

Con este aprendizaje el proceso se realiza sobre un conjunto de datos formado solamente por entradas al sistema. No se tiene información sobre las etiquetas de esos datos. El sistema se basa en el reconocimiento de patrones para poder agrupar los datos en función de sus características y así poder clasificarlos. (Gutierrez Juan et al, 2019)

2.1.9 Aprendizaje por esfuerzo

En este el *machine learning* se basa en el aprendizaje a partir del método prueba y error. El algoritmo se retroalimenta constantemente de los resultados obtenidos en las acciones que realiza, y trata de optimizar el resultado en cada iteración. (Gutierrez Juan et al, 2019)

2.2 Fases de reconocimiento facial:

Se puede definir el sistema de reconocimiento facial como una aplicación dirigida por ordenador que identifica automáticamente a una persona en una imagen digital. Esto es posible mediante un análisis de las características faciales del sujeto extraídas de la imagen o de un fotograma clave de una fuente de video.

El reconocimiento facial es posible gracias a tres fases:

1. Detección del rostro.
2. Extracción de características.
3. Reconocimiento

(Castaño Saavedra y Alonso Sierra, 2019)

2.2.1 Detección del rostro

La detección de un rostro se especifica como las áreas de una imagen digital donde se aprecia el rostro de una persona, esto implica que las áreas que no contienen rostro serán descartadas, o más bien, ignorar el fondo de tal manera que se haga una silueta del rostro identificado. Para la detección de rostro existen varios métodos que resuelven este problema, como el uso de plantillas.

Como solo hay que detectar caras (sin identificar sujetos específicamente) se crean patrones para zonas que representan una cara y zonas que no; generalmente la solución para encontrar patrones que no son rostros, es por medio de la modalidad denominada “*bootstrap*”, o el método más conocido como “*EigenFace*”. (Castaño Saavedra y Alonso Sierra, 2019)

2.2.1.1 Plantilla: Esta es una técnica general empleada para detectar objetos de una escena, donde es representado por formas. Para la detección de una cara este debe tomar una característica del rostro como un ojo o una nariz y se utiliza una plantilla parametrizada que representa por medio de una función de energía los cual interpreta como valles y picos, luego esta plantilla interactúa con la imagen alterando algunos parámetros Una vez ubicada la plantilla se ajustan sus parámetros para minimizar la función de energía. (Castaño Saavedra y Alonso Sierra, 2019)

2.2.1.2 Bootstrap: Esta técnica se basa en patrones que son rostros y se puede ver como clasificador, todos los falsos positivos se agregan como patrones de no-rostro y en una siguiente etapa, si aparecen nuevos falsos positivos, estos se siguen agregando como patrones. (Castaño Saavedra y Alonso Sierra, 2019)

2.2.1.3 AdaBoost: Es un meta-algoritmo de aprendizaje automático basado en la idea de que contar con un grupo de expertos para tomar decisiones es mejor que tener uno solo. Al grupo de expertos se le conoce como ensamble y este representa un clasificador fuerte, es decir, un clasificador con una precisión muy buena. Por su parte, a los expertos que conforman el ensamble se les denomina clasificadores débiles, es decir, clasificadores con una precisión menor. También es denominado un algoritmo adaptativo de *machine learning* cuyo nombre es una abreviatura de *adaptive boosting*.

Como segundo paso se habla del pre procesado, esta depende de la información que se obtuvo en la detección. Esta etapa realiza una serie de transformaciones geométricas sobre la imagen, dejándola preparada para la correcta extracción de

característica y se utilizan cuatro fases para normalizar y alinear la imagen. (Castaño Saavedra, y Alonso Sierra, 2019)

2.2.1.4 Rotación: Una de las utilidades de calcular las coordenadas de los ojos, radica en poder determinar el ángulo de giro de una cara en una imagen y compensarlo. Al tener caras sin giro, el proceso de reconocimiento dará mejores resultados. (Castaño Saavedra y Alonso Sierra, 2019)

2.2.1.5 Escalado: Para conseguir que todos los rostros de las imágenes tengan las mismas proporciones, se utiliza la distancia entre los centros de los ojos para conseguir un radio por el cual la imagen debe ser aumentada o reducida. Esto es necesario, puesto que muchas técnicas de reconocimiento requieren que todos los datos de entrada tengan el mismo tamaño (en nuestro caso la matriz de píxeles). (Castaño Saavedra y Alonso Sierra, 2019)

2.2.1.6 Recorte: Una vez la imagen ha sido rotada y escalada, se procede al recorte de la misma para obtener únicamente la región de interés. Se les da a todas las imágenes las mismas dimensiones con el fin de que todas ellas tengan el mismo tamaño para que sea posible la comparación entre ellas. (Castaño Saavedra y Alonso Sierra, 2019)

2.2.1.7 Ecuilización del histograma: Las imágenes pueden presentar variabilidad en la luminosidad y en el contraste, lo que produce que imágenes similares sean muy diferentes respecto al valor de intensidad de sus píxeles. Mediante la ecualización de su histograma, se pretende que las imágenes que tienen la mayor parte de sus valores de intensidad concentrados en una zona reducida del histograma, pasen a extenderse por todo el rango de valores del histograma. Esto resulta en imágenes con mayor contraste y con menor variabilidad lumínica entre ellas. (Castaño Saavedra y Alonso Sierra, 2019)

2.2.2 Extracción de características

La extracción de características se refiere a la obtención de propiedades o parámetros particulares de cada rostro para luego poder ser clasificados, esto puede tomar tres enfoques diferentes (Castaño Saavedra y Alonso Sierra, 2019):

- Enfoque holístico, basándose en la imagen del rostro como un todo.
- Enfoque mediante características locales, dando mayor importancia a las diferentes partes del rostro (geometría facial).
- Enfoque híbrido, basado en la idea de que el sistema de percepción humana combina características locales y globales para el reconocimiento.

Por otro lado, se pueden ver distintos métodos de extracción de características, parecidos a los enfoques anteriormente enumerados.

Modelo de forma activa - *Active Shape Model (ASM)*. Esta característica se centra en rasgos complejos no rígidos, como lo es la apariencia física. El objetivo es localizar los puntos clave que tiene la cara, mediante un modelado estadístico en una imagen, como lo son; la nariz, boca, cejas, labios y ojos. El modelo estadístico facial se construye a partir de un conjunto de entrenamiento que contiene imágenes con puntos de referencia anotados manualmente. Los ASM se clasifican en tres grupos; *Snakes*, *Deformable Templates* y *Point Distribution Model*. (Castaño Saavedra y Alonso Sierra, 2019)

- *Snakes*: Se usan comúnmente para localizar el contorno de la cabeza. Se inicializa una snake en las proximidades de una cara, y posteriormente se va fijando a los contornos que encuentra hasta asumir la forma de la cara. La evolución de la snake se lleva a cabo minimizando una función de energía, que depende de la energía interna (que depende de las propiedades intrínsecas de la snake y define su evolución natural, expandirse o contraerse) y de la energía externa (que contrarresta la interna y hace que la snake se desvíe de su evolución natural para ajustarse a los contornos). Cuando la snake llegue al equilibrio se habrá ajustado a la forma de la cara. (Carbajal Alvarez et al, 2009)

- *Deformable Templates (Plantillas deformables)*: Es el siguiente paso, usando las snakes para encontrar más rasgos faciales además del contorno de la cara. Por

ejemplo, se pueden encontrar los ojos usando para las snakes un mecanismo de deformación que incluye el cálculo de gradientes de una función de energía. (Carbajal Alvarez et al, 2009)

- *Point Distribution Model* (PDM): Este representa las formas faciales como vectores. Este modelo aprende constelaciones permitidas de dar forma a los puntos de los ejemplos de entrenamiento y usar componentes principales para construir lo que se llama una *Point Distribution Model*. (Castaño Saavedra y Alonso Sierra, 2019)

Análisis de bajo nivel (Low Level Analysis). Este se basa en características visuales de bajo nivel, como lo es el color de piel, movimiento, escala de grises y borde. (Castaño Saavedra y Alonso Sierra, 2019)

El análisis a bajo nivel puede dar información ambigua, por ejemplo, si aparecen en la imagen objetos que tengan un color similar al del modelo de color de piel utilizado. Los métodos que vienen a continuación se basan en la geometría de la cara para caracterizar y posteriormente verificar rasgos a fin de evitar dicha ambigüedad. (Carbajal Alvarez et al, 2009)

Búsqueda de rasgos. Hay varias formas de llevar a cabo este algoritmo, siendo una de ellas la siguiente:

- Búsqueda de la parte superior de la cabeza. Se efectúa una hipótesis sobre lo que puede ser una posible línea del pelo en lo alto de la frente. Puede ser difícil si la persona tiene pelo cubriendo zonas de la frente. (Carbajal Alvarez et al, 2009)

- Búsqueda de los ojos. A partir de dicha línea efectúa un barrido hacia abajo tratando de buscar zonas donde la densidad de gris aumente y disminuya bruscamente en el plano horizontal. Dichas zonas corresponden con las pupilas. Falla si el individuo usa gafas, si uno de los ojos no aparece por cualquier motivo, o evidentemente si la imagen está rotada. (Carbajal Alvarez et al, 2009)

- Uso de plantillas flexibles. La distancia entre la línea del pelo y el plano de los ojos se usa como medida de referencia para inicializar una plantilla flexible que cubre el

resto de rasgos, como la nariz y la boca. La plantilla trata entonces de ajustarse a dichos rasgos usando una función de costes basada en bordes. (Carbajal Alvarez et al, 2009)

- Color de piel: Es muy importante el color de piel para cada ser humano, a la misma vez es mucho más rápido el procesamiento de este que de otras características. (Castaño Saavedra y Alonso Sierra, 2019)

- Movimiento: El movimiento es muy importante, sobre todo cuando el uso de secuencia de video está disponible (grabando), ya que así se puede visualizar las siluetas como las partes de la cara y el resto del cuerpo. (Castaño Saavedra y Alonso Sierra, 2019)

- Escala de grises: Este puede ser muy importante por el hecho de que los rasgos faciales como cejas, pupilas y labios aparecen generalmente más oscuros que el resto de las regiones de la cara. (Morcillo Vizueté, 2021)

Este método es capaz de detectar de forma correcta aproximadamente cuatro de cada cinco caras. (Carbajal Alvarez et al, 2009)

2.2.3 Reconocimiento

Esta etapa final está compuesta por dos ítems, los cuales hacen el trabajo de identificación de los patrones enviados por la etapa anterior y la verificación de estos patrones para una comparación de los patrones obtenidos con unos anteriormente establecidos por el sistema.

Este reconocimiento consiste en la clasificación de las características extraídas del rostro detectado, esta clasificación puede ser realizada de manera controlada, lo cual consiste en que un patrón de entrada es identificado como miembro de una clase ya predefinida. Sin embargo, hay una manera que no es controlada y esta es donde el patrón es asignado a una clase que no está predefinida o en otras palabras a una clase desconocida.

Para el reconocimiento, en cada clase hay un sujeto, por lo tanto, al clasificar las características extraídas de la cara del sujeto se está indicando a que sujeto pertenecen esos patrones, para el diseño de clasificadores se pueden distinguir tres aproximaciones basadas en:

1. Concepto de similitud.
2. Aproximación probabilística.
3. Optimización de un criterio de error.

Como primera parte, las aproximaciones basadas en concepto de similitud son bastante breves, ya que son simples e intuitivas, esto es, porque los patrones similares son asignados a una misma clase y para esto se establece una métrica, la cual define la similitud para después clasificarla por medio una plantilla o mínima distancia usando uno o varios prototipos por clase

Un ejemplo de su efectividad es el uso de esta aproximación en la técnica de Eigenfaces original, la cual aplica la regla del vecino más cercano, utilizando como métrica la distancia Euclides, donde cada prototipo es la media de los patrones de entrenamiento.

El enfoque probabilístico se utilizan conceptos de la teoría de la decisión estadística para establecer los bordes de decisión de las diferentes clases, y se asume que estas características que representan a un patrón tienen una función de densidad probabilística ajustada a la clase.

La tercera aproximación se basa en construir los bordes de decisión optimizando algún criterio de error. El objetivo es minimizar el error de clasificación entre la respuesta deseada y la salida del clasificador. Un ejemplo de este tipo de clasificadores son las redes neuronales, las cuales pueden considerarse como sistemas distribuidos y paralelos que consisten de pequeñas unidades de procesamiento masivamente conectadas. Están conformadas por redes de grafos ponderados donde los nodos son las neuronas artificiales y los bordes (con pesos) son las conexiones entre las neuronas de entrada y salida de la red. Para lograr una

clasificación adecuada son entrenadas con un algoritmo de entrenamiento a partir de un conjunto de datos. (Scarel, German Matias, 2010)

2.3 Tecnologías actuales de reconocimiento facial

2.3.1 Deep Vision AI

Deep Vision AI es una empresa de software de visión por computador que destaca en reconocimiento facial. Utiliza tecnología avanzada propia de visión por computadora para comprender imágenes y vídeos automáticamente, convirtiendo el contenido visual en análisis en tiempo real y conocimientos de valor.

Con más de 500 millones de cámaras existentes en todo el mundo en la actualidad, Deep Vision AI permite analizar transmisiones de cámara a través de diferentes módulos de software basados en inteligencia artificial en una única plataforma de «conectar y usar» (plug-and-play), permitiendo a los usuarios tener alertas en tiempo real y una respuesta más rápida. Su software de reconocimiento facial monitorea continuamente las zonas objetivo para identificar individuos, que se comparan con una lista de seguimiento de personas de interés con tasas de alta precisión. El software es independiente de la cámara, se conecta a cualquier infraestructura de cámara existente y se puede implementar en la nube o en el edge.

Las diferentes capacidades de análisis facial son aplicadas a los casos de uso de inteligencia empresarial para ayudar a reconocer a los clientes importantes en tiempo real, cuantificar la frecuencia de los visitantes o mejorar la seguridad general. Otros atributos, como el recuento, la edad y el sexo de las personas, se utilizan para comprender los cambios demográficos a lo largo del tiempo, el comportamiento del cliente y medir los esfuerzos de marketing. Los clientes generalmente combinan las capacidades de reconocimiento facial con el reconocimiento avanzado de vehículos y otras características basadas en inteligencia artificial que ofrece Deep Vision AI. (Morcillo Vizueté, 2020).

2.3.2 Sense Time

SenseTime, desarrollador líder de plataformas tecnológicas, se dedica a crear soluciones para la industria a través de innovaciones en inteligencia artificial y análisis de big data. La tecnología multifuncional de SenseTime se está expandiendo rápidamente y ya incluye reconocimiento facial, reconocimiento de imágenes, análisis de video inteligente, conducción autónoma y reconocimiento de imágenes médicas. SenseTime ha prestado servicios a más de 400 empresas y agencias gubernamentales reconocidas, incluidas Honda, Qualcomm, China Mobile, UnionPay, Huawei, Xiaomi, OPPO, Vivo y Weibo. Entre su software de plataforma se encuentran (Morcillo Vizueté, 2020):

- **SensePortrait-S:** es un servidor de reconocimiento facial estático, que proporciona funciones de detección de rostros desde una fuente de imagen, así como extracción de características, análisis de atributos, comparación de atributos y recuperación de objetivos de una amplia base de datos de imágenes faciales. (Morcillo Vizueté, 2020).
- **SensePortrait-D:** es un servidor de reconocimiento facial dinámico que proporciona funciones de detección de rostros en múltiples transmisiones de videovigilancia, así como seguimiento facial, extracción de funciones y comparación. (Morcillo Vizueté, 2020)
- **SenseFace:** es una plataforma de vigilancia de reconocimiento facial. Basada en la tecnología de reconocimiento facial, utilizando un algoritmo de aprendizaje profundo, SenseFace proporciona soluciones integradas de análisis de vídeo inteligente, que funciona en vigilancia de objetivos, análisis de trayectoria, gestión de población y análisis de datos relevantes, etc. (Morcillo Vizueté, 2020).

