



UNIVERSIDAD DE GUADALAJARA

VICERRECTORÍA EJECUTIVA

SISTEMA UNIVERSITARIO DE BIBLIOTECAS

Biblioteca Digital

La presente tesis es publicada a texto completo en virtud de que el autor ha dado su autorización por escrito para la incorporación del documento a la Biblioteca Digital y al Repositorio Institucional de la Universidad de Guadalajara, esto sin sufrir menoscabo sobre sus derechos como autor de la obra y los usos que posteriormente quiera darle a la misma.



Centro Universitario de los Valles

Maestría en Ingeniería de software

Sistema para la gestión del acceso y uso de aulas, vía reconocimiento facial basado en Raspberry Pi.

Tesis que para obtener el grado de Maestro en Ingeniería de Software

Presenta

José Manuel Nava Cervantes

Director

Dr. Rodolfo Omar Domínguez García

Codirector

Mtro. José Roberto Lomelí Huerta

Ameca, Jalisco, diciembre de 2022

Índice general

ÍNDICE DE TABLAS.....	V
ÍNDICE DE FIGURAS.....	VI
RESUMEN	X
ABSTRACT	XI
CAPÍTULO 1. INTRODUCCIÓN	2
1.1 PLANTEAMIENTO DEL PROBLEMA.....	3
1.2 JUSTIFICACIÓN.....	4
1.3 CONTRIBUCIONES.....	5
1.4 OBJETIVO GENERAL.....	5
1.4.1 <i>Objetivos particulares</i>	5
1.5 ALCANCES Y LIMITACIONES.....	6
1.6 ESTRUCTURA DEL DOCUMENTO	6
CAPÍTULO 2. MARCO TEÓRICO Y ESTADO DEL ARTE	9
2.1 METODOLOGÍA DE DESARROLLO DE SOFTWARE	9
2.2 CONCEPTOS GENERALES	10
2.3 FASES DE UN SISTEMA DE RECONOCIMIENTO FACIAL	12
2.3.1 <i>Detección</i>	12
2.3.2 <i>Preprocesado</i>	12
2.3.3 <i>Extracción de características</i>	13
2.3.4 <i>Técnicas basadas en apariencias</i>	14
2.3.5 <i>Comparación y decisión</i>	14
2.4 ALGORITMO PARA EL RECONOCIMIENTO FACIAL.....	15
2.4.1 <i>Detección de rostros</i>	15
2.4.2 <i>Algoritmos de entrenamiento</i>	16
2.4.2.2 <i>Fisherfaces</i>	19
2.5 ESTADO DEL ARTE.....	21
2.5.1 <i>Trabajos relacionados</i>	21
2.5.2 <i>Productos comerciales</i>	23
CAPÍTULO 3. REQUERIMIENTOS	27
3.1 FUNCIONES DEL PRODUCTO DE SOFTWARE	27
3.2 REQUERIMIENTOS DEL BANCO DE IMÁGENES.....	27
3.3 REQUERIMIENTO DEL MODELO DE ENTRENAMIENTO.....	28
3.4 REQUERIMIENTOS DEL DISPOSITIVO DE RF	29
3.5 RESTRICCIONES DE DISEÑO DEL PROTOTIPO FÍSICO.....	29
3.5.1 <i>Elicitación de requerimientos</i>	30
3.6 ESPECIFICACIÓN DE REQUERIMIENTOS (IEEE29148)	30
3.6.1 <i>Listado general de requerimientos</i>	31
CAPÍTULO 4. METODOLOGÍA.....	35
4.1 ANÁLISIS Y COMPARACIÓN DE METODOLOGÍAS EXISTENTES	35
4.2 METODOLOGÍA ESPIRAL.....	35
4.2.1 <i>Metodología en espiral</i>	35
4.2.2 <i>Fases de la metodología en espiral</i>	37
4.3 HERRAMIENTAS DEL SOFTWARE DESARROLLADO	38

4.3.1	Comportamiento de los algoritmos de reconocimiento facial.....	38
4.3.2	Algoritmo de detección.....	39
CAPÍTULO 5.	DISEÑO	41
5.1	ARQUITECTURA.....	41
5.1.1	Diagrama general.....	41
5.1.2	Diagrama de clases.....	42
5.1.3	Diagrama de casos de uso	43
5.2	VISTA LÓGICA DEL SISTEMA	45
5.2.1	Diagrama de paquetes.....	45
5.3	VISTA FÍSICA DEL SISTEMA	46
5.3.1	Diagrama de despliegue	46
5.3.2	Diagrama de implementación	47
5.4	VISTA DE PROCESOS DEL SISTEMA	47
5.4.1	Diagrama de actividad.....	47
5.5	COMPORTAMIENTO DEL SISTEMA	49
5.5.1	Diagrama de secuencia.....	49
5.5.2	Modelo entidad-relación.....	50
5.6	INTERFACES GRÁFICAS	51
5.6.1	Interfaz para gestión de usuarios (IGU)	51
5.6.2	Interfaz para Raspberry Pi	52
CAPÍTULO 6.	DESARROLLO E IMPLEMENTACIÓN	55
6.1	HARDWARE A UTILIZAR.....	55
6.1.1	Raspberry Pi.....	55
6.1.2	Mini sensor de presencia PIR HC-SR505.....	56
6.1.3	Cerradura electromecánica.....	56
6.1.4	Módulo cámara Raspberry Pi.....	57
6.2	PROTOTIPADO	58
6.2.1	Prototipo 1 (laptop)	58
6.2.2	Prototipo 2 (MDF Carcasa)	60
6.2.3	Prototipo 3 (Primera impresión externa)	64
6.3	PROTOTIPO FINAL.....	68
6.3.1	Hardware	68
6.4	CODIFICACIÓN	71
CAPÍTULO 7.	PRUEBAS.....	77
7.1	CONCEPTOS IMPORTANTES.....	77
7.2	CASOS DE PRUEBA	77
7.2.1	Listado general de documentación de pruebas	78
7.2.2	Resultados.....	80
CAPÍTULO 8.	RESULTADOS, CONCLUSIONES Y TRABAJOS FUTUROS	91
8.1	RESULTADOS	91
8.1.1	Software para gestión de usuarios	91
8.1.2	Sistema módulo Pi.....	96
8.1.3	Comparativa entre algoritmos de reconocimiento	100
8.1.4	API Reloj Checador	101
8.2	CONCLUSIONES.....	102
8.3	TRABAJOS FUTUROS.....	103
REFERENCIAS	105	
APÉNDICES.....	109	

Índice de tablas

TABLA 1.-	CARACTERÍSTICAS DE LAS IMÁGENES ALMACENADAS	28
TABLA 2.-	FORMATO DE REDACCIÓN DE REQUERIMIENTOS.	31
TABLA 3.-	LISTADO DE REQUERIMIENTOS TABULADOS.....	31
TABLA 4.-	ESPECIFICACIONES TÉCNICAS RASPBERRY PI.....	55
TABLA 5.-	ESPECIFICACIONES TÉCNICAS SENSOR PIR	56
TABLA 6.-	ESPECIFICACIONES TÉCNICAS CERRADURA AX-LOCKR.....	56
TABLA 7.-	ESPECIFICACIONES TÉCNICAS MÓDULO CÁMARA	57
TABLA 8.-	RENDIMIENTO EN PRUEBAS DE ALGORITMOS DE ENTRENAMIENTO	62
TABLA 9.-	TABLA DE RECURSOS EXTERNOS USADOS	68
TABLA 10.-	FORMATO DE REDACCIÓN DE PRUEBAS.....	78
TABLA 11.-	TABLA DE CASOS DE PRUEBA	79
TABLA 12.-	PRUEBA UNITARIA PU-01.	80
TABLA 13.-	PRUEBA UNITARIA PU-05	81
TABLA 14.-	PRUEBA DE INTEGRACIÓN PI-01	82
TABLA 15.-	PRUEBA DE INTEGRACIÓN PI-02	83
TABLA 16.-	PRUEBA DE INTEGRACIÓN PI-17	84
TABLA 17.-	PRUEBA DE INTEGRACIÓN PI-18	85
TABLA 18.-	PRUEBA DE SISTEMA PS-01.	86
TABLA 19.-	PRUEBA DE SISTEMA PS-04.	87
TABLA 20.-	PRUEBA DE ACEPTACIÓN PA-01.	88
TABLA 21.-	PRUEBA DE ACEPTACIÓN PA-01.	89
TABLA 22.-	COMPARATIVA DE RECONOCIMIENTO CON 50 IMÁGENES.....	100
TABLA 23.-	COMPARATIVA DE RECONOCIMIENTO CON 100 IMÁGENES.....	100