2.3.3 Amazon Rekognition

Amazon Rekognition facilita la adición de análisis de imagen y vídeo a sus aplicaciones con tecnología probada, altamente escalable y de aprendizaje profundo, que no requiere experiencia en aprendizaje automático para su uso. Con Amazon Rekognition se puede identificar objetos, personas, texto, escenas y actividades en imágenes y vídeos, además de detectar cualquier contenido inapropiado. Amazon Rekognition también proporciona análisis faciales de alta precisión y capacidades de búsqueda facial que puede usar para detectar, analizar y comparar rostros. Es posible implementar estos recursos en una amplia variedad de casos de uso vinculados con la verificación de usuarios, el conteo de personas y la seguridad pública.

Las características principales de Amazon Rekognition son las etiquetas para identificar todo tipo de objetos, la detección de texto, la moderación de contenido multimedia, la detección y análisis de rostros, la búsqueda y verificación de rostros, el reconocimiento de famosos y el seguimiento del recorrido de personas. (Morcillo Vizuite, 2021).

2.3.4 Face++

Face++ es una plataforma abierta de inteligencia artificial de la empresa china Megvii que ofrece tecnologías de visión por computador que permite agregar fácilmente tecnologías líderes de reconocimiento de imágenes basadas en aprendizaje profundo y de análisis en sus aplicaciones, con interfaces de programación de aplicaciones (API) y kits de desarrollo de software (SDK) simples y potentes.

Esta plataforma utiliza inteligencia artificial y visión por computador en una gran variedad de formas para detectar rostros, analizar 106 puntos de información en el rostro y confirmar la identidad de una persona con un alto grado de precisión. También permite a cualquier desarrollador crear aplicaciones utilizando sus

algoritmos, lo que ha ayudado a convertirla en la plataforma de reconocimiento facial más extensa del mundo, con 300.000 desarrolladores de 150 países que la utilizan. (Morzillo Vizuite, 2020).

2.3.5 Google Cloud Vision

Google ha optado por diferenciar sus servicios de reconocimiento facial en imágenes de los de reconocimiento facial en vídeo. La API Vision de Google Cloud ofrece modelos de aprendizaje automático preparados previamente y muy potentes a través de las API REST y RPC.

Permite asignar etiquetas a imágenes y clasificarlas rápidamente en millones de categorías predefinidas. También detectar objetos y caras, leer texto impreso y manuscrito, y conseguir metadatos de gran valor. Mientras que la API de Video Intelligence cuenta con modelos de aprendizaje automático entrenados previamente que reconocen de forma automática una gran cantidad de objetos, lugares y acciones en vídeos almacenados y en streaming. Resulta muy eficiente para los usos más habituales y mejora con el tiempo a medida que se introducen nuevos conceptos. (Morzillo Vizuite, 2020).

2.3.6 Open CV

OpenCV (Open Source Computer Vision Library) es una biblioteca de software, código libre de visión por computador y de aprendizaje automático (machine learning).

OpenCV se creó para proporcionar una infraestructura común para aplicaciones de visión por computador y para acelerar el uso de la percepción de la máquina en los productos comerciales. Al ser un producto con licencia BSD (Berkeley Software Distribution), OpenCV facilita que las empresas utilicen y modifiquen el código.

La biblioteca tiene más de 2500 algoritmos optimizados, que incluyen un conjunto completo de algoritmos de aprendizaje automático y visión por computador clásicos y de última generación. Estos algoritmos se pueden usar para detectar y reconocer rostros, identificar objetos, clasificar acciones humanas en vídeos, rastrear movimientos de cámara, rastrear objetos en movimiento, extraer modelos 3D de objetos, producir nubes de puntos 3D a partir de cámaras estéreo, buscar imágenes similares de una base de datos de imágenes, eliminar ojos rojos de imágenes tomadas con flash, seguir los movimientos oculares, reconocer paisajes y establecer marcadores para superponerlos con realidad aumentada, etc. Open CV tiene más de 47 mil usuarios y un número estimado de descargas superior a 18 millones. La biblioteca se utiliza ampliamente en empresas, grupos de investigación y organismos gubernamentales.

Junto con empresas bien establecidas como Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota que emplean la biblioteca, hay muchas nuevas empresas como Applied Minds, VideoSurf y Zeitera, que hacen un uso extensivo de Open CV. Los usos desplegados de Open CV abarcan desde unir imágenes de street view, detectar intrusiones en vídeos de vigilancia en Israel, monitorear equipos de minas en China, ayudar a los robots a navegar y recoger objetos en Willow Garage, detectar accidentes por ahogamiento en piscinas en Europa, gestionar arte interactivo en España y Nueva York, revisar las pistas de aterrizaje en busca de escombros en Turquía, inspeccionar las etiquetas de los productos en fábricas de todo el mundo hasta la detección rápida de rostros en Japón.

Tiene interfaces C ++, Python, Java y MATLAB y es compatible con Windows, Linux, Android y Mac OS. OpenCV se inclina principalmente hacia aplicaciones de visión en tiempo real y aprovecha las instrucciones MMX y SSE cuando están disponibles. Actualmente, se están desarrollando activamente interfaces CUDA y OpenCL con todas las funciones. Hay más de 500 algoritmos y aproximadamente 10 veces más funciones que componen o admiten esos algoritmos. OpenCV está escrito de forma

nativa en C ++ y tiene una interfaz con plantilla que funciona a la perfección con los contenedores STL. (Morcillo Vizuetete, 2020).

2.4 Hardware

2.4.1 Placas de desarrollo

En general, las placas de desarrollo son dispositivos compuestos por circuitos impresos que funciona mediante un microprocesador, el cual posee una lógica específica, sea cual sea el modelo, la cual permite a desarrolladores, ingenieros y demás comunidad tener la capacidad de programar a un bajo nivel un dispositivo para el cualquier uso electrónico haciendo uso de entradas (Inputs) y salidas (Outputs). Son ampliamente utilizadas para la realización de proyectos emprendedores y de innovación debido a sus bajos costos y de fácil acceso.

Existe bastantes tipos de placas de desarrollo y entre ellas se distinguen algunas características que la hacen ideales para algunos proyectos, están diferencias parten de su potencial y función principal, algunas orientadas más a IoT (Internet de las cosas), otros para inteligencia artificial, y otras en poder de procesamiento etc. (Castaño Saavedra y Alonso Sierra, 2019).

2.5 Software

2.5.1 Raspbian

Raspbian es un sistema operativo libre y gratuito basado en Debian y optimizado para el hardware de la Raspberry Pi. Contiene unos 35 mil paquetes pre compilados, de forma tal que es fácil instalar el que necesitemos en la Raspberry Pi. El punto más fuerte de Raspbian es que hay una comunidad inmensa a sus espaldas, la cantidad de información sobre tutoriales o solución de problemas es la razón por la cual es tan dinámico trabajar con esta distribución. (Noël Ruault, 2016).

2.5.2 IDE Python (Thonny)

IDE es el acrónimo del término inglés Integrated Development Environment o, lo que es lo mismo, Entorno de Desarrollo Integrado. (Unir, La universidad del internet, 2022).

Thonny es un nuevo IDE de Python para aprender y enseñar programación que puede hacer que la visualización de programas sea una parte natural del flujo de trabajo de los principiantes. Entre sus características destacadas se encuentran diferentes formas de recorrer el código paso a paso, evaluación de expresiones paso a paso, visualización intuitiva de la pila de llamadas y modo para explicar los conceptos de referencias y montón. Apoya la investigación educativa al registrar las acciones del usuario para reproducir o analizar el proceso de programación. Es de uso gratuito y abierto para la extensión. (Aivar Annamaa. 2015).

CAPÍTULO TERCERO.

3. METODOLOGÍA Y PRUEBAS

3.1. El proyecto constara de cuatro fases: Recopilación de la información, definición de los requerimientos, selección de la técnica apropiada e implementación de prototipo.

3.1.1 Recopilación de la información: Se recopiló más información sobre técnicas de reconocimiento facial. Además, se estudiaron los mecanismos de control de acceso en la vivienda.

3.1.2 Definición de los requerimientos: Se definieron los requerimientos de hardware y software necesarios para la adaptación del control de acceso.

3.1.3 Selección de la técnica apropiada: Se realizó una comparación de algunos algoritmos capaces de tomar fotos al usuario y tomar aspectos de ella para compararlas con la base de datos del propietario.

Se realizó una extracción de características de técnicas utilizadas para el reconocimiento de imágenes faciales.

Se escogió la técnica que está más acorde a las condiciones típicas que requiere el sistema y se hizo una profundización teórica de la técnica seleccionada, las condiciones es que sea Open Source, económico y se pueda usar con Raspberry Pi 4.

3.1.4 Implementación de prototipo: Se llevó a cabo una interconexión física entre los sensores y la placa de desarrollo, así como la realización del programa para el reconocimiento facial y las pruebas correspondientes.

3.2 Pruebas y procedimiento

3.2.1 Recopilación de la información

Usuarios del sistema. Los usuarios son una parte fundamental del sistema, ya que estos son los que hacen uso del reconocimiento facial para un control de acceso en el hogar, por este motivo es necesario de cierto modo contar con algunas restricciones, y recomendaciones, las cuales otorguen un mejor manejo del sistema, además de esto poder brindar más seguridad y calidad a la hora de hacer el reconocimiento facial. Para que una persona se vuelva usuario del sistema tiene que estar registrado y con ello tener una serie de fotos para que el sistema en su defecto extraiga las características y reconozca estas fotos o imágenes.

Estas son llamadas imágenes de referencia, y se almacenan en una base de datos, y para que el sistema haga una comparación tiene que tener una foto o imagen de entrada, la cual es llamada imagen de prueba. Para cada usuario se tendrá en cuenta una base de datos.

Esta base de datos agrupará las fotos de un solo usuario para que el sistema no correlacione la información con otro usuario. Para este grupo de fotos, el algoritmo deberá extraer y agrupar las características y almacenar esta información para su

comparación futura. Cabe decir que estas características deben ser muy precisas debido a que, por ejemplo, una mancha o el color de piel son parámetros muy generales para una comparación de una foto de referencia con una foto de prueba.

Debido a que el algoritmo debe extraer las características de la foto y que, según sea el caso de la técnica de reconocimiento facial, serán una o varias características, este proceso demanda procesamiento de máquina y un tiempo prudente, sin embargo, cuando hay varias fotos del usuario, el algoritmo deberá pasar por cada foto, analizarla y extraer las características, lo cual aumentará considerablemente el procesamiento de cada imagen.

Ahora, si se habla del ingreso de una imagen de prueba al sistema, el algoritmo nuevamente deberá analizarla y extraer las características de esta foto y aparte de eso debe comparar las características extraídas con las ya existentes en el sistema y esto compete a la comparación de cada una de las características de las fotos de los usuarios registrados.

Esto indica el gran trabajo y procesamiento que tiene que hacer la técnica de reconocimiento facial, a la hora de su funcionamiento, por esta razón entre más usuarios haya en el sistema tanto el tamaño de la base de datos como el poder del procesador tiene que incrementar.

Teniendo en cuenta lo anterior para cada locación se debe tener en cuenta un número mínimo y máximo de personas registradas en el sistema, como en este proyecto se habla de hogares y no de empresas, multinacionales, ni sitios con gran número de personal. Teniendo esto claro, para los hogares en México se debe estimar un mínimo y un máximo de personas que habitan un solo hogar.

Funcionamiento del sistema según el medio o factores externos. Para el funcionamiento ideal del sistema se debe tener en cuenta algunas características debido a que estas pueden afectar la viabilidad del reconocimiento facial.

3.2.1.1 Características de la cámara: incluso bajo la misma iluminación, la distribución del color de la piel para la misma persona difiere de una cámara a otra

dependiendo de las características del sensor de la cámara. El color reproducido por una cámara CCD “(Charge Coupled Device o, en español, Dispositivo de Carga Acoplada) el cual es un tipo de sensor que es sensible a la luz y trabaja a manera de líneas de píxeles con una cobertura de los colores primarios (RGB)”³⁶ Y estas cámaras dependen de la reflectancia espectral, que prevalece condiciones de iluminación y sensibilidades del sensor de la cámara. (Castaño Saavedra y Alonso Sierra, 2019)

3.2.1.2 Etnia: el color de la piel también varía de una persona a otra, debido a que pertenece a diferentes grupos étnicos y de personas de diferentes regiones. Por ejemplo, el color de la piel de personas pertenecientes a grupos asiáticos, africanos, caucásicos e hispanos; es diferente el uno del otro, por lo tanto, este cambio afecta el reconocimiento facial y el sistema debe ser capaz de adaptarse a dichos cambios. (Carbajal Alvarez et al, 2009)

3.2.1.3 Características individuales: características individuales como la edad, el sexo y las partes del cuerpo también afectan el color de la piel, apariencia. (Carbajal Alvarez et al,2009)

3.2.1.4 Otros factores: diferentes factores como la apariencia del sujeto (maquillaje, peinado y gafas), colores de fondo, Las sombras y el movimiento también influyen en la apariencia del color de la piel. (Castaño Saavedra y Alonso Sierra, 2019)

3.2.1.5 La variación de iluminación: Está una de las características del medio que más afecta al reconocimiento facial, y debido a esta, muchos investigadores y científicos buscan la forma de mitigar el efecto dañino que hace la luz para el reconocimiento facial, debido a que es muy difícil identificar patrones o características específicas cuando las tonalidades de la piel o características cambian de color.

Y aunque de cierta manera es práctico extraer características deseadas de una imagen bajo un entorno controlado, y a esto se refiere a un entorno con luz controlada fondo uniforme o de una sola tonalidad y solo en el caso de

reconocimiento facial por medio de video un entorno donde el usuario esté estático, sin embargo, si hablamos de reconocimiento bajo un entorno no controlado la cara debe ser reconocida bajo diversos efectos como variaciones o cambio en la distribución de la fuente de luz y en el nivel de iluminación (interior, exterior, reflejos, sombras, luces no blancas) produce un cambio en el color de la piel en la imagen esto es llamado problema de constancia de color o sobreexposición y subexposición. La sobreexposición es un efecto dado a la exposición excesiva de luz sobre un material sensible a la luz que en este caso es la cámara digital, este efecto se da o se produce cuando hay demasiada luz en una fotografía. La subexposición produce un efecto de una imagen demasiado oscura, perdiendo detalle en las sombras, que tienen un aspecto negro profundo y carecen de detalle, lo cual afectaría a gran medida la extracción de características para el reconocimiento facial. (Castaño Saavedra y Alonso Sierra, 2019)

3.2.2 Definición de los requerimientos

Los requerimientos necesarios para la implementación del sistema de apertura y cierre empleando reconocimiento facial. Estos requerimientos ayudaron definieron tanto la placa de desarrollo, el software y algoritmo más eficiente para realizar el proyecto.

Algunos de los requerimientos que se necesitan son: unidad de procesamiento, cámara, batería, unidad de almacenamiento, algoritmo, y software.

3.2.2.1 REQUERIMIENTOS DE LA UNIDAD DE PROCESAMIENTO

Algunos de los parámetros o requerimientos que requiere el sistema son los siguientes.

CPU y velocidad de reloj: Esto afecta a gran medida el rendimiento en general de la placa y es la que define que tan veloz es la placa para ejecutar cálculos. (Castaño Saavedra y Alonso Sierra, 2019)

RAM: Esta variable afecta el número de tareas que se pueden ejecutar simultáneamente. También afecta la rapidez los datos se pueden procesar, como el

intercambio de datos de RAM a no volátil, el almacenamiento incurre en un gran rendimiento, gastos generales. (Castaño Saavedra y Alonso Sierra, 2019)

Unidad de procesamiento gráfico: Es preciso que la unidad de procesamiento tenga el rendimiento necesario para poder procesar el video e imágenes que se capturaran. (Pérez Cristian, y Piccolia Fabiana, 2010)

Almacenamiento: El almacenamiento afecta el tamaño de los programas, sistemas operativos y datos generados o descargados que pueden almacenarse en una placa de desarrollo. (Castaño Saavedra y Alonso Sierra, 2019)

Número de pines de I/O para un propósito en general: Se utilizan para conectar componentes a la placa de desarrollo; por lo tanto, más pines típicamente significa más componentes conectados simultáneamente. (Castaño Saavedra y Alonso Sierra, 2019)

OPEN-SOURCE: Tanto el software y el hardware deben ser de libre distribución, permitir trabajos derivados, libre de discriminación a personas, grupos o áreas de iniciativa, la licencia de los mismos no puede ser específica de un producto ni tampoco debe restringir otro software, así mismo la placa de desarrollo deberá contar con trabajos derivados para poder utilizarlos de referencia. (Goñi Oramas, 2012).

Consumo de energía: Este puede jugar un papel importante en las elecciones de diseño. Para proyectos portátiles, o proyectos que necesariamente necesiten una fuente de poder continua. (Castaño Saavedra y Alonso Sierra, 2019)

Requerimientos del algoritmo

Los requerimientos del algoritmo dieron las pautas para escoger la técnica de reconocimiento facial ideal, ya que esta comprende desde la detección del rostro hasta la comparación y clasificación de la foto. Por lo tanto, este análisis se hizo desde enfoques diferentes, debido a que se tiene que identificar las características ideales y no ideales del comportamiento del algoritmo en entornos controlados y no controlados.

Para esto, se hizo una lista de requerimientos esenciales del algoritmo:

- Capacidad de detectar y reconocer un rostro
- La eficiencia del algoritmo con cierto número de usuarios.
- La capacidad del algoritmo al reaccionar al contraste de la luz natural, luz artificial y poca luz.
- Influencia en la variación de aspectos de la cara como lo son accesorios y ángulos diferentes.
- Bajo coste computacional.
- Velocidad de respuesta y ejecución.

El software es donde se realiza las tareas demandadas por el algoritmo. Por esta razón es muy importante saber qué requerimientos tiene el sistema con respecto al software, dado que estos requerimientos de software son características que debe tener el software instalado en el dispositivo. Estos requerimientos pueden ser de sistema operativo o aplicaciones instaladas.

Los requerimientos con respecto al sistema operativo y de las aplicaciones instaladas van enfocados al uso de software de uso libre debido al costo extra que generaría la adquisición de un software en el mercado.

Además, el software debe ser compatible con la placa de desarrollo y con sus requerimientos, por esta razón se debe tener cuidado con que programa, aplicación, sistema operativo y hasta el lenguaje de programación se utiliza en este proyecto.

Algoritmo de viola-jones: Es un método de detección de objetos, se destaca por su bajo costo computacional, lo que permite que sea empleado en tiempo real.

Su desarrollo fue motivado por el problema de la detección de caras, donde sigue siendo utilizado, pero puede aplicarse a otras clases de objetos que, como las caras, estén caracterizados por patrones típicos de iluminación.

Se basa en una serie de clasificadores llamados Haar-like features que se pueden calcular eficientemente a partir de una imagen integral, se agrupan en una cascada empleando un algoritmo de aprendizaje basado en AdaBoost para conseguir un alto

rendimiento en la detección, así como una alta capacidad discriminativa en las primeras etapas.

Existen tres tipos de características de este algoritmo y son representadas en la figura, en la parte izquierda, mientras que en la parte derecha se puede observar algunos ejemplos de características Haar comunes de un detector de caras. Este algoritmo considera regiones rectangulares en una ventana de detección, suma las intensidades de los píxeles en cada región y calcula la diferencia entre estas sumas. (Gutiérrez Juan et al, 2019).

Haar-like features: Son los elementos básicos con los que se realiza la detección. Estos clasificadores son características muy simples que se buscan en las imágenes y que consisten en la diferencia de intensidades luminosas entre regiones rectangulares adyacentes. Las características quedan, por tanto, definidas por unos rectángulos y su posición relativa a la ventana de búsqueda y adquieren un valor numérico resultado de la comparación que evalúan. (Jian et al, 2017).

Eigen Face: es una técnica de reconocimiento facial y una representación de imagen facial, donde los vectores ortonormales se combina de forma lineal. Las imágenes tratadas por esta técnica, se obtienen de los vectores de la matriz de covarianza por medio del conjunto de imágenes almacenadas en una base de datos, donde i_1, i_2, \dots, i_k son un conjunto de imágenes faciales (k), y cada una se ordena lexicográficamente.

Los vectores de la matriz L (son los grandes), estos tienen un subespacio lineal y pueden reconstruir la imagen de la cara con el mínimo error de reconstrucción en el sentido de los mínimos cuadrados. (Castaño Saavedra y Alonso Sierra, 2019)

3.2.3 Selección de la técnica apropiada

Los requerimientos anteriormente mencionados son fundamentales para la realización del proyecto, por esta razón aspectos como el almacenamiento de información haciendo las veces de base de datos. Capacidad de procesamiento, el

cual debe ser esencial para un funcionamiento ideal, adaptación de una cámara digital otorgada por algunas librerías, el ensamblaje de dispositivos a una placa de desarrollo y además de eso, el bajo costo que debe tener el proyecto.

Con esto en cuenta y obteniendo todas las características de cada placa de desarrollo que anteriormente se evidenció, se pasó a escoger la placa de desarrollo, la cual contenga la mayoría de las características de las condiciones previstas para este proyecto. Y con esto se decide que la placa de desarrollo ideal es la Raspberry Pi 4 modelo B, debido a su gran procesamiento y su gran capacidad de memoria RAM, lo cual mejoraría en gran medida el procesamiento, que, aunque no tiene memoria interna, lo cual sería bastante indispensable para el almacenamiento de imágenes, esta placa de desarrollo contiene un apartado para introducir una memoria externa lo cual facilita algunos procesos.

Además de eso, aunque no está dentro de los dispositivos más económicos, su costo es relativamente bajo debido a su capacidad y teniendo en cuenta que es *Open Source*, supone en gran medida utilizar la parte gratuita para no generar dificultades futuras a la hora de utilizarlo para este proyecto. Además de esto, la Raspberry Pi 4 tiene una gran ventaja en cuanto a las otras placas debido a su gran número de librerías y dispositivos que funciona como herramientas junto a esta placa. El caso es para la utilización de una cámara, y esta placa tiene bastante forma de utilizar una cámara digital como medio de entrada de información y según el apartado del módulo de cámara V2 Raspberry Pi 4 es compatible con esta placa.

Con respecto a la cámara, la cual es la entrada de datos al sistema otorgando fotografías de los usuarios para su reconocimiento facial. En este caso se escogió la cámara *Salandens Camara web HD 1080p* cumple con los requerimientos.

Ahora, debido a que se utilizó la placa de desarrollo Raspberry Pi, esta dispone de una distribución Linux basada en Debian, el cual es un sistema operativo. Y para instalarlo se utilizó una distribución de Debían especialmente para Raspberry Pi, la cual es Raspbian.

El lenguaje de programación que se utilizó para el proyecto es Python.

Ahora bien, para realizar el reconocimiento facial se ha utilizado la herramienta Open CV, Esta herramienta es muy útil debido a que es una biblioteca o librería de visión artificial (Rama de la inteligencia artificial), esto quiere decir, visión por ordenador y permite el análisis de imagen y aprendizaje automático, usando una infinidad de algoritmos para identificar y reconocer los objetos que se decidan. Esta biblioteca es multiplataforma, lo que significa que puede ejecutarse en diferentes sistemas operativos como: Windows, iOS, Mac OS, Android y Linux; Y también se puede utilizar en diferentes lenguajes de programación como: Java, Objective C, C#, Python, siendo este último del interés del proyecto. Además de esto, este software es totalmente libre, como sus siglas lo indican “Open CV” (*Open Source Computer Vision*).

3.2.4 Implementación de prototipo

Según lo planteado en la metodología del proyecto, se basa en un desarrollo en cascada, el cual es un modelo lineal donde se plantea una serie de procesos secuenciales para llegar a un objetivo final. Este objetivo final, como bien se sabe, es la implementación de un sistema de reconocimiento facial para el control de acceso en el hogar, el cual tiene especificaciones y requerimientos.

Para el desarrollo de este sistema se necesitaron varias herramientas, las cuales ayudaron en gran proporción a realizarlo. Estas herramientas de software y de hardware tienen que ajustarse a las condiciones establecidas en el apartado de requerimientos, además, de tener en cuentas las condiciones típicas de acceso a la vivienda.

Una de estas herramientas es la de hardware, el cual ya se habló en el apartado anterior y se aclaró lo indispensable del uso de dispositivos que cumplan con los requerimientos mínimos, y para esto se tuvieron en cuenta una lista de dispositivos los cuales cumplen con características diferentes y se concluyó lo siguiente:

- La placa de desarrollo Raspberry Pi 4 B puede emplearse para el sistema a desarrollar.
- La cámara Salandens Camara web HD1080p cumple con los requerimientos.

- Dispositivo encargado de ofrecer la interfaz de usuario, es el computador de escritorio o portátil del usuario.
- Las herramientas de software o desarrollo, son utilizadas para crear programas informáticos que a su vez tienen una finalidad, como por ejemplo una aplicación. Estas herramientas deben poseer características compatibles con la placa ordenadora Raspberry Pi 4 B, por esta razón se concluyó lo siguiente:
- Se usó el sistema operativo Linux distribución Debían, ideal para la Raspberry Pi.
- El lenguaje de programación que se utilizó es Python versión 3.
- Para la realización de los algoritmos se utilizó Open CV versión 3.2.0.
- La técnica que se escogió en este proyecto es *Eigen-faces*, ya que cumple con los requerimientos necesarios.

Diagrama eléctrico.

Se identificó las partes físicas que componen el proyecto y como conectarlas entre sí para el debido funcionamiento. Para esto es necesario saber qué componentes contiene la Raspberry Pi 4.

- Viene con procesador Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) SoC de 64 bits y 1,5 GHz
- 4 GHz y 5,0 GHz IEEE 802.11ac inalámbrico, Bluetooth 5.0, BLE
- Transporta Gigabit Ethernet para conectar computadoras en el espacio físico
- RAM está disponible con diferentes memorias. Elija entre SDRAM LPDDR4-3200 de 2 GB, 4 GB u 8 GB (según el modelo)
- Puerto de pantalla MIPI DSI de 2 carriles
- Puerto de cámara MIPI CSI de 2 carriles
- Un puerto USB 3.0; Un puerto USB 2.0 y un puerto de alimentación USB-C.

- 2 x puertos micro-HDMI que pueden admitir pantallas 4K
- Puerto de audio estéreo y video compuesto de 4 polos
- H264 (descodificación 1080p60, codificación 1080p30), H.265 (descodificación 4kp60)
- Gráficos OpenGL ES 3.0
- Encabezado GPIO de 40 pines estándar de Raspberry Pi (que es compatible con módulos anteriores)
- Contiene una ranura para tarjeta Micro-SD para cargar el sistema operativo y el almacenamiento de datos
- 5 V CC a través del cabezal GPIO (mínimo 3 A*)
- Alimentación a través de Ethernet (PoE) habilitada (requiere PoE HAT separado)
- Temperatura de funcionamiento: 0 – 50 grados C ambiente
- 5V CC mediante conector USB-C (mínimo 3A*)

(Shaw Andrew, 2021)

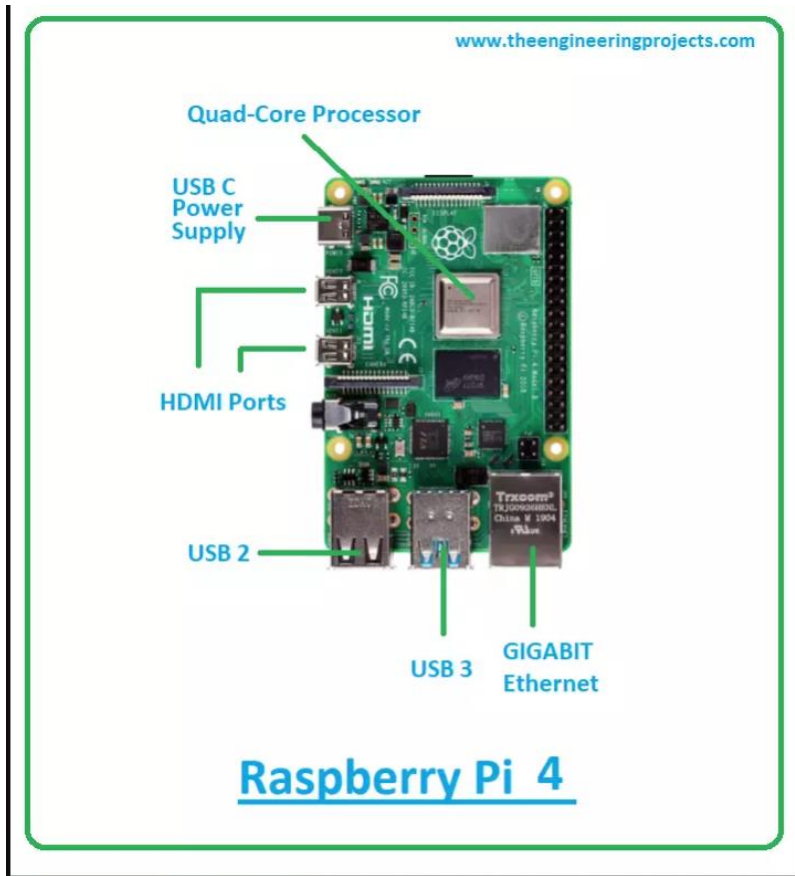
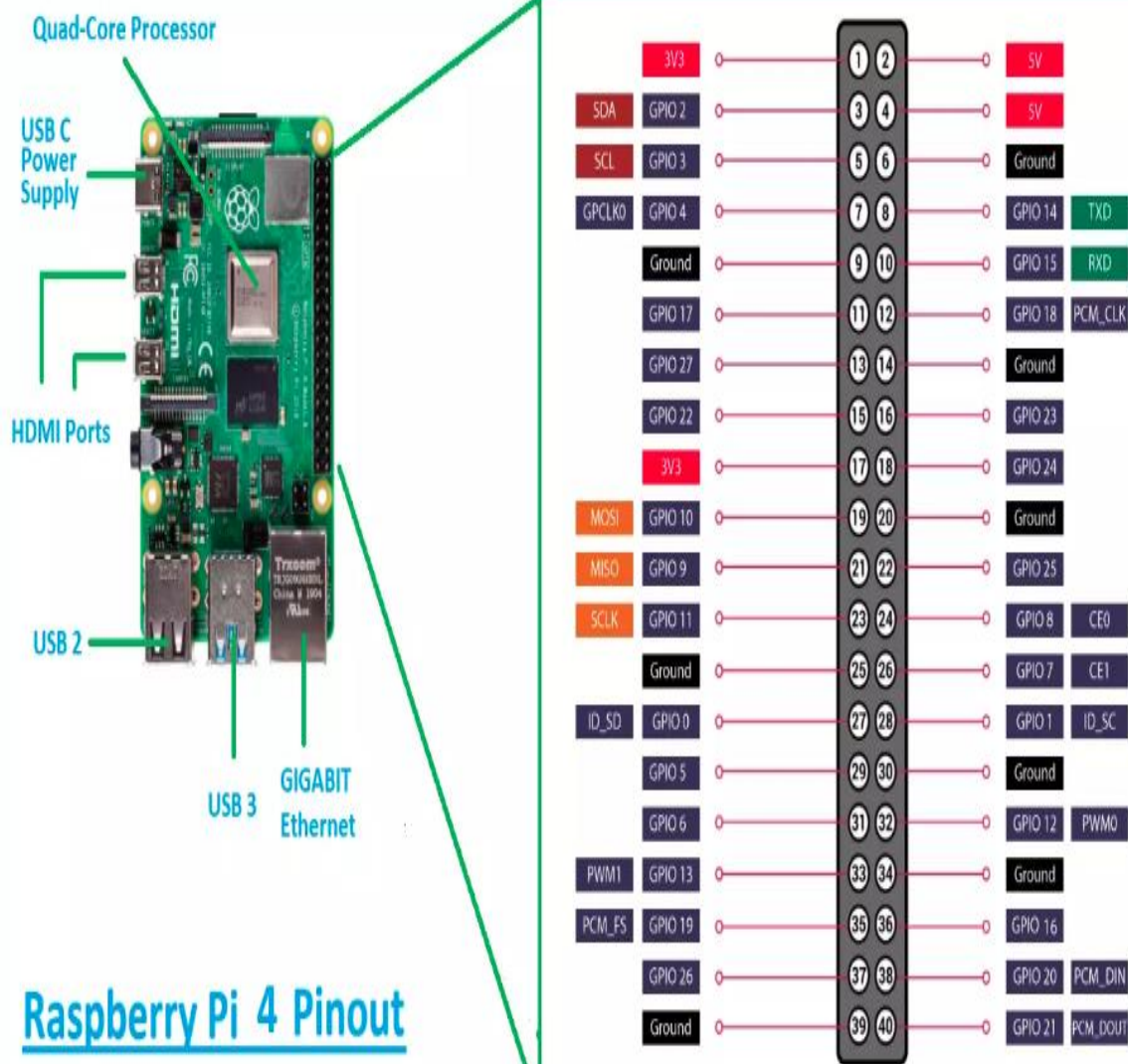