Índice de figuras

FIGURA 1.-	MODELO EN ESPIRAL PROPUESTO POR BARRY BOEHM EN 1986.....	9
FIGURA 2.-	DIAGRAMA GENERAL DE UN SISTEMA DE RECONOCIMIENTO FACIAL.	12
FIGURA 3.-	DIAGRAMA ETAPA ENTRENAMIENTO DE UN SISTEMA DE RECONOCIMIENTO FACIAL.	12
FIGURA 4.-	TAXONOMÍA DEL RECONOCIMIENTO FACIAL UTILIZADA EN ESTA INVESTIGACIÓN	13
FIGURA 5.-	DIFERENTES MÁSCARAS (KERNEL) DE HAAR.	15
FIGURA 6.-	CARACTERÍSTICAS RELEVANTES EN HAAR CASCADE.	16
FIGURA 7.-	EJEMPLO DE APLICACIÓN DEL OPERADOR LBP [35].	20
FIGURA 8.-	DEFINICIÓN DE ZONAS POR EL OPERADOR LBP [36].	20
FIGURA 9.-	DISPOSITIVO USE SMART LOCK.....	24
FIGURA 10.-	CERRADURA WIFI- RECONOCIMIENTO FACIAL CON 6 MODOS DE APERTURA.....	25
FIGURA 11.-	FORMA DE ALMACENAR LAS IMÁGENES EN LAS CARPETAS.	28
FIGURA 12.-	DIMENSIONES FÍSICAS DEL SISTEMA.....	29
FIGURA 13.-	CICLO DE VIDA DEL PROYECTO Y SUS FASES PLANTEADAS	36
FIGURA 14.-	EXPLICACIÓN GRÁFICA DEL PROCESO DE GESTIÓN DE USUARIO.	38
FIGURA 15.-	DETECCIÓN DE ROSTROS CON HAAR CASCADE	39
FIGURA 16.-	ARQUITECTURA DEL SISTEMA BASADA EN RASPBERRY PI.	41
FIGURA 17.-	DIAGRAMA DE CLASES.....	42
FIGURA 18.-	DIAGRAMA DE CASO DE USO PARA EL USUARIO NUEVO EN EL SISTEMA.	43
FIGURA 19.-	DIAGRAMA DE CASOS DE USO PARA USUARIO REGISTRADO.	44
FIGURA 20.-	DIAGRAMA DE PAQUETES	45
FIGURA 21.-	DIAGRAMA DE DESPLIEGUE.....	46
FIGURA 22.-	DIAGRAMA DE IMPLEMENTACIÓN.	47
FIGURA 23.-	DIAGRAMA DE ACTIVIDAD.....	48
FIGURA 24.-	DIAGRAMA DE SECUENCIA.	49
FIGURA 25.-	MODELO ENTIDAD-RELACIÓN (USUARIOS IGU).	50
FIGURA 26.-	MODELO ENTIDAD-RELACIÓN (CREDENCIALES IGU)	50
FIGURA 27.-	VENTANA PRINCIPAL DE INTERFAZ PARA LA GESTIÓN DE USUARIOS IGU.	51
FIGURA 28.-	VENTANA AL INICIALIZAR EL SISTEMA.	52
FIGURA 29.-	VENTANA AL BUSCAR COINCIDENCIA.....	52
FIGURA 30.-	VENTANA AL RECONOCER AL USUARIO.	53
FIGURA 31.-	CARPETAS CON IMÁGENES MUESTREADAS.	58
FIGURA 32.-	RECONOCIMIENTO FACIAL CON VIDEO ENTRANTE.	59
FIGURA 33.-	RECONOCIMIENTO FACIAL POSITIVO.....	59

FIGURA 34.-	RECONOCIMIENTO FACIAL NEGATIVO.	60
FIGURA 35.-	PROTOTIPADO RASPBERRY PI.	60
FIGURA 36.-	SCRIPT PARA MUESTREAR IMÁGENES.	61
FIGURA 37.-	IMÁGENES MUESTREADAS.	62
FIGURA 38.-	RECONOCIMIENTO FACIAL EN RASPBERRY POSITIVO.	63
FIGURA 39.-	RECONOCIMIENTO FACIAL EN RASPBERRY NEGATIVO.	64
FIGURA 40.-	FRENTE CARCASA 3D.	64
FIGURA 41.-	CAJA CARCASA 3D.	65
FIGURA 42.-	TAPA IMPRESA PARTE TRASERA.	66
FIGURA 43.-	TAPA IMPRESA PARTE TRASERA.	66
FIGURA 44.-	CARCASA ARMADA.	67
FIGURA 45.-	ENSAMBLAJE REALIZADO EN SOLIDWORK PARA VERIFICACIÓN DEL MÓDULO PI.	69
FIGURA 46.-	CARCASA IMPRESA DEL MÓDULO PI.	69
FIGURA 47.-	TAPA CORRESPONDIENTE AL MÓDULO PI.	70
FIGURA 48.-	MONTAJE DE COMPONENTES PARA VALIDACIÓN.	70
FIGURA 49.-	PRUEBA DE MONTAJE PROTOTIPO 2.	71
FIGURA 50.-	CÓDIGO PARA ETAPA DE MUESTREO.	72
FIGURA 51.-	CÓDIGO PARA ETAPA DE ENTRENAMIENTO.	74
FIGURA 52.-	CÓDIGO PARA ETAPA DE RECONOCIMIENTO FACIAL.	75
FIGURA 53.-	VENTANA PRINCIPAL DE INTERFAZ PARA LA GESTIÓN DE USUARIOS.	91
FIGURA 54.-	VENTANA CONSULTAR PROFESOR.	92
FIGURA 55.-	VENTANA PROFESOR CONSULTADO.	92
FIGURA 56.-	VENTANA LOG IN.	93
FIGURA 57.-	MENSAJE LOGIN NEGATIVO.	93
FIGURA 58.-	VENTANA ALTA USUARIO.	94
FIGURA 59.-	VENTANA DE CONFIRMACIÓN ALTA USUARIO.	94
FIGURA 60.-	SCRIPT PYTHON PARA MUESTREO DE IMÁGENES.	95
FIGURA 61.-	CARPETA CON IMÁGENES DEL ROSTRO DEL NUEVO USUARIO.	95
FIGURA 62.-	VENTANA MODIFICAR / BORRAR USUARIO.	96
FIGURA 63.-	MONTURA DEL MÓDULO PI.	97
FIGURA 64.-	INICIALIZACIÓN CORRECTA DEL MÓDULO PI.	97
FIGURA 65.-	SISTEMA EN MODO DE ESPERA.	98
FIGURA 66.-	SISTEMA EN MODO DE DETECCIÓN DE ROSTROS EN LA IMAGEN ENTRANTE.	98
FIGURA 67.-	RECONOCIMIENTO FACIAL CON ESTATUS “CONOCIDO”	99
FIGURA 68.-	RECONOCIMIENTO FACIAL CON ESTATUS “DESCONOCIDO”.	99

FIGURA 69.-	ENDPOINT DE PRUEBA PARA REGISTRO ASISTENCIA.	101
FIGURA 70.-	RESPUESTA A LA SOLICITUD DEL MÓDULO PI..	101

Acrónimos y abreviaturas

URI: Uniform Resource Identifier (Identificador uniforme de recursos).

XML: Extensible Markup Language (Lenguaje de marcado extensible).

JPG: Joint Photographic Group (Grupo fotográfico en conjunto).

DPI: Dots Per Inch (Puntos por pulgada).

VNC: Virtual Network Computing (Computación virtual en red).

CUValles: Centro Universitario de los Valles.

CReCE: Centro Regional para la Calidad Empresarial.

IGU: Interfaz de gestión de usuarios

Resumen

Debido a las nuevas tecnologías y al auge del Internet de las Cosas (IoT), la automatización de todo tipo de objetos e infraestructura se va haciendo una realidad y pasa a ser una demanda y/o necesidad de los diversos grupos que conforman la sociedad humana. Por otro lado, el acceso a los espacios como casas, edificios, escuelas, aulas, entre otros, se van automatizando debido al desarrollo de las nuevas tecnologías, así como a la disminución de costos de los dispositivos electrónicos, tales como sensores y actuadores y a las ventajas que esto representa como: comodidad, seguridad, ahorro de tiempo y gestión de uso [1].

Asimismo, con la disponibilidad de plataformas embebidas asequibles para innumerables aplicaciones, es posible desarrollar dispositivos programables de bajo costo (nodos IoT) para adquirir diferentes tipos de datos, los cuales pueden ser utilizados para diversos fines, como es la toma de decisiones; beneficiándose de las aplicaciones del mundo de IoT. Este proceso también está impactando en las Universidades, ya que estas cuentan con grandes infraestructuras, como edificios, laboratorios y aulas. Por lo que la automatización en los accesos de los espacios universitarios permitirá contar con las ventajas antes mencionadas.

La automatización en la gestión de los accesos para las aulas de aprendizaje y el registro de la asistencia del personal docente representan algunas ventajas, para sus usuarios (profesores) así como para los encargados de administrar dichos espacios (jefaturas de departamento, coordinación de personal y servicios generales). En este caso la modernización de ciertos procesos académicos-administrativos que involucran a los profesores, no escapan de este nuevo paradigma de la automatización vía el IoT, ya que a través de la aplicación de estas nuevas tecnologías se podrá automatizar el acceso a las aulas de aprendizaje, a través de reconocimiento facial, lo que conlleva como primer paso la detección del rostro, vía una imagen capturada por la cámara; como segundo paso se compara con las imágenes almacenadas en la base de datos; si es reconocido el rostro entonces procederá a comunicarse con el sistema de información Reloj Checador, el cual verifica si tiene actividad, si es así registra su asistencia y procede a activar la cerradura eléctrica de la puerta.

Abstract

Due to new technologies and the rise of the Internet of Things (IoT), the automation of all kinds of objects and infrastructure is becoming a reality and becomes a demand and/or need of the various groups that make up human society. On the other hand, access to spaces such as houses, buildings, schools, classrooms, among others, are being automated due to the development of new technologies, as well as the reduction in costs of electronic devices, such as sensors and actuators, and the advantages that this represents such as: comfort, safety, time saving and use management [1].

Also, with the availability of affordable embedded platforms for countless applications, it is possible to develop low-cost programmable devices (IoT nodes) to acquire different types of data, which can be used for various purposes, such as decision making; benefiting from the applications of the world of IoT. This process is also having an impact on universities, since they have large infrastructures, such as buildings, laboratories and classrooms. Therefore, the automation in the accesses of the university spaces will allow to have the aforementioned advantages.

The automation in the access management for the learning classrooms and the registration of the attendance of the teaching staff represent some advantages, for their users (teachers) as well as for those in charge of administering said spaces (department heads, coordination of personnel and General services). In this case, the modernization of certain academic-administrative processes that involve teachers, do not escape this new paradigm of automation via the IoT, since through the application of these new technologies access to classrooms can be automated. learning, through facial recognition, which entails face detection as a first step, via an image captured by the camera; as a second step it is compared with the images stored in the database; If the face is recognized, then it will proceed to communicate with the Reloj Checador information system, which verifies if it has activity, if so, it records its attendance and proceeds to activate the electric door lock.

Capítulo 1

Introducción

Capítulo 1. Introducción

Debido a las nuevas tendencias en la tecnología y al auge del Internet de las Cosas (IoT, por sus siglas en inglés), la automatización y gestión de todo tipo de objetos e infraestructuras se va haciendo una realidad y pasa a ser una demanda y/o necesidad de los diversos grupos que conforman la sociedad humana organizada. Desde las formas de comunicarnos, hasta las aplicaciones en las áreas menos pensadas en épocas pasadas, las tecnologías de la información están otorgando un estilo de vida con enormes beneficios como seguridad y confort, a través de cada vez más complejos sistemas digitales utilizados en diferentes actividades.

En el caso de la gestión de accesos a espacios como casas, edificios, lugares de trabajo, escuelas, entre otros, se va automatizando como consecuencia del desarrollo de las TIC (Tecnologías de la información y comunicaciones); así como a la disminución en gran medida de los costos en los diferentes dispositivos electrónicos, tales como sensores, actuadores y a las ventajas que esto representa como: comodidad, seguridad, ahorro de tiempo, gestión de su uso, entre otros [1]. Es menester mencionar cómo ha evolucionado la tecnología referente a los accesos en espacios cerrados; primeramente, a través de las tradicionales puertas con cerraduras y llave, pasando por puertas que activan sus cerraduras por teclados numéricos, RFID y reconocimiento de huellas, hasta llegar a lo que actualmente está en boga, que es la apertura de las puertas con cerraduras eléctricas vía reconocimiento facial [2].

Gracias a la alta disponibilidad de plataformas embebidas asequibles para innumerables aplicaciones, es posible desarrollar dispositivos programables de bajo costo (nodos IoT) para adquirir diferentes tipos de datos, los cuales pueden ser utilizados para diversos fines, como la automatización de diferentes espacios, la toma de decisiones; beneficiándose de las aplicaciones del amplio mundo del IoT.

Por lo que la implementación de un sistema para la gestión del acceso y uso de aulas vía reconocimiento facial, basado en Raspberry Pi, es viable y permite al CUVallés integrarse a este tipo de tecnologías con el objetivo de contribuir a la modernización de la gestión de su infraestructura y de los servicios que se proporciona a su comunidad. Esta propuesta genera una serie de ventajas para sus usuarios como son: el docente que accede al aula a través de la cerradura eléctrica y registra su asistencia al mismo tiempo, así como facilitar la labor del personal de servicios generales en estos procesos de apertura y gestión de aulas.

1.1 Planteamiento del problema

Actualmente la gestión de las actividades en las aulas de aprendizaje se lleva a cabo principalmente por tres instancias: Jefaturas de Departamento, Coordinación de Personal y la Coordinación de Servicios Generales. En el caso de la apertura de las aulas, esta es realizada de manera manual por el área de vigilancia del CUValles de manera rutinaria, la cual abre las aulas a las 8:00 hrs. y las cierra a las 18:00 hrs. En otros casos el interesado solicita la apertura del aula a las autoridades correspondientes. Consecuentemente, si un aula se encuentra cerrada es necesario esperar al personal de vigilancia para su apertura; retrasando el inicio de la sesión presencial. Además de no contar con un registro en tiempo real del uso de las aulas. Bajo este mismo tenor, la problemática que representa al docente requerir transportarse a los distintos módulos de checado para registrar su asistencia y después presentarse al aula correspondiente para impartir su sesión presencial.

Las puertas actualmente cuentan con cerraduras convencionales (llave y perilla manual), lo que implica que el personal indicado porte una serie de llaves físicas para la operación de dichas puertas. Automatizar el acceso de las aulas de aprendizaje permite al personal docente y encargados de éstas, no depender directamente de las autoridades para hacer un mejor uso de las aulas de aprendizaje en lo que al acceso se refiere. El llevar a cabo estas tareas, plantea una serie de desafíos de eficiencia, puntualidad, aceptación y certidumbre para el profesor, así como coadyuvar en los procesos administrativos del CUValles. Este proceso vendría a mejorar la gestión del acceso a ellas, ya que disminuirá el retraso de apertura y al mismo tiempo se registraría la asistencia del profesor al aula.

1.2 Justificación

Los sistemas de autenticación con tecnología biométrica se están convirtiendo en parte integral de sistemas inteligentes complejos [3] ya que presentan cada vez más grados de confiabilidad, por su eficiencia y precisión [4]. La tecnología de reconocimiento facial tiene muchas aplicaciones típicas en el campo de la seguridad pública, la economía civil y el entretenimiento en el hogar [2], [5]. La perspectiva de este desarrollo tecnológico es interesante y se respalda por su demanda en el mercado [6].

Múltiples empresas y universidades actualmente utilizan el registro de asistencia del personal. Tomando la situación actual de los sistemas utilizados para la captura de asistencia, el más utilizado es el reconocimiento por huella dactilar [7]. Sin embargo, la proyección de estos últimos años indica una clara inclinación al reconocimiento facial como mecanismo de autenticación primario [2].

El sistema de reconocimiento facial que se propone pretende coadyuvar a mejorar el sistema actual de control de asistencia y acceso a las aulas del Centro Universitario de los Valles. Disminuir el traslado de los docentes a los módulos de chequeo ubicados en diferentes puntos del centro universitario y automatizar la apertura de las aulas y a la vez registrar su asistencia.

Esta forma planteada de acceso a las aulas vía reconocimiento facial considera toda la información y experiencia actualmente existente del CUValles derivada de sus sistemas en funcionamiento y así, incluir funcionalidades que vuelvan más cómodo el registro y acceso de los profesores a sus aulas.