Figura 1. Raspberry Pi 4, extraída de The engineering Project, (Andrew Shaw, 2021)



Raspberry Pi 4 Pinout

Figura 2. Raspberry Pi 4 Diagrama, extraída de The engineering Project, (Andrew Shaw ,2021).

Pruebas

Configuración e instalación de entorno de trabajo.

Lo primero que se realizó fue instalar el entorno de trabajo, se instala Raspbian en una memoria micro SD previamente formateada.

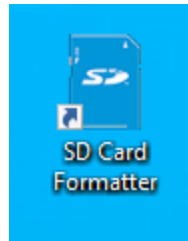


Figura 3. Logotipo de SD Card Formatter, (Elaboración propia con información de Windows 10)



Figura 4. Raspberry Pi pantalla de instalación.

Con Raspbian instalado de forma correcta en la micro SD, esta se inserta en la Raspberry Pi 4. No es necesario contar con un monitor para la placa de desarrollo, esta puede configurarse gracias a *PuTTY*, *Advanced Ip Scanner* y *VNC viewer*.

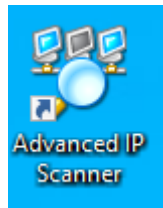


Figura 5. Logotipo de Advanced IP Scanner, (Elaboración propia con información de Windows 10)



Figura 6. Logotipo de PuTTY, (Elaboración propia con información de Windows 10)

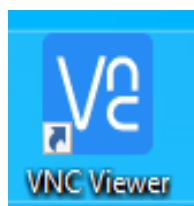


Figura 7. Logotipo de VNC Viewer, (Elaboración propia con información de Windows 10)

Teniendo en cuenta lo anterior y estando en el escritorio de la Raspberry Pi 4 B, se procede a abrir la terminal, ya sea en menú o con Ctrl + Alt + T, para instalar los paquetes necesarios como lo es OpenCV que es una librería bastante completa en cuanto al tratamiento de imágenes, reconocimiento de características faciales, algoritmos de comparación, entre otras, esto se hace por medio de comandos por eso es necesario abrir la terminal del sistema operativo.

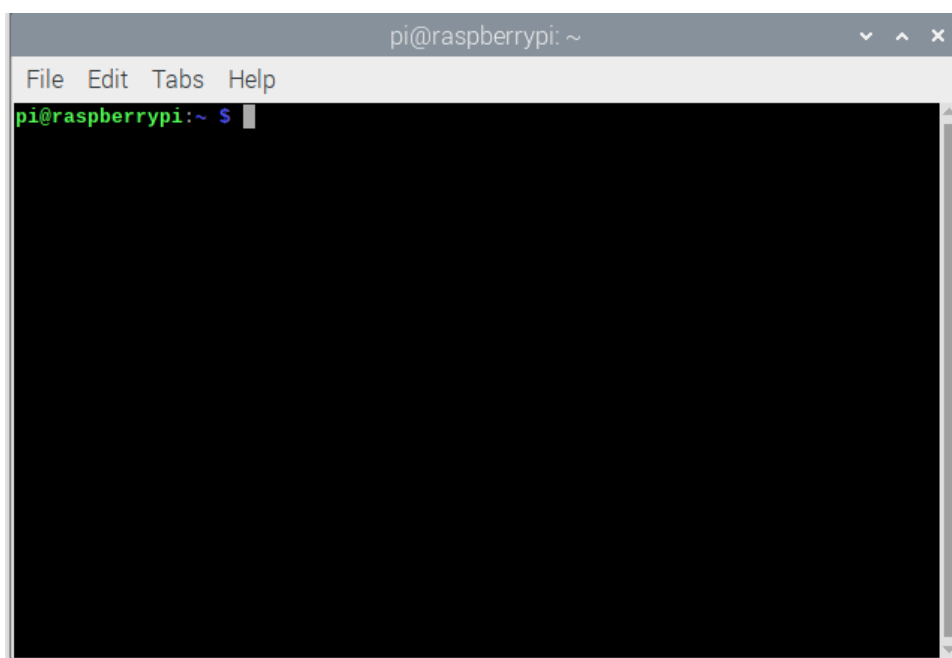


Figura 8. Raspberry Pi 4, Terminal de usuario, (Elaboración propia usando Raspberry Pi OS).

Captura de rostro

Ya instaladas las herramientas necesarias en la Raspberry pi 4 B se procede a diseñar el algoritmo. Lo primero que se debe entender es el cómo se realiza la captura del rostro, que es lo que más importa para este proyecto. Teniendo en cuenta que OpenCV ya se instaló y sabiendo que esta biblioteca trata las imágenes y sus características, se debe hablar del archivo en cascada, `haarcascade_frontalface.xml`.

```
cascade = "haarcascade_frontalface_default.xml"
```

Figura 9. Línea de comando haarcascade, (Elaboración propia usando IDE Thonny Python).

Este documento tiene una gran importancia en todo el proyecto, debido a que es el encargado de obtener las características de una persona en cuanto a rasgos faciales como el mismo archivo lo dice, contiene una base de datos de la parte frontal de la cara de miles de personas y previamente ya fue entrenado con todas estas fotos, lo que realiza es un algoritmo de comparación entre rasgos faciales y así determina los ojos, la boca, la nariz, las cejas y cada característica facial que puede llegar a tener una persona, de esta manera el entrenamiento que realiza es bastante completo, permitiendo una mayor eficacia en la comparación de características faciales.

Después, utilizando un programa de fotografía, nuestra cámara tomará las fotos y permitirá crear una base de datos con el nombre del o de los usuarios.

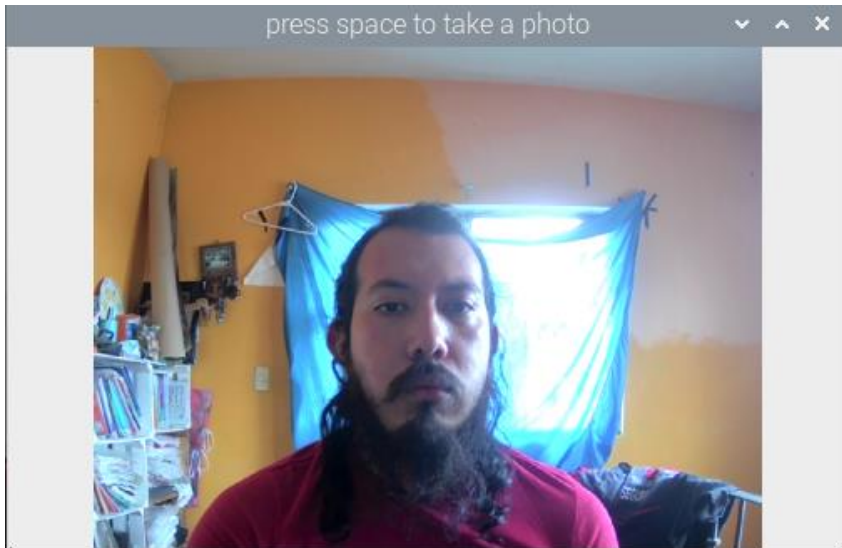


Figura 10. Captura de rostro de usuario

Posterior a esto se entrena el algoritmo con las fotos tomadas por el usuario. Cabe aclarar que este entrenamiento se debe hacer con los rostros de los usuarios que se deseen agregar, así como crear su ruta de acceso y respectiva carpeta de rostros.

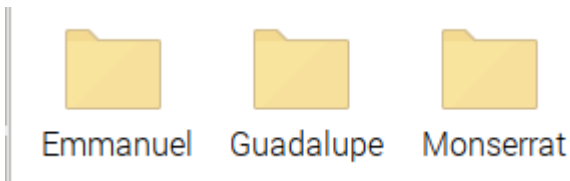


Figura 11. Carpetas de usuarios, (Elaboración propia usando Raspberry Pi OS).

Pruebas de eficacia

Se realizaron pruebas utilizando, diferentes objetos como lentes, cubrebocas y gorras. Así como distintos tipos de cámaras y de usuarios.

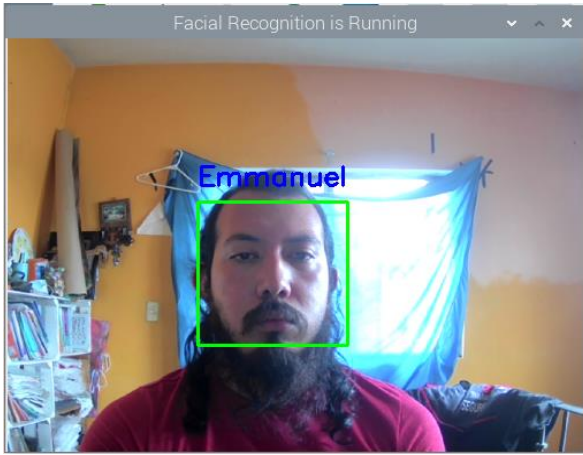


Figura 12. Usuario

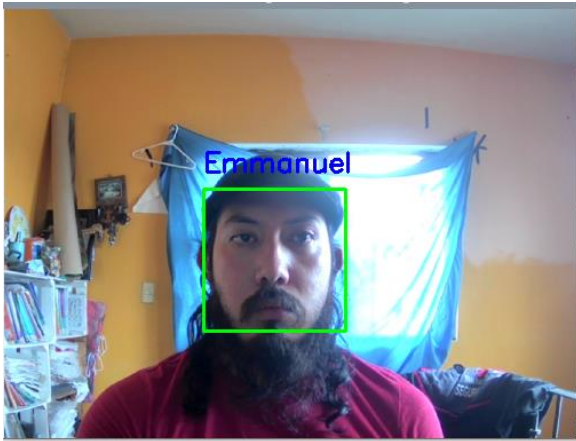


Figura 13. Usuario con gorra



Figura 14. Usuario con cubrebocas

Cuando se realizó la prueba con un cubrebocas KN95, el programa no detecto el rostro.

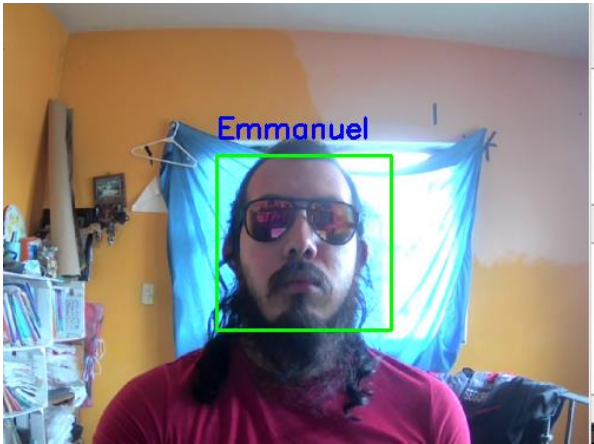


Figura 15. Usuario con lentes de sol

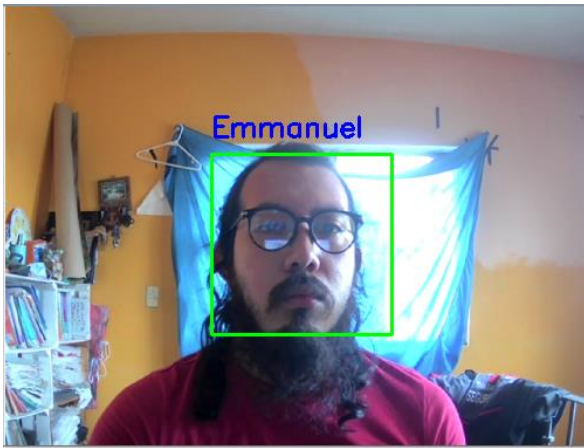


Figura 16. Usuario con lentes



Figura 17. Usuario con cubrebocas tricapa

Se realizan pruebas con una cámara de baja definición, pero debido la resolución de la cámara, el programa no detecto el rostro.



Figura 18. Usuario cámara de mala calidad.



Figura 19. Usuario cámara de mala calidad rostro inclinado

Se realizan pruebas con otro tipo de iluminación y ángulos del rostro.

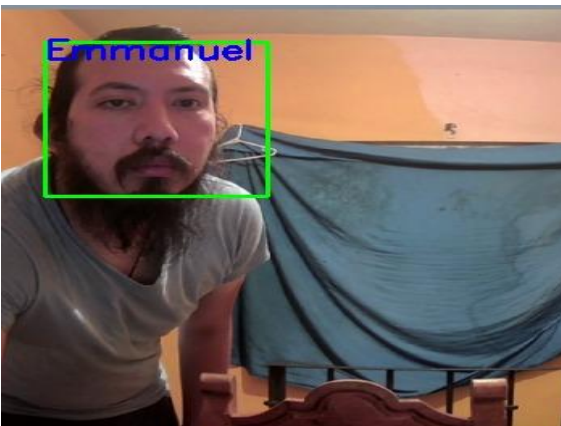


Figura 20. Usuario luz de noche y rostro ángulo contrapicado

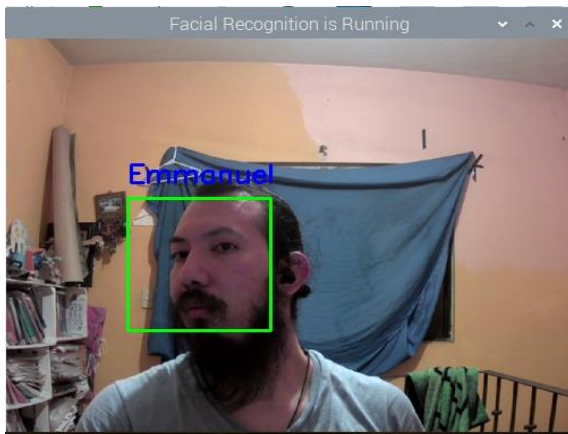


Figura 21. Usuario rostro de lado

También el cómo registra el programa a usuarios no registrados.

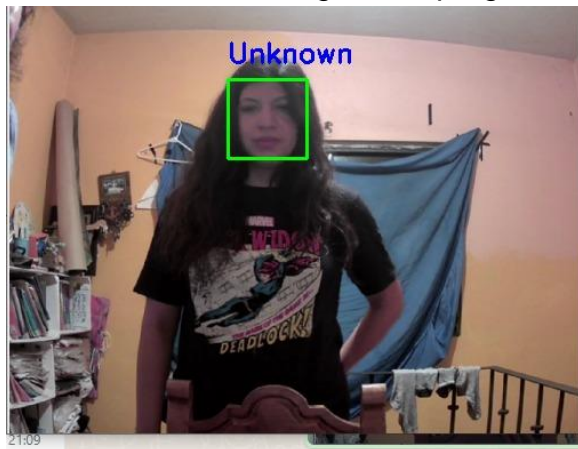


Figura 22. Usuario no registrado

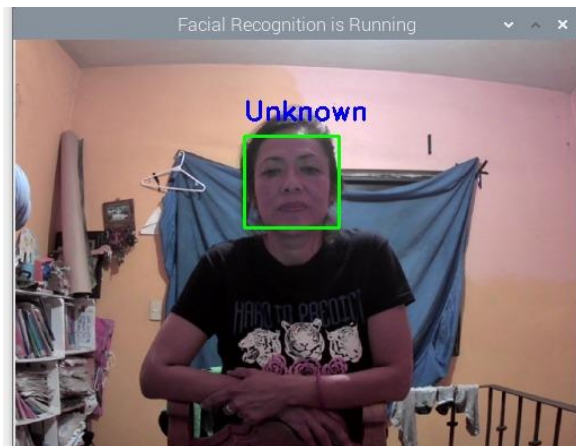


Figura 23. Usuario no registrado segunda prueba

Entre las distintas pruebas se detectaron ocasiones en las que el software no reconoce el rostro, como lo puede ser con el rostro en ángulos de contra picada y

Sin embargo, en ángulos de picada y con el rostro ligeramente de lado, no tiene problemas.

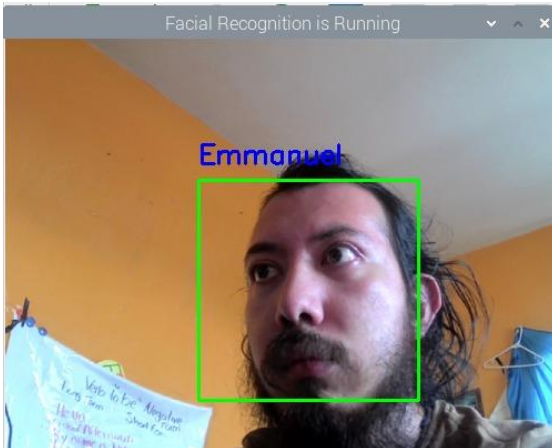


Figura 27. Usuario rostro en casi $\frac{3}{4}$

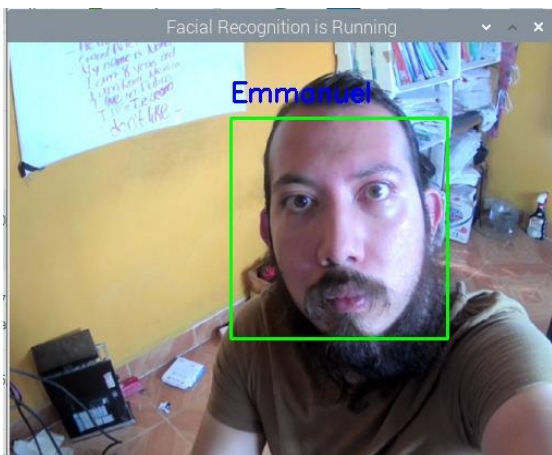


Figura 28. Usuario en ángulo picado

Distintos tipos de iluminación

Se realizan pruebas utilizando distintos de iluminación en la noche, desde una iluminación correcta hasta que está tan oscuro que el software no reconoce más el rostro.

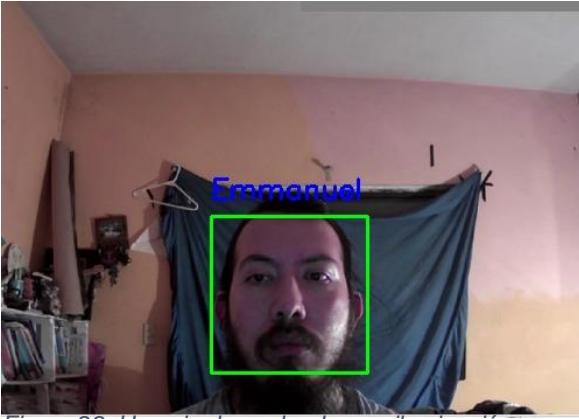


Figura 29. Usuario de noche, buena iluminación



Figura 30. Usuario noche, mala iluminación.



Figura 31. Usuario noche, mala iluminación 2

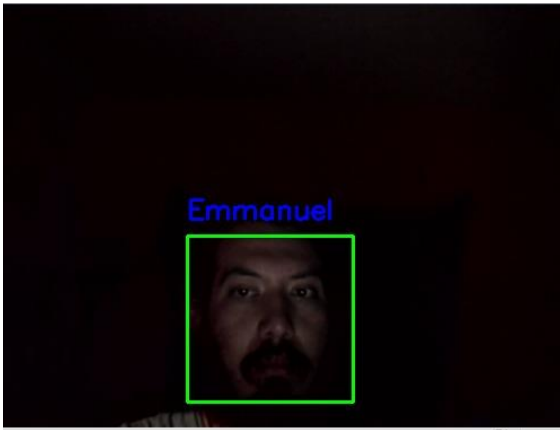


Figura 32. Usuario noche, mala iluminación 3

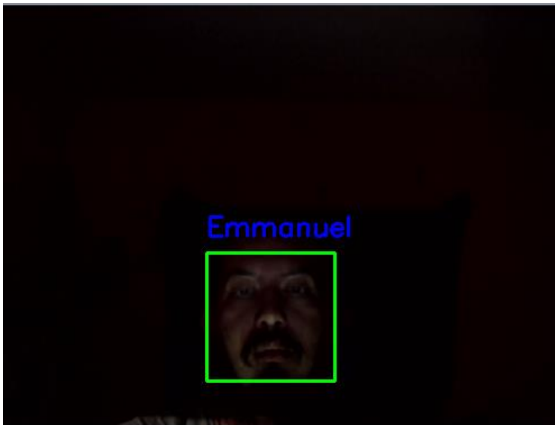


Figura 33. Usuario noche, mínima iluminación requerida

Menor luz mostrada que utilizada en la figura 32 el software ya no reconoce el rostro.



Figura 34. Software ya no reconoce rostro.

Se realiza una prueba con la luz del celular iluminando para verificar que el software reconoce el rostro.

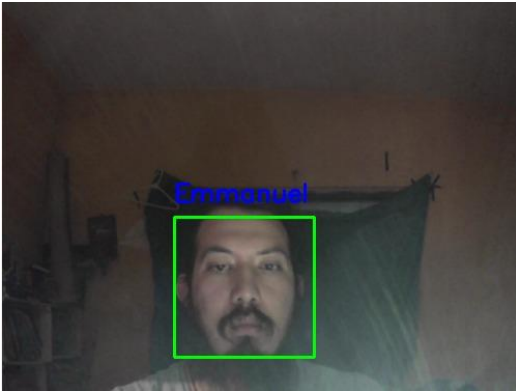


Figura 35. Usuario iluminado por luz de cámara

Se continúa realizando pruebas con diez usuarios, se verifica el número de veces que se detectó el rostro de los usuarios, se realizan diez pruebas con cada usuario. Las condiciones para que se cuente como éxito es que el rostro sea reconocido en los primeros cinco segundos, aun si después de ese tiempo el software reconoce el rostro no se tomará como éxito.

Se ensambla el prototipo de la Raspberry Pi 4 y se prueba con el código.

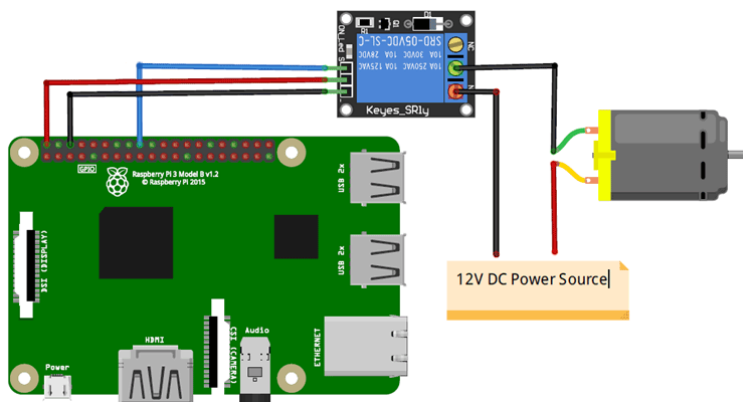


Figura 36. Diagrama de conexiones del sistema, (Elaboración propia usando Raspberry Pi Azure IoT Online Simulator).

Empleando Raspberry Pi Azure IoT Online Simulator se genera el diagrama de conexiones.

Se realizan las conexiones de la Raspberry Pi 4 B a la fuente de voltaje modulo rele. GPIO17, 5 V y tierra. Se conectan la fuente de voltaje al solenoide y al módulo rele, de la misma forma el solenoide se conecta al módulo rele y a la fuente de voltaje.

door unlock	door unlock
door unlock	door unlock
door unlock	door unlock
door unlock	door unlock
door unlock	door lock
door unlock	door unlock
door unlock	door lock
door unlock	door unlock
door unlock	door unlock
door lock	door lock

Figura 37. El sistema abriendo y cerrando dependiendo del rostro que se reconozca, (Elaboración propia usando IDE Thonny Python).

CAPÍTULO CUARTO.

4. RESULTADOS FINALES

Se genera un sistema de apertura y cierre empleando un sistema reconocimiento facial, este sistema se decantó por utilizar la placa de desarrollo Raspberry Pi 4 debido a la gran cantidad de información que existe y poder de procesamiento que tiene, así como tener la capacidad de usar una cámara tanto USB como una hecha para la placa de desarrollo, el número de pines y la comodidad de poder conectar la placa usando jumpers de distintos tipos, estos mismos jumpers se utilizan para conectar el módulo rele con el solenoide y la placa de desarrollo. (Figura 38.)

Se generaron tres códigos para cada una de las partes del software, una para tomar fotos, fotos que deben ser de frente, para que el algoritmo Haars cascade funcione debido a las zonas del rostro que este utilice para el reconocimiento, el segundo código es el de entrenamiento, se recomiendan por lo menos diez fotos para que el entrenamiento sea adecuado, si son menos la IA puede confundir familiares y pensar que son el mismo usuario en la parte del reconocimiento facial. El tercer código es el que permite reconocer el rostro, así como permitir que el solenoide se abra (Figura 40) y cierre (Figura 39), el sistema solo reconoce el rostro en ciertos ángulos

(Figuras 12, 24-28), siempre y cuando el sistema reconozca un rostro este abrirá el solenoide.

El sistema se recomienda que no se utilice en accesos principales, sino en accesos secundarios, así como gavetas o cajones donde se quiera más seguridad combinándolo con un sistema análogo.

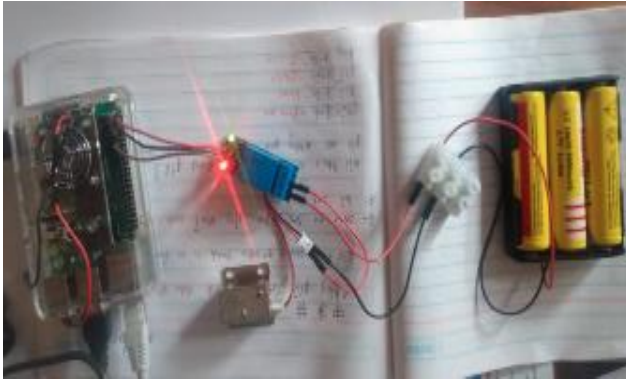


Figura 38. Sistema conectado y energizado, diagrama basado en el de (Namrata Singh et al, 2022)



Figura 39. Solenoide cerrado



Figura 40. Solenoide abierto

Resultados de las pruebas realizadas con diez usuarios, se verificó el número de veces que se detectó el rostro de los usuarios, se realizaron diez pruebas con cada usuario. Las condiciones para que se cuente como éxito es que el rostro sea reconocido en los primeros cinco segundos, aun si después de ese tiempo el software reconoce el rostro, no se tomó como éxito. El máximo tiempo que se llegó a registrar fueron siete segundos. (Tabla 1).

Usuario	Edad	Rostro de frente	Rostro de lado	Rostro 3/4	Rostro de frente Luz artificial	Rostro de frente oscuridad
Usuario E. L.	26	10	0	8	10	10
Usuario L. B.	49	8	0	7	9	9
Usuario M. L.	23	10	0	9	10	10
Usuario R. L.	51	9	0	8	10	10
Usuario A. B.	13	10	0	9	10	10
Usuario R. O.	4	8	0	7	10	9
Usuario G. E.	26	10	0	9	9	7
Usuario A. A.	27	10	0	9	10	9
Usuario L. O.	26	10	0	9	10	8
Usuario R. L.	10	10	0	9	10	8

Tabla 1. Resultados de pruebas

CAPITULO QUINTO.

5.1 Conclusiones.

Se logran los objetivos al realizar un sistema simple que ayuda a hacer las cosas más seguras, creando un sistema de doble verificación, aun si se tienen las llaves o una de las cerraduras se llega a ganzuar, el solenoide sigue funcionando, también puede utilizarse en gabinetes y cajones que poseen objetos peligrosos como químicos de limpieza de esta forma evitar que los niños tengan acceso (Figura 1).

El proyecto es más económico que las versiones comerciales, teniendo un costo entre \$2432.00 mxn a \$2837.00 la Raspberry Pi 4, mientras que las versiones comerciales pueden tener un costo desde los 24,999.25 mxn como es el caso de la cerradura digital con reconocimiento facial JKF31, o los \$13,900.00 mxn de Cerradura Multi-Biométrica de Reconocimiento Facial y Huella Digital ZM100.

Debido a la versión de Open CV que se empleó (4.2.0) y a la placa de desarrollo Raspberry Pi 4 que se usó, es posible cambiar la memoria micro SD y que funcione en versiones anteriores como lo pueden ser la Raspberry Pi 3 y Pi Zero 2 (Fabrerbas Ferran, 2020); sin embargo, debido a la versión de Open CV que se usa este proyecto con estas especificaciones solo funcionan en sistemas retro compatibles con la Raspberry Pi 4 y que posean una memoria mínima RAM de 1 GB (Open CV, 2017).