Para el diseño y la implementación del proyecto, la Raspberry Pi como procesador principal, satisface las necesidades requeridas por sus características técnicas [8] y su relativo bajo costo [9]. Tanto el uso de los distintos periféricos proporcionados por la Raspberry Pi Foundation [10] tales como: el módulo cámara [11] y una pantalla de 3.5 pulgadas [12], así como la implementación de los distintos algoritmos y métodos de la librería OpenCV sustentan la viabilidad de este desarrollo.

De manera intuitiva podemos deducir que a la mayoría de los profesores les será favorable el contar con un sistema de acceso automatizado y de baja intrusión.

1.3 Contribuciones

La contribución fundamental de este desarrollo es la implementación de un sistema que hace uso de las tecnologías del IoT, el reconocimiento facial (RF, por sus siglas en inglés) y la plataforma Raspberry Pi, para ejecutar dos funciones principales. La primera, permite el acceso automatizado del personal docente a las aulas de aprendizaje del CUValles vía reconocimiento facial y la segunda, facilita el registro de la asistencia a su clase; coadyuvando a mejorar este tipo de procesos de la siguiente manera:

- Puede no ser necesario que los docentes tengan que trasladarse a los módulos de checado ubicados en los diferentes puntos del centro universitario para registrar asistencia y luego asistir a su aula correspondiente a impartir su clase.
- El profesor solamente tiene que asistir a su aula correspondiente en tiempo y forma, posicionarse frente al sistema de RF y esperar unos segundos a que el sistema abra la puerta y registre su asistencia, según sea el caso.
- En lo que respecta a las autoridades encargadas de la gestión de las aulas, el sistema automatiza su labor de apertura y disminuye la necesidad del uso del recurso humano; a su vez que se obtiene un mejor control en dicha gestión al registrar cada operación.

1.4 Objetivo general

El objetivo general de este proyecto es diseñar e implementar un sistema automatizado para la gestión del acceso del personal académico a las aulas de aprendizaje del CUValles, utilizando tecnología IoT, reconocimiento facial, y Raspberry Pi, con la finalidad de ceder o negar el acceso al usuario y registrar su asistencia, según sea el caso.

1.4.1 Objetivos particulares

Los objetivos particulares de esta investigación son:

- Diseñar e implementar la base de datos biométrica.
- Diseñar e implementar el hardware del sistema de acceso por reconocimiento facial.
- Implementar los algoritmos encargados de la detección de rostros, entrenamiento del sistema y reconocimiento facial.
- Diseñar e implementar el sistema de activación de la cerradura eléctrica para el acceso.
- Implementar la API-REST para la comunicación entre los sistemas de reconocimiento facial y “Reloj Checador”.

- Diseñar e implementar una interfaz gráfica para la gestión de los usuarios por parte de los administradores del sistema.
- Diseñar e implementar una interfaz gráfica en la pantalla Raspberry, para informar al usuario.

1.5 Alcances y limitaciones

Los alcances y las limitaciones de este proyecto se describen a continuación:

- Solo se registra la asistencia y se abre la cerradura al usuario correspondiente en tiempo y forma. En caso de que un usuario requiera registrar su asistencia en un aula que no corresponde a su clase, no será posible dicha operación.
- Ya que el sistema proporciona dos funcionalidades principales, estas operan de manera conjunta, es decir; no es posible el registro de la asistencia sin la apertura del aula y viceversa.
- En la situación donde algún usuario externo al sistema (intendencia, mantenimiento, profesor emergente), pero importante para el aula, requiera de acceder, no será posible vía reconocimiento facial, sino de manera manual (llave física).
- En caso de una falla o interrupción en el suministro eléctrico, la cerradura funciona, solo con la llave físico. Asimismo, interiormente se puede manipular su apertura de manera manual.

1.6 Estructura del documento

El documento se estructura de la siguiente forma:

- Capítulo 2: En este capítulo, se aborda el marco teórico y el estado del arte de este proyecto, donde primeramente se presentan algunos conceptos relacionados con tecnologías como la visión por computadora y la placa Raspberry Pi. Por otra parte, se profundiza en las diferentes fases de un sistema de reconocimiento facial. Además, se presenta un análisis de algunos trabajos relacionados directamente con el proyecto. De estos se obtienen datos relevantes como los diferentes algoritmos utilizados para las distintas fases del reconocimiento facial, capacidad de procesamiento del hardware en la Raspberry Pi y su funcionamiento en distintas aplicaciones como los son: seguridad, control y gestión de accesos en diferentes lugares.
- Capítulo 3: En esta parte, se describen los diferentes requerimientos del proyecto, desde su

elicitación hasta su redacción, utilizando como referencia el estándar IEEE 29148.

- Capítulo 4: En esta sección se realizó una comparativa entre las principales metodologías utilizadas en la industria y en el ámbito académico; determinando la espiral como la óptima para este proyecto. Igualmente se explica más a profundidad las diferentes fases del ciclo de vida del proyecto en su totalidad; las distintas herramientas y algoritmos utilizados y adaptados para esta tesis.
- Capítulo 5: El diseño se describe en esta parte de la investigación, utilizando diferentes recursos como lo son: el Lenguaje Unificado de Modelado (UML) y algunos estándares IEEE, la ingeniería de requerimientos, el uso de diagramas para definir y explicar el comportamiento del sistema y una explicación detallada de las fases.
- Capítulo 6: El desarrollo e implementación del proyecto se detallan en este capítulo. Los diferentes recursos de hardware y software a utilizar, por ejemplo: los lenguajes de programación, que algoritmos se van a implementar para las distintas etapas del reconocimiento facial, tipo y formato de imágenes para operar el sistema, las distintas librerías de código abierto que llevarán a cabo las tareas requeridas. Por último, la forma de integrar los diferentes recursos de software y hardware para la realización del prototipado en su preparación para la etapa previa a la fase de pruebas.
- Capítulo 7: La etapa de pruebas está descrita en esta sección del manuscrito. Aquí, se especifica la forma de diseñarlas, redactarlas e implementarlas; así como los resultados esperados en sus diferentes rubros como son: pruebas unitarias, de integración, de sistema y de aceptación. Todo esto para cumplir con los requerimientos descritos en los distintos documentos.
- Capítulo 8: Las conclusiones y resultados son descritas en esta parte del documento, después de haber sido ejecutadas las diferentes fases de desarrollo y haber obtenido resultados.

Capítulo 2

Marco teórico y Estado del arte

Capítulo 2. Marco teórico y Estado del arte

En este capítulo se incluye una descripción de los conceptos generales y los elementos de software y hardware en los que se sustenta este trabajo de tesis. Esta sección incluye una revisión de trabajos relacionados con este proyecto para sustentar y justificar el uso de las tecnologías a utilizar. Dichos trabajos revisados permitieron comparar resultados obtenidos por diferentes sistemas de reconocimiento facial, detallando aspectos como rendimiento, tiempo de respuesta, algoritmos, lenguajes y librerías utilizadas. Además, se muestran, algunos productos existentes en el mercado que implementan el reconocimiento facial como método de identificación biométrico, para gestionar accesos en espacios controlados.

2.1 Metodología de desarrollo de software

El desarrollo en espiral es un modelo de ciclo de vida de software definido por primera vez por Barry Boehm en 1986 [13]. Las actividades de este modelo se conforman en una espiral ver (Figura 1), en la que cada bucle o iteración representa un conjunto de actividades. Las actividades no están fijadas a ninguna prioridad, sino que las siguientes se eligen en función del análisis de riesgo, comenzando por el bucle interior [13]. Es por esto por lo que la metodología descrita fue la seleccionada en esta investigación.

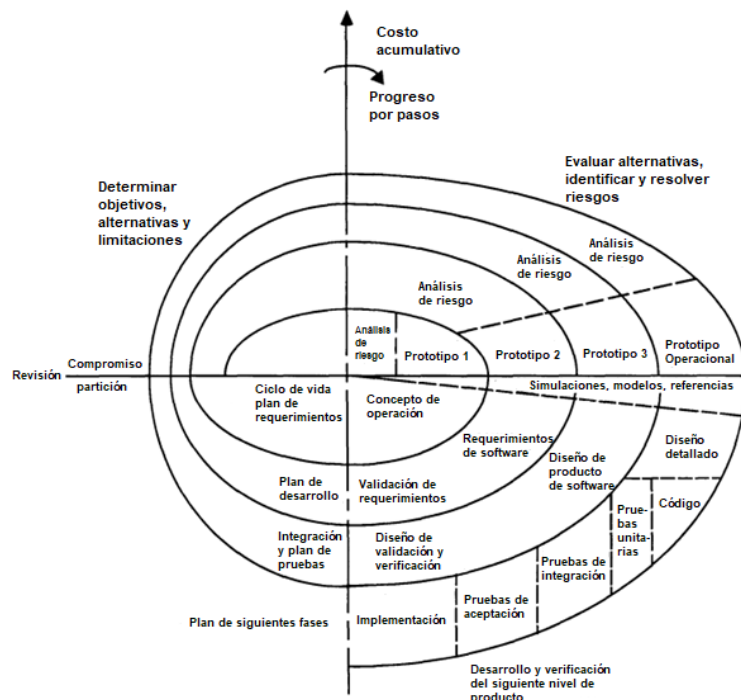


Figura 1.- Modelo en espiral propuesto por Barry Boehm en 1986

2.2 Conceptos generales

En esta sección se definen conceptos generales que son la base para este proyecto de tesis.

Reconocimiento facial: El reconocimiento facial se ha convertido en uno de los más importantes métodos de identificación del usuario. Existe la necesidad de un método de identificación y autenticación de usuarios más rápido y preciso que los usados en los sistemas convencionales [14]. El reconocimiento facial es una de las técnicas de reconocimiento de patrones biométricos más importantes que se utiliza en un amplio espectro de aplicaciones. Los expertos en seguridad han sugerido a la biometría como una forma de mejorar la seguridad en los sistemas. La tendencia comenzó con la huella digital y ahora está cambiando hacia el reconocimiento de caras [14]. En los últimos años, la detección de rostros se ha convertido en una técnica muy popular. La detección de rostros es ampliamente utilizada en la industria publicitaria, entretenimiento, servicios, interfaces de usuario interactivas y codificación de video. El reconocimiento facial es un procedimiento de dos partes, una de ellas es la detección de la cara y la otra, el reconocimiento del usuario como tal. Hay muchas formas de realizar ambas acciones dependiendo de lo que se requiera para el sistema [14].

Raspberry Pi: Desarrollado por la Raspberry Pi Foundation en el Reino Unido, para fines educativos, se trata de un ordenador que utiliza un microprocesador, memoria RAM y tarjeta gráfica; todo integrado en un solo chip. Por lo tanto, se trata de un sistema tipo “System on a chip” (SoC) [15].

Raspbian: Es un sistema operativo gratuito basado en Debian (Linux) optimizado para el hardware Raspberry Pi [10].

Cámara Raspberry Pi: Modulo externo desarrollado por la Raspberry Pi Foundation. Se puede utilizar para tomar videos de alta definición, así como fotografías fijas [11].

IoT: Internet of Things o Internet de las Cosas, hace referencia a una tecnología avanzada en la conexión de objetos cotidianos a internet que intercambian, agregan y procesan información sobre su entorno físico para proporcionar servicios de valor añadido a los usuarios finales [16].

Biometría: La biometría es el estudio de métodos que permiten reconocer a seres humanos basándose en factores genéticos o en determinados rasgos físicos o de conducta [17].

Latencia: El intervalo entre la estimulación y la respuesta en un sistema [18].

Protocolo TCP/IP: El protocolo TCP/IP está formado por un conjunto de protocolos que permiten compartir recursos entre los ordenadores de una red [19].

Lenguaje de programación JAVA: Java es una tecnología que se usa para el desarrollo de aplicaciones que convierten a la Web en un elemento más interesante y útil. Java permite jugar, cargar fotografías, chatear en línea, realizar visitas virtuales y utilizar servicios como: cursos en línea, servicios bancarios en línea y mapas interactivos [20].

Lenguaje de programación Python: Se trata de un lenguaje interpretado o de script, con tipado dinámico, multiplataforma y orientado a objetos [21].

RFID: Acrónimo del inglés (Radio Frequency Identification) donde, a grandes rasgos es una tecnología de comunicación inalámbrica que se utiliza para identificar de forma exclusiva objetos o personas etiquetados [22].

Librería OpenCV: OpenCV (Open Source Computer Vision Library) es una biblioteca de software de visión artificial y aprendizaje automático de código abierto. Ésta se creó para proporcionar una infraestructura común para aplicaciones de visión por computadora y para acelerar el uso de la percepción de la máquina en los productos comerciales. Al ser un producto con licencia BSD, OpenCV facilita que las empresas utilicen y modifiquen el código.

La biblioteca tiene más de 2500 algoritmos optimizados, que incluye un conjunto completo de algoritmos de aprendizaje automático y visión por computadora clásicos y de última generación. Estos algoritmos pueden usarse para detectar y reconocer rostros, identificar objetos, clasificar acciones humanas en videos, rastrear movimientos de cámara, rastrear objetos en movimiento, extraer modelos 3D de objetos, producir nubes de puntos 3D a partir de cámaras estéreo, unir imágenes para producir una imagen de alta resolución de una escena completa, 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. OpenCV tiene una comunidad de más de 47 mil personas y un número estimado de descargas superior a 18 millones. La biblioteca se utiliza ampliamente en empresas, grupos de investigación y organismos gubernamentales [23].

API-Rest: Del inglés Application Programming Interface (API) es la interfaz entre un programa de aplicación y los protocolos de comunicación de un sistema. Debido a que las aplicaciones clientes-servidor se valen de protocolos de transporte para comunicarse, se deben de especificar los detalles sobre el tipo de servicio (cliente o servidor), los datos a transmitir y el receptor donde situar los datos. Un API define un conjunto de procedimientos que puede efectuar una aplicación al interactuar con el protocolo. Por lo general, un API tiene procedimientos para cada operación básica [24].

Representational State Transfer (REST): es un estándar que fija una estructura para que exista compatibilidad entre cualquier lenguaje de programación. Los recursos REST siempre se manipulan a partir de una Uniform Resource Identifier (URI) y todas las peticiones al API deben estar asociadas a uno de los comandos de HTTP. Dentro de una API-REST todo recurso que sea expuesto debe tener una URI única como identificador, los clientes que quieran acceder a estos recursos deberán hacerlo con el método HTTP correspondiente [25] :

- **GET:** para obtener/leer un recurso.
- **POST:** para escribir/crear un nuevo recurso.
- **PUT:** para modificar un recurso, debe mandar toda la información completa del recurso.
- **DELETE:** para borrar un recurso.

2.3 Fases de un sistema de reconocimiento facial

Se pueden definir 4 fases en un sistema de reconocimiento facial [26]: Detección, Preprocesado, Extracción de características y Comparación y decisión como se observa en la (Figura 2).

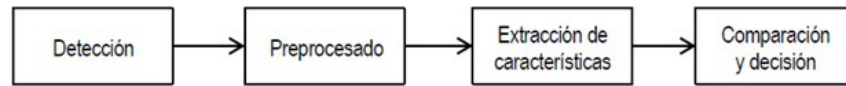


Figura 2.- Diagrama general de un sistema de reconocimiento facial.

Un sistema de reconocimiento facial tiene además una etapa previa, que se conoce como entrenamiento, y se ejecuta una sola vez. En esta etapa ocurren las fases de detección, preprocesado y extracción de características indicadas en la (Figura 3) [26].



Figura 3.- Diagrama etapa entrenamiento de un sistema de reconocimiento facial.

2.3.1 Detección

La fase de detección consiste en detectar, en una imagen, un rostro humano y aislarlo del resto. Existen diferentes algoritmos encargados de realizar dicha función, unos más complejos que otros. De los algoritmos más utilizados, debido a su bajo coste computacional y su enfoque para la detección de objetos, es el Viola-Jones [27]. Este algoritmo minimiza el tiempo de cálculo y logra una alta precisión de detección [22]. El algoritmo Viola-Jones básicamente está basado en clasificadores llamados Haar-like features [28], los cuales a grandes rasgos funcionan como una agrupación de características, en forma de cascada, a través de un algoritmo basado en AdaBoost [29].

2.3.2 Preprocesado

En esta fase de preprocesado se lleva a cabo un proceso de transformación de la imagen, en modelos geométricos y la prepara para la extracción de características. En este punto se utilizan cuatro diferentes etapas [26]:

- **Rotación:** Se detectan los ojos, para determinar el ángulo de giro en una imagen y

compensarlo.

- **Escalado:** Se utiliza para obtener las imágenes del mismo tamaño, variando la distancia entre los ojos (en la imagen).
- **Recorte:** Se obtiene en un nuevo archivo una imagen recortada del rostro, de la imagen original.
- **Ecualización del histograma:** Se normaliza la variabilidad de los píxeles, ya que, al momento de capturar la imagen, la luz se concentra en puntos diferentes de cada uno. Esto ayuda a obtener una imagen más homogénea [26].