Es posible cambiar usuarios de forma remota siempre y cuando la placa de desarrollo este conectada a internet utilizando programas como lo es VNC Viewer. (Figura 7)

5.2. Recomendaciones.

Crear una interfaz de usuario para que sea más fácil la adición y remoción de usuarios.

Utilizar el modelo de cámara dependiendo los requerimientos de imagen.

Emplear una fuente de voltaje externa en caso de que se vaya la luz el sistema pueda seguir funcionando más tiempo.

Utilizar una placa de menor tamaño a las Raspberry Pi 4 para que sea más fácil de montar.

Utilizar el sistema en los accesos secundarios y gabinetes, el sistema debe ser acomodado de manera compacta imposibilitando el manipularlo de forma exterior.

Bibliografía

[1] Aivar Annamaa. (2015). Introducing Thonny, a Python IDE for learning programming.

[2] Andrew Ryan, Jeffery F. Cohn, Simon Lucey, Jason Saragih, Patrick Lucey, Fernando De la Torre, Adam Rossi. (2009). "Automated Facial Expression Recognition System," 43rd Annual 2009 International Carnahan Conference on Security Technology, pp. 172-177.

[3] Cadena Moreano José Augusto, Montaluisa Pulloquina Raúl Humberto, Flores Lagla Galo Alfredo, Chancúsig Chisag Juan Carlos, Guaypatín Pico Oscar Alejandro. (2017). Reconocimiento facial con base en imágenes.

[4] Carbajal Álvarez Mauricio, Chiñas Camargo María Alma, Ochoa Hernandez Juan Jose. (2009) Caracterización de rostros mediante procesamiento de imágenes digitales.

[5] Castaño Saavedra David Leonardo, Alonso Sierra Juan David. (2019) Sistema de reconocimiento facial para control de acceso a viviendas.

- [6] Costa Mari Daniel. (2020). Análisis de un sistema de reconocimiento facial a partir de una base de datos realizado mediante Python.
- [7] Domingo Jaramillo Cristina. (2021). Utilización del sistema de reconocimiento facial para preservar la seguridad ciudadana.
- [8] Dolores María. Lozano Ruiz. (2010) Un modelo para el desarrollo de sistemas de detección de situaciones de riesgo capaces de integrar información de fuentes heterogéneas. Aplicaciones.
- [9] Fabregas Ferran. (2020). Aprender Raspberry Pi 4 Con 100 Ejercicios Practicos / 3 Ed.
- [10] Goñi Oramas Ángel. (2012). Metodología para la gestión de proyectos de Consultoría en Migración a Tecnologías de Software Libre y Código Abierto.
- [11] Gutiérrez Juan F. Rendón Fredy G. Torres Alison B. (2019). Revisión de las técnicas de inteligencia artificial aplicadas en seguridad informática.
- [12] Giuseppe Bonaccorso. (2017). Machine Learning Algorithms.
- [13] Jia Nan, Li Cuimei, Qi Zhiliang, Wu Jianhua. (2017). Human face detection algorithm via Haar cascade classifier combined with three additional classifiers.
- [14] Morcillo Vizueté Francisco. (2020). Desarrollo de un sistema de reconocimiento facial utilizando Deep Learning con OpenCV.
- [15] Morcillo Vizueté, F. (2021). Estudio de las variables a considerar para el desarrollo de un modelo viable de aplicación de un sistema de identificación facial segura en un entorno de organizaciones de servicios. Universitat Politècnica de València. <http://hdl.handle.net/10251/174822>
- [16] Margarita García. 6 de agosto 2020. Aumentan robos en Cuernavaca. El Txoro matutino. <https://eltxoromatutino.com/aumentan-robos-en-cuernavaca/>

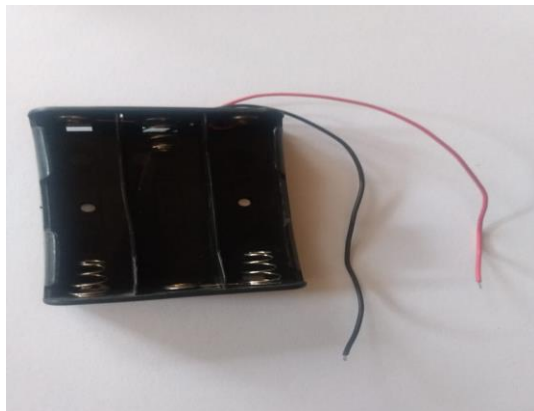
- [17] Namrata Singh, Dr. Ramveer Singh, Ranjan Kumar, Shivam Paliwal, Shipra Srivastava. (2022). ESP32 CAM Face Detection Door Lock. International Research Journal of Engineering and Technology (IRJET), (09)02, 1392-1393.
- [18] Noël Ruault Martínez. (2016) Inteligencia Artificial para Sistemas Colaborativos. Septiembre 2016.
- [19] Open CV. (2017). Laptop: Minimum System Requirement for OpenCV. <https://answers.opencv.org/question/179923/laptop-minimum-system-requirement-for-opencv/>
- [20] Pérez Cristian, y Piccolia Fabiana. (2010). Estimación de los parámetros de rendimiento de una GPU.
- [21] Rouhiainen Lasse. (2018). Inteligencia artificial 101 cosas que debes saber hoy sobre nuestro futuro.
- [22] Scarel, German Matías. (2010). sistema de reconocimiento facial, Universidad Nacional del Litoral, Facultad de Ingeniería y Ciencias Hídricas, pág 18. http://sinc.unl.edu.ar/sincpublications/2010/SMS10/sinc_SMS10.pdf
- [23] Shaw Andrew (2021) What is Raspberry Pi 4? Pinout, Specs, Projects & Datasheet <https://www.theengineeringprojects.com/2021/03/what-is-raspberry-pi-4-pinout-specs-projects-datasheet.html>, (20 de enero 2023).
- [24] Unir, La universidad del internet. (2022), 28 de abril. ¿Qué es un IDE en programación? <https://mexico.unir.net/ingenieria/noticias/ide-programacion/#:~:text=IDE%20es%20el%20acr%C3%B3nimo%20del,programaci%C3%B3n%20para%20desarrollar%20aplicaciones%2C%20juegos%E2%80%A6>, (28 de Abril del 2022)
- [25] Vázquez López Miguel Ángel. (2014). Sistema de Reconocimiento Facial Mediante Técnicas de Visión Tridimensional.

Anexos

Anexo 1



Anexo 1. Modulo Rele



Anexo 2. Porta pilas



Anexo 3. Jumpers



Anexo 4. Raspberry Pi 4



Anexo 5. Fuente de alimentación para Raspberry Pi 4



Anexo 6. Batería de litio 18650



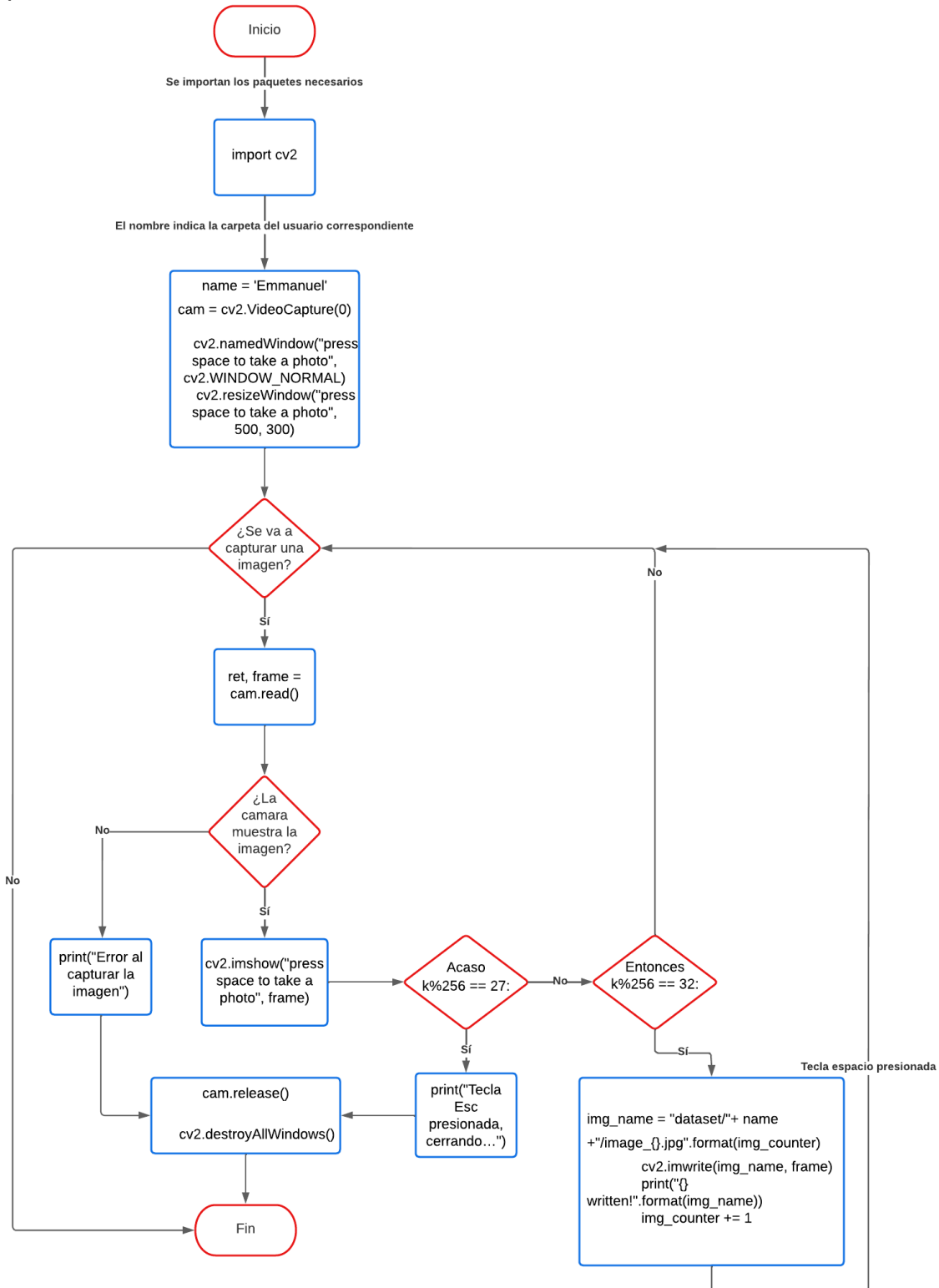
Anexo 7. Cámara web



Anexo 8. Módulo de cámara para Raspberry Pi

Anexo 9. Diagrama de código para captura de rostro

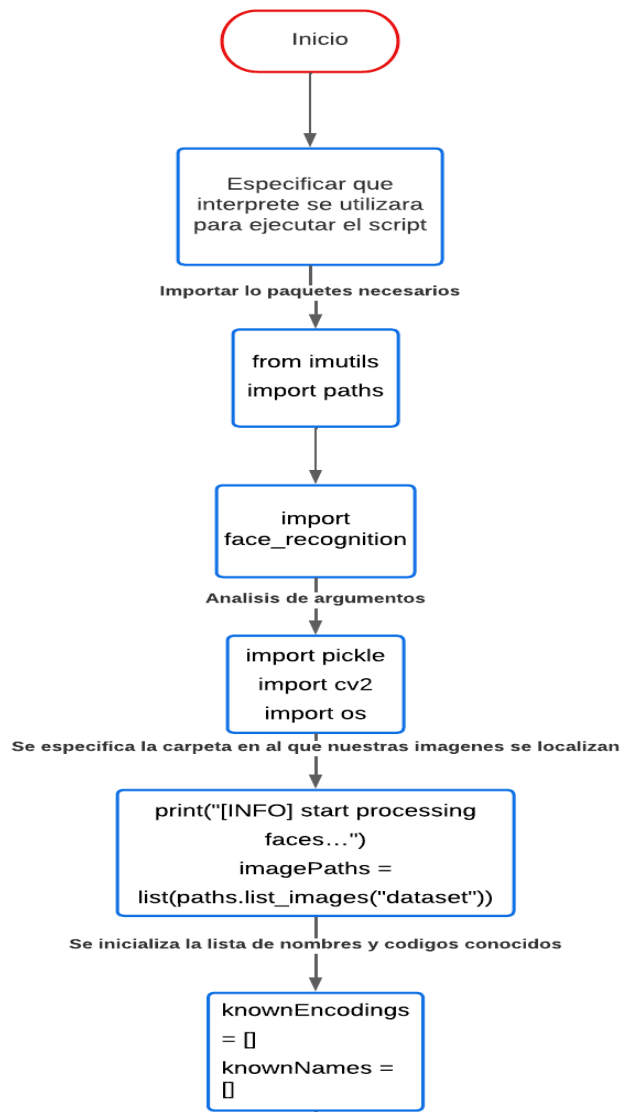
Para comodidad del lector, algunos de los diagramas serán divididos en varias partes.



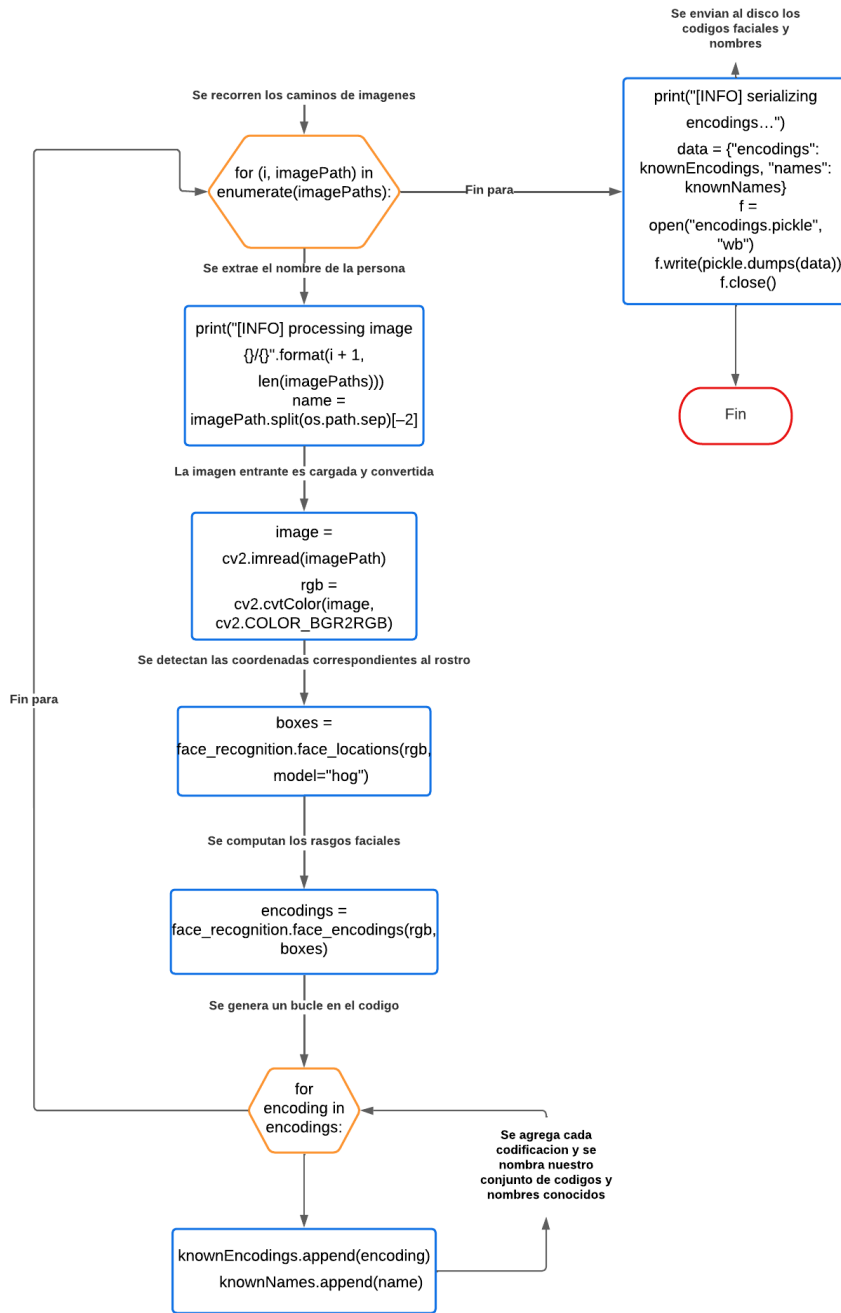
Anexo 10. Diagrama modelo de entrenamiento



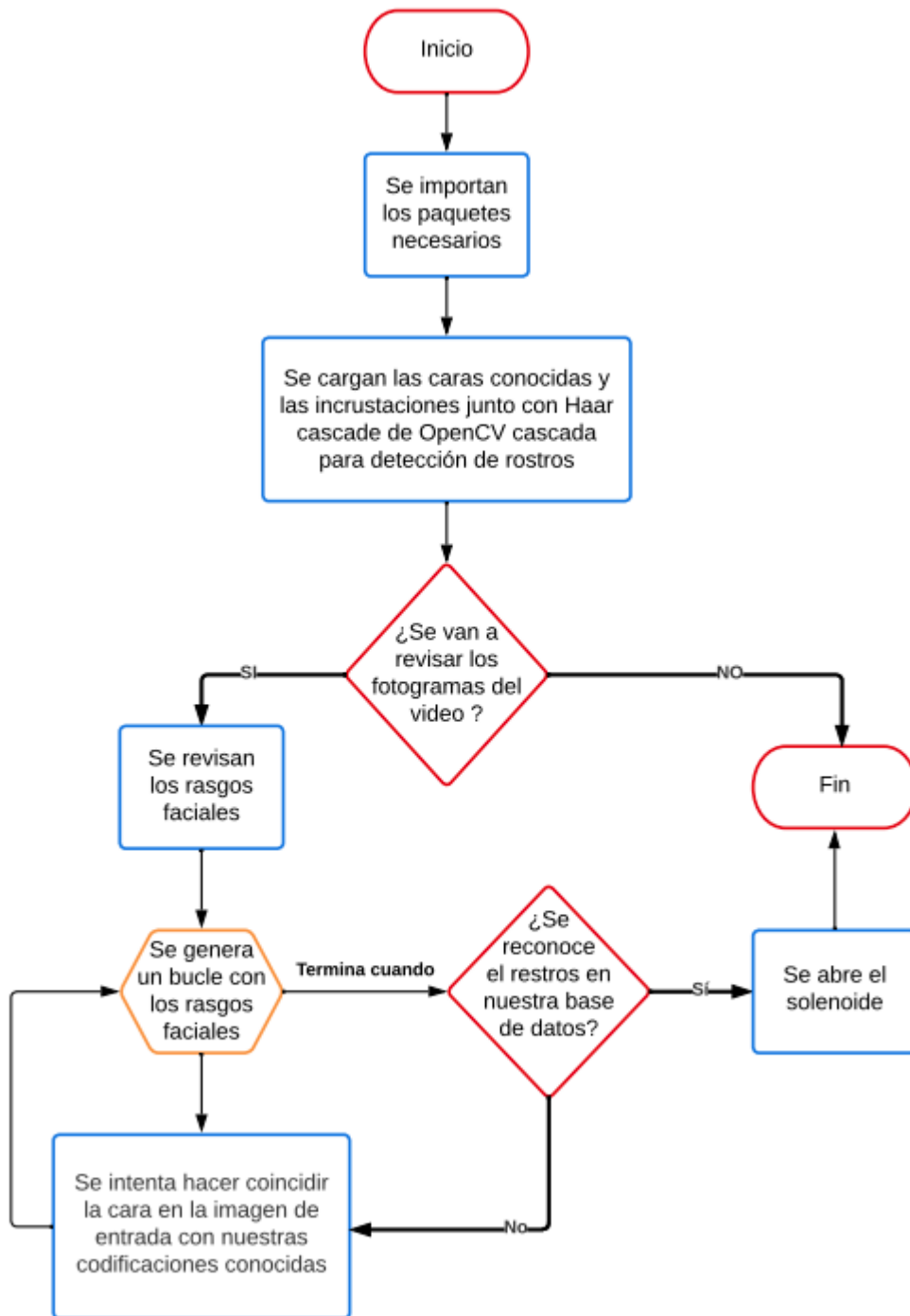
Anexo 11. Diagrama de modelo de entrenamiento, primeros dos bloques



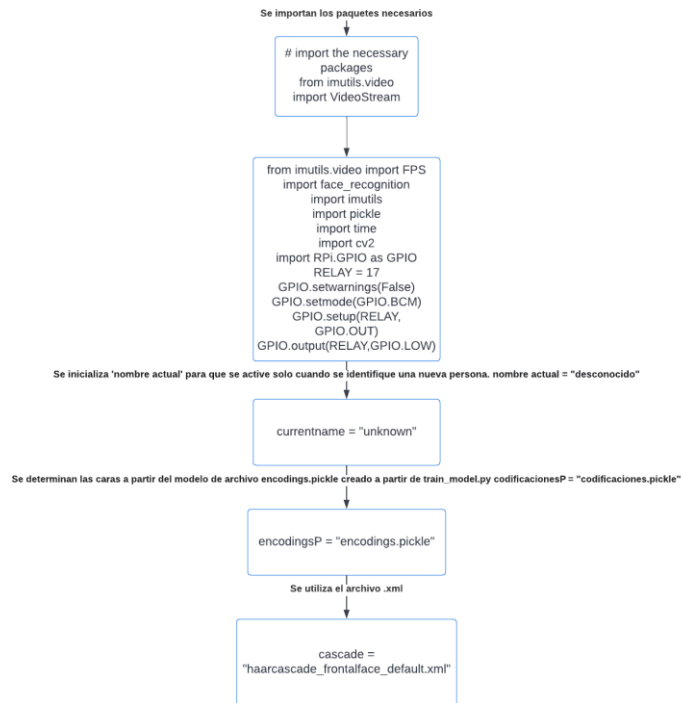
Anexo 12. Diagrama de modelo entrenamiento, segundos bloques.



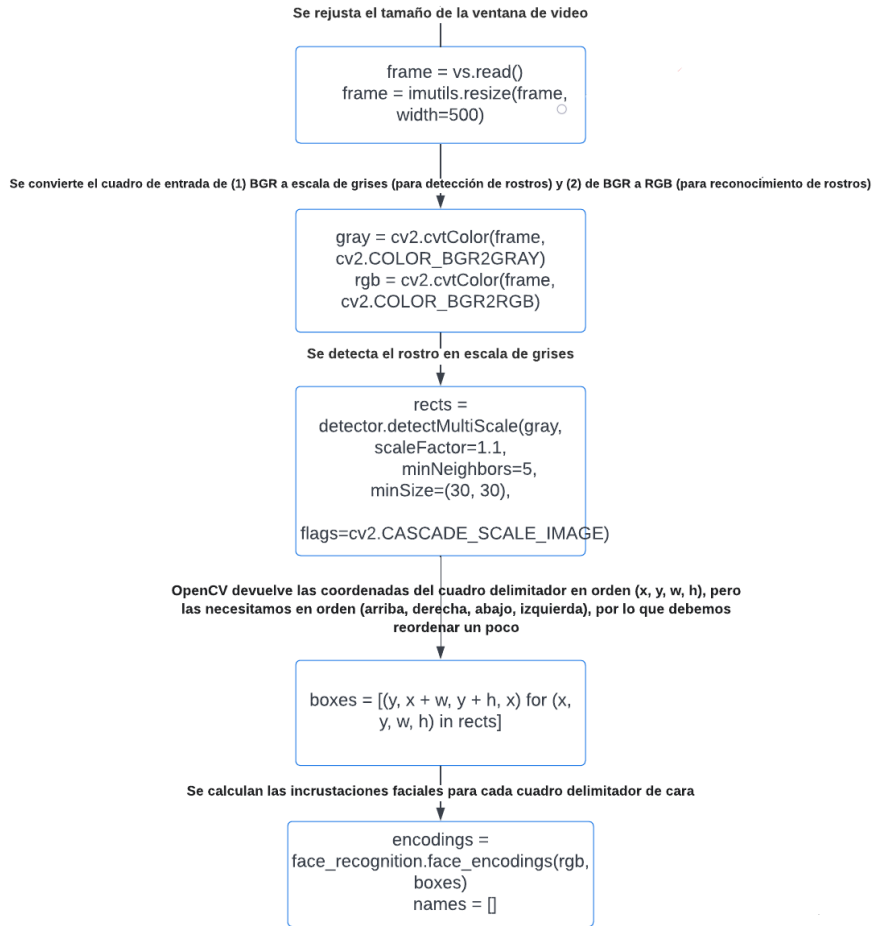
Anexo 13. Diagrama reconocimiento facial y candado



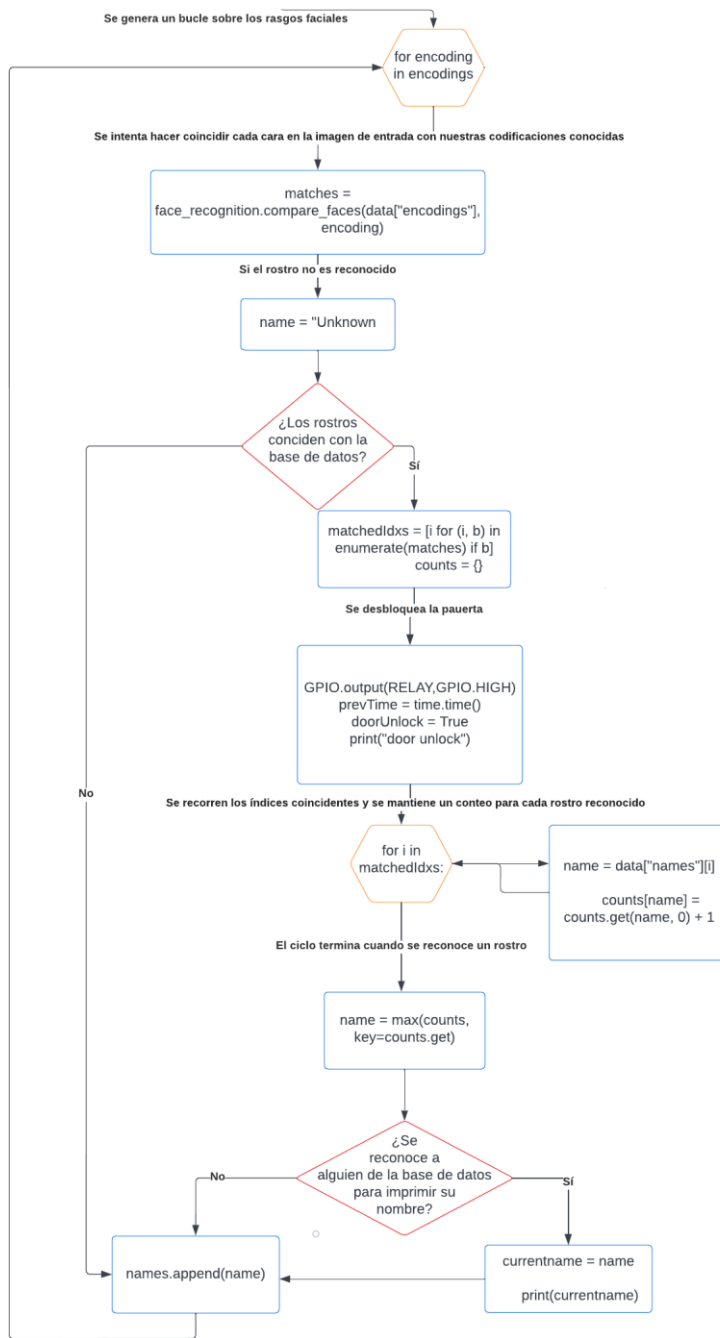
Anexo 14. Diagrama modelo de entrenamiento, desglose de importación de paquetes.



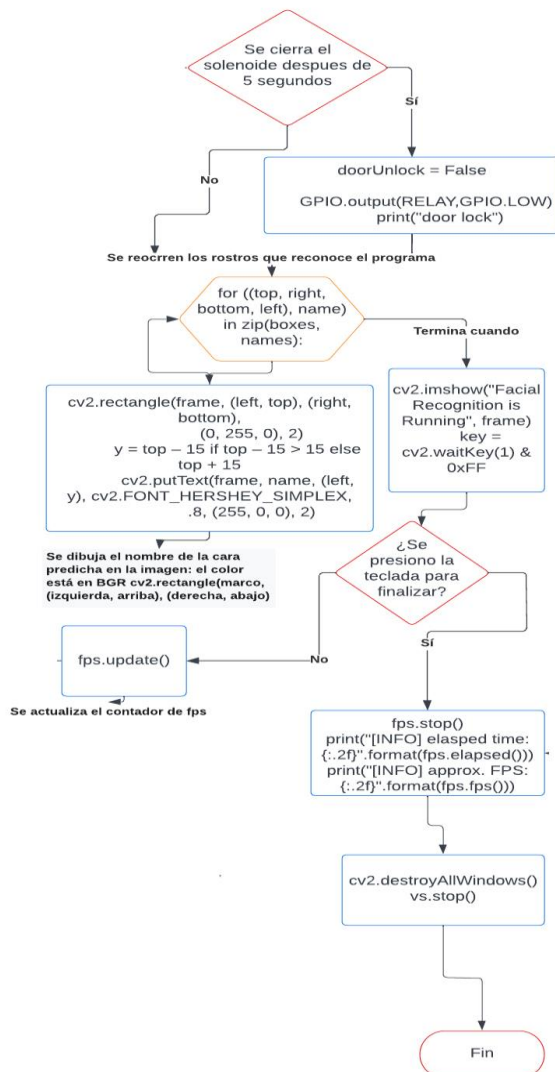
Anexo 15. Diagrama reconocimiento facial, desglose de bloques de rasgos faciales.



Anexo 16. Diagrama reconocimiento facial, bucle de reconocimiento facial.



Anexo 17 Reconocimiento facial, desglose de solenoide.