2.3.3 Extracción de características

Aquí se emplean diferentes algoritmos para obtener información relevante de la imagen, con el objetivo de poder realizar la comparación satisfactoriamente. Existen dos grandes grupos en las técnicas de reconocimiento facial ver (Figura 4); las dependientes de pose y las independientes de pose (ya que requieren hardware específico y complejo no se abordará el segundo grupo). Las primeras se subdividen en dos áreas: la técnica basada en apariencias, donde se analizan la textura de la imagen a partir de la cual se aplican diferentes técnicas estadísticas y se extrae la información y las segundas, basadas en modelos; su función principal es extraer las características de la forma y textura del rostro [26].

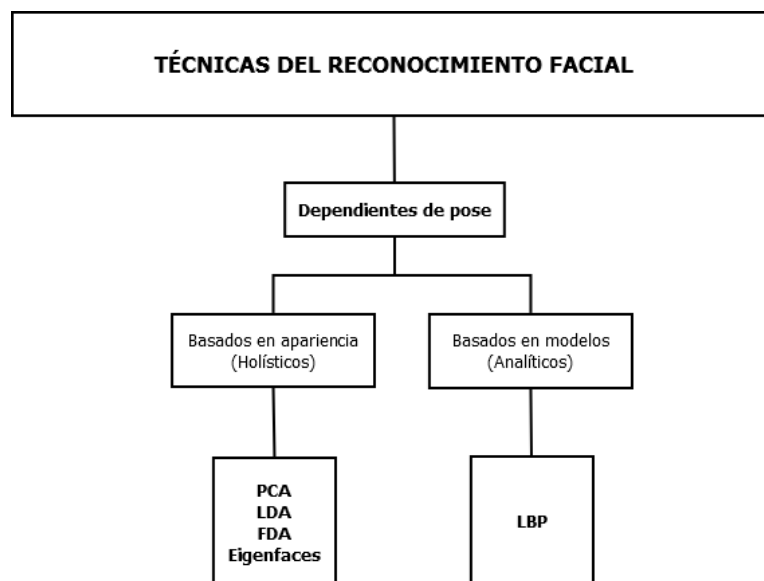


Figura 4.- Taxonomía del reconocimiento facial utilizada en esta investigación

2.3.4 Técnicas basadas en apariencias

Las técnicas basadas en apariencias se centran en la apariencia y transforman la imagen en un espacio para poder aplicar métodos estadísticos. Las ventajas principales de esto es que se puede operar con imágenes de baja calidad en su resolución, presenta una relativa rapidez en lo referente a la ejecución (funciona muy bien en tiempo real). La desventaja de estas técnicas es, sin duda, que requiere una gran cantidad de imágenes, en la etapa de entrenamiento y factores externos como la iluminación, la pose o la expresión facial, repercuten directamente en el resultado final, en mayor o menor grado [26].

Algunos algoritmos que utilizan estas técnicas son:

PCA: del inglés Principal Component Analysis, se basa en las propiedades estadísticas de las imágenes y es un método óptimo para reducir el número de dimensiones necesarias para representar un conjunto de vectores [26].

LDA: del inglés Linear Discriminant Analysis, este método es una técnica de aprendizaje supervisado de aproximación estadística para clasificar datos, con base en muestras de grupos desconocidos [30].

FDA: del inglés Frequenty Doimain Analysis, estas técnicas como su nombre lo dice, funcionan con el análisis en el dominio de la frecuencia; ofrecen una representación de la imagen en función de los componentes de baja frecuencia que presentan alta energía [26].

Eigenfaces: Basado en la técnica PCA, este algoritmo transforma la imagen a un subespacio, haciendo posible obtener vectores de características de menor dimensionalidad sin una pérdida de información discriminativa importante [26].

2.3.5 Comparación y decisión

En la fase de comparación y decisión, se compara la información obtenida de la imagen entrante, con la almacenada en la base de datos a través de diferentes métodos, divididos en medidas de similitud y clasificadores tales como: distancia Euclídea (comparación más básica entre dos puntos), distancia Bhattacharyya (cuantifica la similitud entre dos distribuciones de probabilidad), clasificador K-Nearest Neighbours (técnica no paramétrica de clasificación de objetos) y clasificador Support Vector Machines (modelo de aprendizaje que encuentra distancia máxima entre dos planos) [26].

2.4 Algoritmo para el reconocimiento facial

Para la tarea más importante del sistema se requiere de un algoritmo básico, pero muy poderoso, encargado de realizar el complejo proceso de detectar un rostro. En esta sección se aborda de forma teórica cómo funciona y cuál es su aplicación para esta investigación.

2.4.1 Detección de rostros

2.4.1.1 Haar Cascade

La detección de objetos mediante clasificadores en cascada basados en características de Haar es un método eficaz de detección de objetos propuesto por Paul Viola y Michael Jones en su artículo Rapid object detection using a boosted cascade of simple features [27]. Es un enfoque basado en el aprendizaje automático en el que una función de cascada se entrena a partir de muchas imágenes positivas (imágenes con el objeto de interés) e imágenes negativas (imágenes sin el objeto de interés). Luego se usa para detectar los objetos que se estudian en otras imágenes.

En este trabajo de tesis se usa el algoritmo de Haar Cascade para la detección de rostros. Inicialmente, el algoritmo necesita muchas imágenes positivas e imágenes negativas para entrenar al clasificador y extraer características. En este caso, se utilizan las máscaras de Haar que se muestran en la (Figura 5). Cada máscara o kernel detecta características de los niveles de gris en toda la imagen, calculando un valor único para cada píxel.

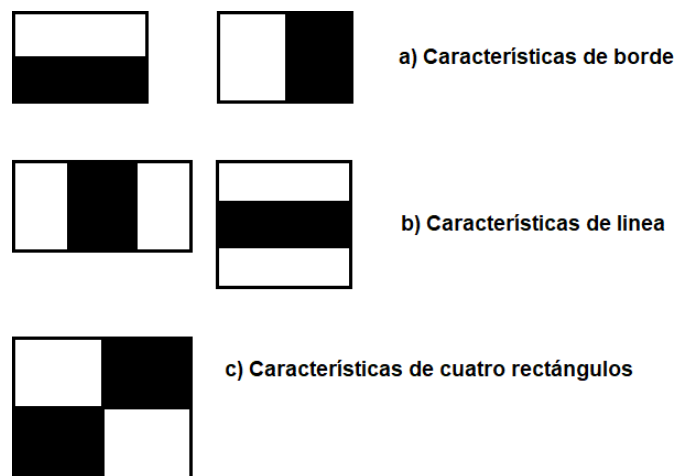


Figura 5.- Diferentes máscaras (kernel) de Haar.

No todas las características encontradas pueden ser relevantes. Consideremos la imagen de la (Figura 6), donde la fila superior muestra la extracción de dos características. La primera característica para extraer se centra en la propiedad de que la región de los ojos suele ser más oscura que la región de la nariz y las mejillas. La segunda característica, considera que los ojos son más oscuros que el puente de la nariz. Si aplica la misma máscara a la zona de la imagen donde se ubican las mejillas u otras partes no se obtiene información relevante de contraste. En este punto interviene AdaBoost, que permite refinar la extracción de características.



Figura 6.- Características relevantes en Haar Cascade.

2.4.2 Algoritmos de entrenamiento

2.4.2.1 PCA

El análisis de componentes principales (PCA) fue propuesto independientemente por Karl Pearson (1901) y Harold Hotelling (1933) para convertir un conjunto de variables posiblemente correlacionadas en un conjunto más pequeño de variables no correlacionadas. La idea es que un conjunto de datos de alta dimensión a menudo se describe mediante variables correlacionadas y, por lo tanto, solo unas pocas dimensiones significativas representan la mayor parte de la información. El método PCA encuentra las direcciones con la mayor variación en los datos, llamadas componentes principales [31].

2.4.2.1.1 Eigenfaces

El problema con la representación de la imagen que se nos da es su alta dimensionalidad. Las imágenes bidimensionales $p \times q$ en escala de grises, abarcan un espacio $m = p \times q$ -espacio vectorial dimensional, por lo que una imagen con 100×100 píxeles se encuentran en un 10,000-espacio de imagen dimensional. Ya que no todas las dimensiones son útiles para nosotros, solo podemos tomar una decisión si hay alguna variación en los datos, por lo que se buscan los componentes que representan la mayor parte de la información con el análisis de componentes principales (PCA) [31].