Anexo 18. Código de captura de rostro

```
import cv2

name = 'Emmanuel' #Se reemplaza con el nombre del usuario a capturar rostro
cam = cv2.VideoCapture(0)
cv2.namedWindow("press space to take a photo", cv2.WINDOW_NORMAL)
cv2.resizeWindow("press space to take a photo", 500, 300)

img_counter = 0

while True:
    ret, frame = cam.read()
    if not ret:
        print("No se pudo capturar imagen")
        break
    cv2.imshow("Presione la tecla espacio para capturar foto", frame)
    k = cv2.waitKey(1)
    if k%256 == 27:
        # Tecla ESC presionada
        print("Tecla ESC presionada, cerrando...")
        break
    elif k%256 == 32:
        # Tecla espacio presionada
        img_name = "dataset/"+ name +"/image_{}.jpg".format(img_counter)
        cv2.imwrite(img_name, frame)
        print("{} written!".format(img_name))
        img_counter += 1

cam.release()

cv2.destroyAllWindows()
```

Anexo 19. Código de entrenamiento de reconocimiento de rostros

```
#!/usr/bin/python

# Se importan los paquetes necesarios
from imutils import paths
import face_recognition

# Se importa argparse
import pickle
import cv2
import os

# Nuestras imágenes están localizadas en la carpeta dataset
print("[INFO] Inicio de procesamiento de rostros...")
imagePaths = list(paths.list_images("dataset"))

# Se inicializa la lista de códigos y nombres conocidos
knownEncodings = []
knownNames = []

# Se genera un bucle en los caminos de imagenes
for (i, imagePath) in enumerate(imagePaths):
    # Se extrae el nombre de la persona del camino de imágenes
    print("[INFO] processing image {}/{}".format(i + 1,
        len(imagePaths)))
    name = imagePath.split(os.path.sep)[-2]

    # Se carga la imagen de entrada y se convierte de load RGB (OpenCV ordering) a
    # dlib ordering (RGB)

    image = cv2.imread(imagePath)
    rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
```

Se detectan las coordenadas (x,y) de las cajas del rostro correspondiente detect the (x, y)-coordinates of the bounding boxes

```
boxes = face_recognition.face_locations(rgb,  
    model="hog")
```

Se computan los rasgos faciales

```
encodings = face_recognition.face_encodings(rgb, boxes)
```

Se genera un bucle en los codigos

```
for encoding in encodings:
```

Se agrega cada codificación más el nombre a nuestro conjunto de nombres y códigos conocidos

```
    knownEncodings.append(encoding)
```

```
    knownNames.append(name)
```

Se envían las codificaciones faciales y los nombres al disco

```
print("[INFO] serializing encodings...")
```

```
data = {"encodings": knownEncodings, "names": knownNames}
```

```
f = open("encodings.pickle", "wb")
```

```
f.write(pickle.dumps(data))
```

```
f.close()
```

Anexo 20. Código de apertura y cierre con reconocimiento facial

```
#!/usr/bin/python

# Se importan los paquetes necesarios
from imutils.video import VideoStream
from imutils.video import FPS
import face_recognition
import imutils
import pickle
import time
import cv2
import RPi.GPIO as GPIO

RELAY = 17

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
GPIO.setup(RELAY, GPIO.OUT)
GPIO.output(RELAY,GPIO.LOW)

# Inicializa 'nombre actual' para que se active solo cuando se identifique una nueva persona.
currentname = "unknown"

# Determine las caras a partir del modelo de archivo encodings.pickle creado a partir de
train_model.py

encodingsP = "encodings.pickle"

# Se utiliza el archivo xml a continuación.

#https://github.com/opencv/opencv/blob/master/data/haarcascades/haarcascade_frontalface_defau
lt.xml

cascade = "haarcascade_frontalface_default.xml"

# Se cargan las caras conocidas y las incrustaciones junto con Haarcascade de OpenCV para
detección de rostrosprint("[INFO] loading encodings + face detector...")

data = pickle.loads(open(encodingsP, "rb").read())
```

```

detector = cv2.CascadeClassifier(cascade)

# Se inicializa la transmisión de video y permite que el sensor de la cámara se active
print("[INFO] comenzando transmisión de video...")vs = VideoStream(src=0).start()
#vs = VideoStream(usePiCamera=True).start()
time.sleep(2.0)

# Inicio el contador de FPS
fps = FPS().start()

prevTime = 0
doorUnlock = False

# Se recorren fotogramas de la secuencia de archivos de vídeo
while True:
    # Se toma el marco de la secuencia de video encadenada y se cambia su tamaño a
    # 500 px (para acelerar el procesamiento)
    frame = vs.read()
    frame = imutils.resize(frame, width=500)

    # Se convierte el cuadro de entrada de (1) BGR a escala de grises (para detección de
    # la cara ) y (2) de BGR a RGB (para reconocimiento facial)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    rgb = cv2.cvtColor(frame, cv2.COLOR_BGR2RGB)

    # Se detecta el rostro en escala de grises
    rects = detector.detectMultiScale(gray, scaleFactor=1.1,
    minNeighbors=5, minSize=(30, 30),
    flags=cv2.CASCADE_SCALE_IMAGE)

    # OpenCV devuelve las coordenadas del cuadro delimitador en orden (x, y, w, h), pero
    # las necesitamos en orden (arriba, derecha, abajo, izquierda), por lo que debemos reordenar un poco

```

```

boxes = [(y, x + w, y + h, x) for (x, y, w, h) in rects]

# Se computan las incrustaciones faciales para cada cuadro delimitador de cara
encodings = face_recognition.face_encodings(rgb, boxes)
names = []

# Se genera un bucle sobre las incrustaciones faciales
for encoding in encodings:
    # Se intenta hacer coincidir cada cara en la imagen de entrada con nuestras
    # codificaciones conocidas
    matches = face_recognition.compare_faces(data["encodings"],
        encoding)
    #Si el rostro no se reconoce que se imprimi Unkown.
    name = "Unknown"

    # Se revisa y se encuentran concordancias
    if True in matches:
        # Se encuentra los índices de todas las caras coincidentes y luego se
        # inicializa un diccionario para contar el número total de veces que se ha emparejado cada cara
        matchedIdxs = [i for (i, b) in enumerate(matches) if b]
        counts = {}

        # Se desbloquea la cerradura
        GPIO.output(RELAY,GPIO.HIGH)
        prevTime = time.time()
        doorUnlock = True
        print("door unlock")

        # Se recorren los índices coincidentes y mantiene un conteo para cada rostro
reconocido
        for i in matchedIdxs:

```

```

        name = data["names"][i]
        counts[name] = counts.get(name, 0) + 1

        # Determine la cara reconocida con el mayor número de votos (nota: en el
        # caso de un empate improbable, Python seleccionará la primera entrada en el diccionario)
        name = max(counts, key=counts.get)

        # Si se identifica a alguien en su conjunto de datos, aparece el nombre en la
pantalla
        if currentname != name:
            currentname = name
            print(currentname)

        # Se actualiza la lista de nombres
        names.append(name)

#Se cierra la cerradura después de cinco segundos
        if doorUnlock == True and time.time() - prevTime > 5:
            doorUnlock = False
            GPIO.output(RELAY,GPIO.LOW)
            print("door lock")

        # Se recorren las caras reconocidas
        for ((top, right, bottom, left), name) in zip(boxes, names):
            # Se dibuja el nombre de la cara predicha en la imagen: el color está en BGR
            cv2.rectangle(frame, (left, top), (right, bottom),
                (0, 255, 0), 2)
            y = top - 15 if top - 15 > 15 else top + 15
            cv2.putText(frame, name, (left, y), cv2.FONT_HERSHEY_SIMPLEX,
                .8, (255, 0, 0), 2)

```

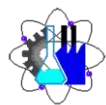
```
# Se muestra la imagen en nuestra pantalla
cv2.imshow("Facial Recognition is Running", frame)
key = cv2.waitKey(1) & 0xFF

# Cuando la tecla q es presionada se termina el proceso
if key == ord("q"):
    break

# Se actualiza el contador de FPS
fps.update()

# Se detiene el temporizador y se muestra la información de FPS
fps.stop()
print("[INFO] elapsed time: {:.2f}".format(fps.elapsed()))
print("[INFO] approx. FPS: {:.2f}".format(fps.fps()))

# Se cierran todas las ventanas
cv2.destroyAllWindows()
vs.stop()
```

Cuernavaca, Mor., a 13 de diciembre de 2022

**DR. JOSE GERARDO VERA DIMAS
DR. LUIS CISNEROS VILLALOBOS
DRA. ALINA MARTÍNEZ OROPEZA
DR. OUTMANE OUBRAME
MTRO. FCO. JAVIER BECERRA GONZALEZ
P R E S E N T E**

Me permito comunicarles que han sido designados integrantes del **COMITÉ REVISOR** del trabajo de: **TESIS**

Titulado:

Armado y desarme de sistema de acceso empleando reconocimiento facial

Que presenta (el) o (la) **C. JOSÉ EMMANUEL LAGUNAS BUENOS AIRES**, del programa educativo de **LICENCIATURA EN INGENIERÍA ELÉCTRICA-ELECTRÓNICA**.

A T E N T A M E N T E
Por una humanidad culta

**DRA. VIRIDIANA AYDEÉ LEÓN HERNÁNDEZ
DIRECTORA**

Se anexa firma electrónica

D I C T Á M E N

**DRA. VIRIDIANA AYDEÉ LEÓN HERNÁNDEZ
DIRECTORA DE LA FCQeI
P R E S E N T E**

En respuesta a su amable solicitud para emitir DICTÁMEN sobre el trabajo que se menciona, me permito informarle que nuestro voto es:

VOTO	NOMBRE	FIRMA
	DR. JOSE GERARDO VERA DIMAS	
	DR. LUIS CISNEROS VILLALOBOS	
	DRA. ALINA MARTÍNEZ OROPEZA	
	DR. OUTMANE OUBRAME	
	MTRO. FCO. JAVIER BECERRA GONZALEZ	

El voto del comité es aprobatorio, se anexan firmas electrónicas
VALH/fjbg



UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

VIRIDIANA AYDEE LEON HERNANDEZ | Fecha:2022-12-15 22:48:02 | Firmante

i32iVvkuwNMZYEWUym9Q0rJ2CE5vxh31HrYkU5mJmFIEV/PKQjqFKHpV7GvC888L4+3ZRvhSeaNnMfhS3qEQT/kOYXJaCdFFnPsxXNbHbF74siNlu6vCdgyqDLBurS8u9RrUu3
LOKL5+pMFk9pE/5zTxiS0HkAr1wtlKOEJnUsg/qGZOPqFiTbpyQahrW45nzMR2H7XkzBlyPnplZMslRlPtZbK9dYOyyD69Ofa9TmRmeRXdQm/xAENoMNVVvq3C9WTtF882VEa
FPWXRxBd9Aq2eLmBp5ePLLuM42UYTlrq92+CECnNWSex71kuiJLZfhPIUfYcfudJqJalliXGhhQ==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o
escaneando el código QR ingresando la siguiente clave:



8vi1yTUK

<https://efirma.uaem.mx/noRepudio/m46YJD1w6IXROkRclodW3nivhPwZnoyV>





UNIVERSIDAD AUTÓNOMA DEL
ESTADO DE MORELOS

Se expide el presente documento firmado electrónicamente de conformidad con el ACUERDO GENERAL PARA LA CONTINUIDAD DEL FUNCIONAMIENTO DE LA UNIVERSIDAD AUTÓNOMA DEL ESTADO DE MORELOS DURANTE LA EMERGENCIA SANITARIA PROVOCADA POR EL VIRUS SARS-COV2 (COVID-19) emitido el 27 de abril del 2020.

El presente documento cuenta con la firma electrónica UAEM del funcionario universitario competente, amparada por un certificado vigente a la fecha de su elaboración y es válido de conformidad con los LINEAMIENTOS EN MATERIA DE FIRMA ELECTRÓNICA PARA LA UNIVERSIDAD AUTÓNOMA DE ESTADO DE MORELOS emitidos el 13 de noviembre del 2019 mediante circular No. 32.

Sello electrónico

JOSE GERARDO VERA DIMAS | Fecha:2022-12-16 11:52:17 | Firmante

UqMVh15S7Raavy1B/ieeABWFyQHrI0DarvclNI67g5G2D+SV+zmyQUyyKCVsSdXuQV4M0K5XN8nBL3ZVipqER8h/ptN8eSE9KHmzT3d2fCk7fRxu4X10J4HDznCrsoP2ZKxciShpwz5eGWodxz64UUZ8vUblg1VpmT66jYZ0C0kdl1PVVvfDxTzcxWTGZZIEKOjh5L6foxcXRTQ+dMkUto+MERAp/rQLb9Dsd7YNIRWlseJvBul0PT8qrgbl+mmOPXfCfywP24Pk3He8oFyCvEo5ziwuzwcp4UwSv6TMD7WmyirBfwzNUqvaf13bVCpktwxrIR85XkqIRSVsQ==

FRANCISCO JAVIER BECERRA GONZALEZ | Fecha:2022-12-16 12:00:05 | Firmante

C7mttkajAcDxvaedejwSfYaUzRppKo2hj3/dD0S7BllrgTScnkwnphymtpaxOApQ7sN/Ak68jNNsGH8LXBSBAWpOeXf3Hf01+3oXfgNcDU+J5aagmbxId+h1zSaQB40IP19INayNP4P/errdOsszsOS5dp+vOkRU3VveM+gEWbl1j5cl41Nutha5AfHmRdxu1HqllCHz5cDMA55s9e3OMDMnHLVexWcy7QMRgQRTCo5j1kUHQvdFM4Ehweui2whWEKedIPNMe/ljud7o+h+K9NqvJcOLa0eKz6RCKGy7hF/b3rTk/Hv3WMM9T+mGq5NwWtCSvGcp5As5X+zL7YN0A==

OUTMANE OUBRAM | Fecha:2022-12-16 15:28:11 | Firmante

ZdmwZzqvdYY3vKv9K9b9JeAhFBrixGLiBpTWdy9yDhwy82cfknsHbtBq/DGtt0QrAjOeBjOefJcDgCF3oRbOvNcA8n Cz0vwwf+dpA0y4KZUGkvKOLcCE+rdTZslCPkh8Mz0VHQULRDIb4nJmQgl56LPuiKEyC0Nj1Ece6n/4x+PTussG9S5WuWfZ0axWYCeusJkOww97bAjnn9E02nV5/ZlVwJf7juhtJTVADEtzRABb5PughliLebWkRnOr+XL6X8zfp9WwyN8vLDDsjfZxIBCayHYT+9GcakZyQ2399ysR+CXwBStqZtFpNs78rpGkPm0MldKkNvka5DvB0sHMA==

ALINA MARTINEZ OROPEZA | Fecha:2022-12-19 12:07:14 | Firmante

mpOzvZ2Nr8s2kmCFyaaht0zzgK81V+Sw5orxYJ+3UH8OIFkDJy+TGqBIXGeyUAsbFkjMJWqedxXSpPHBKuB3rVGjzNFrQh4sc5T4mKGCuOEE3itud9HZ2Ulf6VYpx5nFSPqSoa7KNqVSRpTi3wPqlrDLDDIbWgNCaEM08bkoevR2iyyzSsApybQD65v5MCHaivvP+FLY7Y4LVD1BDH9dtvlfH3ocKLH2Kib5F+WRt3fQYE1+wEP5liKEA3aunNUx/2CzA/DfthBliZ+O9q4QA1WZd5+jwJtlc2YilZwoQHhKbf7g00u513bZRUtuM2GjJNxc/hJlWQk/5kShA==

LUIS CISNEROS VILLALOBOS | Fecha:2022-12-24 15:08:24 | Firmante

bGvot2BfruOhrdM60ES0hfFsWwNoiFFyLXwbj3xJYW4DfMjQrWakI1K7iR4Hq1Wum90XEWiGNz6NGQm0BWg5wYmht4HvGTy6sNeuRobKod34gumRxxoelSNEeLQOFsxc5t1tbGvY8G8ua1XkFRBdUbHz95qDuLcMhHefySpSW4gT6MSr27nAzBpIkNleywrSnH7yQjrrUQ0trjbvFvG1Q2F7ntgkUSOXp9mBTXlJ4e2jh/A//C08O6W5/m+GFwwMGdZQYzXssmbSCJKVfDeRnJNJBBSol9774hlqmZzjiUUT1KSSVfpc9LexHXDKdeBrylpxKlfgKuo1pxNGMzW==

Puede verificar la autenticidad del documento en la siguiente dirección electrónica o escaneando el código QR ingresando la siguiente clave:



JS9aGVitR

<https://efirma.uaem.mx/noRepudio/xdYElor7IOMVWyC0WzQhDFo08ZqCxsG>