2.4.2.1.1.1 Descripción algorítmica del método Eigenfaces

Sea $X = x_1, x_2, \dots, x_n$ un vector aleatorio con observaciones $X_i \in \mathbb{R}^D$

1. Cálculo de la media μ :

$$\mu = \frac{1}{n} \sum_{i=1}^n X_i$$

2. Cálculo de la matriz de covarianza S :

$$S = \frac{1}{n} \sum_{i=1}^n (X_i - \mu)(X_i - \mu)^T$$

3. Cálculo de los eigenvalores λ_i y eigenvectores propios V_i de S :

$$SV_i = \lambda_i V_i, i = 1, 2, \dots, n$$

4. Orden de los vectores propios descendiendo por su valor propio. Los componentes principales k

son los autovectores correspondientes a la k valores propios más grandes.

Los componentes principales k del vector observado X , luego son dados por:

$$y = W^T(x - \mu)$$

Donde $W = (V_1, V_2, \dots, V_k)$

La reconstrucción a partir de la base PCA viene dada por:

$$X = Wy + \mu$$

donde $W = (V_1, V_2, \dots, V_k)$

El método Eigenfaces luego realiza el reconocimiento facial mediante las siguientes operaciones:

- Proyección de todas las muestras de entrenamiento en el subespacio de PCA.
- Proyección de la imagen de la consulta en el subespacio PCA.
- Búsqueda del vecino más cercano entre las imágenes de entrenamiento proyectadas y la imagen de consulta proyectada [32].

2.4.2.2 Fisherfaces

El algoritmo Fisherfaces aprovecha el hecho de que, bajo condiciones reconocidamente idealizadas, la variación dentro de la clase se encuentra en un subespacio lineal del espacio imagen. Por lo tanto, las clases son convexas y, por lo tanto, linealmente separables. Se puede realizar una reducción de la dimensionalidad utilizando una proyección lineal y aun así conservar la separabilidad lineal. Este es un fuerte argumento a favor del uso de métodos lineales para la reducción de la dimensionalidad en el problema de reconocimiento facial, al menos cuando se busca la insensibilidad a las condiciones de iluminación [33].

Dado que el conjunto de aprendizaje está etiquetado, tiene sentido usar esta información para construir un método más confiable para reducir la dimensionalidad del espacio de características. Aquí argumentamos que, al usar métodos lineales específicos de clase para la reducción de la dimensionalidad y clasificadores simples en el espacio de características reducido, se pueden obtener mejores tasas de reconocimiento que con el método Eigenfaces. El Discriminante Lineal de Fisher (FLD) [34] es un ejemplo de un método específico de clase, en el sentido de que trata de “dar forma” a la dispersión para hacerla más confiable para la clasificación [33].

Deje que la matriz de dispersión entre clases se defina como:

$$S_B = \sum_{i=1}^c N_i (\mu_i - \mu)(\mu_i - \mu)^T$$

y la matriz de dispersión dentro de la clase se define como:

$$S_W = \sum_{i=1}^c \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T$$

donde μ_i es la imagen media de la clase X_i y N_i es el número de muestras en la clase X_i [33].

2.4.2.3 Local Binary Pattern Histograms

Este algoritmo básicamente es una técnica eficiente para el análisis de texturas en escala de grises. La idea principal del algoritmo se centra en dividir la imagen en pequeñas regiones cuadradas de 3 x 3 píxeles. En cada una de estas regiones se aplica el análisis mediante el uso del Local Binary Pattern, el cual consiste en comparar cada píxel central con los que le rodean, de manera que, si el valor central es mayor o igual al vecino, se le asigna un valor binario de “0”, mientras que si es menor se le asigna un “1”, tal como lo muestra la (Figura 7). Con estos valores binarios se obtiene un número (orden según las agujas de un reloj) que, pasado a decimal, se convierte en el descriptor LBP de cada uno de los píxeles de una imagen, con los cuales se obtiene la imagen LBP que sirve para acentuar los rasgos faciales

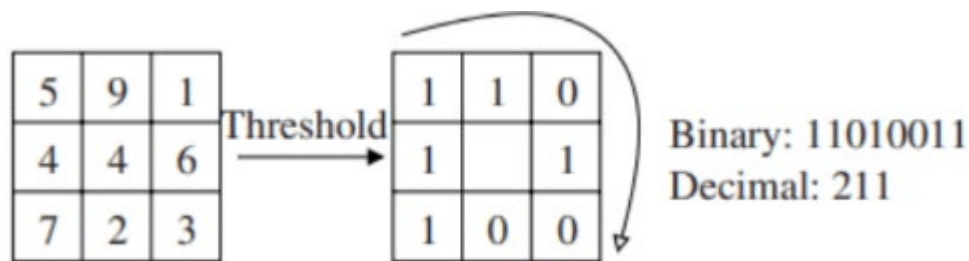


Figura 7.- Ejemplo de aplicación del operador LBP [35].

Así pues, todos los píxeles con valor 1 como binario, marcan una zona de contraste; permitiendo la distinción de puntos, bordes e incluso zonas planas en la imagen, como se observa en la (Figura 8).

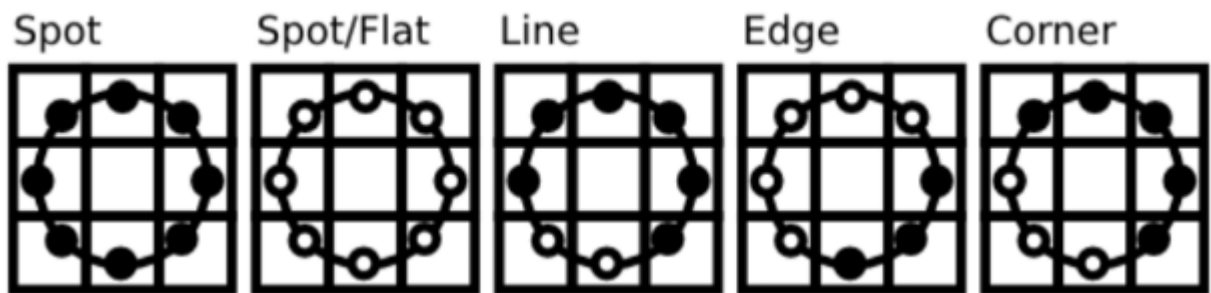


Figura 8.- Definición de zonas por el operador LBP [36].

2.5 Estado del arte

En esta sección se presenta un análisis del estado del arte en los siguiente dos rubros: el reconocimiento facial y la gestión de accesos; todo esto diseñado e implementado en la Raspberry Pi como procesador central. Como aspectos fundamentales para la revisión se consideraron los siguientes tópicos: algoritmos, métodos, librerías implementadas, metodología y hardware utilizados, así como la arquitectura y el diseño de diferentes sistemas de reconocimiento facial. Todo esto para logra una gestión en espacios cerrados.

2.5.1 Trabajos relacionados

El uso del reconocimiento facial como mecanismo biométrico de autenticación, está siendo una realidad y cada vez más situaciones o áreas de la vida cotidiana están migrando a esta forma de identificación [2]. En este sentido se investigaron algunos artículos que presentan alta relevancia en lo referente a la administración de los accesos en lugares cerrados, por medio del reconocimiento facial y la Raspberry Pi.

Partiendo de esta visión es posible diseñar e implementar sistemas para gestionar accesos en espacios controlados, con el uso de herramientas de software libre. Tal como lo muestran en [37] donde implementaron un sistema de seguridad inteligente para el hogar. Utilizando la librería OpenCV, el lenguaje Python y el algoritmo LBP para procesamiento de imágenes, lograron obtener un 80 % de precisión en pruebas utilizando imágenes en tiempo real. Además, utiliza mecanismos de defensa en caso de intrusión, como el envío de SMS y correo electrónico. Estos resultados y funcionalidades permiten tener una gestión segura y controlada del entorno a administrar.

Existe una relación al momento de implementar los algoritmos a utilizar para obtener los resultados esperados. Viola-Jones, Haar Cascade, LBP, Eigenfaces y Fisherfaces son los más utilizados, como lo muestran los trabajos [38], [39]. En el primero obtuvieron una tasa de reconocimiento con un 95 % de precisión y un tiempo inferior a 0.5 segundos, en condiciones con luz artificial y en el segundo, una tasa de reconocimientos del 99.999 % en ambientes estáticos, mencionando limitantes mínimas de velocidad al procesador gráfico de la Raspberry.

Todo sistema que utilice el reconocimiento facial como método biométrico, está sujeto a factores externos que pueden alterar el resultado final esperado, como son: luz visible, accesorios faciales como gafas y maquillaje y el uso de barba por parte de los usuarios, antes y/o después de haberse dado de alta en el sistema. Para solucionar estos conflictos, los investigadores en [40] elaboraron un sistema que logró una tasa de precisión del 85,7 % en su fase experimental con imágenes con diferentes distorsiones, por ejemplo: cara girada, con anteojos o no nítida. Todo esto utilizando los algoritmos de reconocimiento facial anteriormente mencionados, y complementando con técnicas de redes neuronales.

Una forma de volver eficiente el sistema y aprovechar al máximo el hardware es implementando la programación paralela, es decir; volver al sistema multihilos por medio del paradigma cliente-servidor,

como lo hicieron en [41] donde apoyándose en bases de datos preestablecidas, lograron obtener un crecimiento exponencial en el rendimiento y tiempo de detección al implementar los hilos de la siguiente manera: se obtienen las características de la imagen entrante y se envían a través de la red, creando peticiones múltiples al servidor para poder encontrar una coincidencia más rápidamente.

La robustez en los sistemas de identificación por reconocimiento facial puede ser analizada desde diferentes ángulos, ya que las múltiples causas de fallos, presentes en todo sistema, se ven disminuidas al agregar mecanismos extras de identificación, como son: reconocimiento por huella dactilar, lector RFID [22] y el uso de contraseñas, tal como lo hicieron en [42], donde los investigadores proponen un sistema de seguridad de alto nivel, donde implementan, además del reconocimiento facial, otras formas para acceder al sistema. En este caso agregaron, en orden jerárquico, los siguientes módulos: un teclado numérico para el ingreso de contraseñas, un escáner de huella dactilar y un lector de RFID; ocasionando un control exhaustivo en el acceso al lugar a gestionar.

La cantidad de imágenes es relevante en estos sistemas como lo demuestran los investigadores del siguiente trabajo titulado *Embedded Video Processing on RaspberryPi* [43], donde utilizaron la Raspberry Pi como procesador y comparador de imágenes en tiempo real, por medio de un clasificador LBP, llegando a la conclusión de que “la cantidad de imágenes positivas utilizadas para el proceso de entrenamiento no fue suficiente, pero el clasificador podría mejorarse agregando más muestras”.

Otro ejemplo interesante en la implementación de Raspberry Pi, su módulo cámara y el lenguaje Python con algoritmos de reconocimiento facial, es el siguiente trabajo [44]. Aquí los desarrolladores aplicaron un sistema antirrobo en vehículos en tiempo real. Además de utilizar las ventajas del IoT, al notificar con un mensaje de texto al propietario del vehículo cualquier intento de robo y almacenando la foto del invasor en una base de datos, obteniendo resultados loables.

Al priorizar la seguridad como parte importante de la vida cotidiana, estos investigadores de origen asiático crearon un sistema de puerta segura [45], implementando una combinación de cámara web, Raspberry Pi, un relevador, una cerradura de puerta solenoide y un módulo GSM. Por medio de algoritmos, codificados en lenguaje Python y su librería OpenCV, como el clasificador Haar Cascade para detección facial y el histograma de patrón binario local (LBPH) para reconocimiento del usuario, obtuvieron resultados satisfactorios tras varias operaciones en su etapa de pruebas, utilizando reconocimiento facial, uso de contraseña y mensaje de alerta (módulo GSM) como mecanismos de seguridad.

Con un enfoque en la mejora del registro de asistencia, para reducir los inconvenientes entre el tiempo y el esfuerzo en su captura, el siguiente artículo [46] presenta una aplicación bastante interesante del reconocimiento facial en un aula de clases a través de un sistema de asistencia automática basada en la combinación de tecnología de visión computarizada e interacción con el portal académico existente. Este trabajo ha sido probado con proyección a gran escala en más de 2200 estudiantes universitarios de primer año; arrojando resultados numéricos que demuestran la eficacia entre precisión y rendimiento como primer prototipo funcional.

En el siguiente trabajo [47] se experimenta con una hibridación de algoritmos mediante la aplicación de algunos modelos de reconocimiento facial clásicos, pero más rápidos, a saber, Eigenfaces, Fisherfaces, Local Binary Patterns (LBP) y el modelo de red neuronal convolucional (CNN) de última generación.

La técnica de hibridación de modelo que propusieron se basa en diferentes niveles realizados por partición de rostros; logrando un rendimiento aproximadamente de 33,43 % más rápido que los convencionales, pero manteniendo la misma precisión de su propio conjunto de datos de rostros de un salón de clases de 15 estudiantes. Todo esto en tiempo real mientras se desarrollaba una clase.

Utilizando técnicas convencionales de detección y reconocimiento de rostros, como la detección Haar y PCA, este artículo llevó el reconocimiento facial a un nivel en el que el prototipo pueda reemplazar el uso de contraseñas y tarjetas RFID para acceder a sistemas y edificios de alta seguridad. Con el uso del kit Raspberry Pi, lograron hacer que el sistema sea rentable, fácil de usar y con un relativo alto rendimiento [48] .

Para demostrar el potencial de las unidades gráficas y debido al crecimiento en el uso de los teléfonos móviles, la siguiente investigación considera el problema del reconocimiento facial y propone un sistema que aprovecha las GPU integradas para permitir el reconocimiento facial a gran escala. El sistema proporciona una novedosa implementación paralela del reconocedor Fisherface que permite un número configurable y escalable de subprocesos paralelos. El sistema también aprovecha el gran almacenamiento secundario disponible para escalar aún más la base de datos de reconocimiento. Una implementación de prototipo inicial permite un tamaño de base de datos de hasta 1024 rostros con un tiempo de reconocimiento de 210 ms en el sistema iPhone5. La GPU dio como resultado una mejora de casi 10 veces en el rendimiento y una reducción de 7 veces en la potencia [49].

2.5.2 Productos comerciales

El reemplazo de las cerraduras convencionales por cerraduras de puerta inteligente se va volviendo una realidad ya que estas últimas se están convirtiendo en uno de los dispositivos de consumo digital más populares debido a la comodidad del usuario y al precio asequible [50].

En el mercado existen diferentes dispositivos que avalan la alta confiabilidad del uso de biometría para el diario vivir. Un caso interesante es el siguiente: USE Smart Lock, (Figura 9). Este es un sistema de cerradura inteligentes con reconocimiento facial, escáner de huellas y otras funcionalidades extras. USE Smart Lock mantiene la cerradura con llave tradicional y todo el sistema inteligente se encuentra embebido en ella [51].



Figura 9.- Dispositivo USE Smart Lock

Otro tipo de cerradura que presenta el reconocimiento facial como mecanismo de gestión de accesos es la siguiente [52]. Esta al igual que la anterior, presenta las siguientes ventajas (Figura 10):

- Múltiples formas de desbloqueo (6 modos): reconocimiento facial, huella, llave, clave, aplicación, tarjeta y una aplicación móvil propia
- Reconocimiento facial mejorado: mayor precisión incluso con accesorios como anteojos.
- Capacidad de reconocimiento facial múltiple: Hasta 100 usuarios.



Figura 10.- Cerradura Wifi- Reconocimiento Facial con 6 Modos De Apertura

Capítulo 3

Requerimientos

Capítulo 3. Requerimientos

En este capítulo, se describen los diversos tipos de requerimientos para este proyecto, utilizando los estándares IEEE29148 [53] y el Código de Ética y Práctica Profesional de Ingeniería de Software 5.2 [54] previamente consultados para satisfacer las necesidades de los stakeholders y así realizar un producto final con todas las funcionalidades y restricciones especificadas en la documentación. Además, los requerimientos sirven de guía para un correcto diseño y aplicación en sus diferentes partes del ciclo de vida del software.

3.1 Funciones del producto de software

Las funciones principales del sistema son las siguientes:

- Inicialización por aproximación: Cuando el usuario se acerca, el sistema se activa por medio del sensor de proximidad.
- Detección de rostro: El sistema pretende detectar un rostro en la imagen de entrada.
- Análisis de imagen: En caso de encontrar un rostro, se extraen sus características.
- Reconocimiento facial: Se detectan coincidencias entre la imagen entrante y las imágenes entrenadas en el sistema.
- Registro de asistencia: Solo en caso de ser reconocido, el sistema registra la asistencia del usuario, comunicándose con el sistema Reloj Checador a través de la API-REST.
- Acceso al aula: El sistema otorga o niega el acceso al aula a través de una cerradura electromecánica, según sea el caso.
- Información a usuario: El sistema muestra en la pantalla el estatus del usuario.

3.2 Requerimientos del banco de imágenes

Se necesita un banco de imágenes de los usuarios del sistema gestionado por las autoridades competentes. Dichas imágenes son obtenidas por medio de la interfaz gráfica construida en Java la cual genera un banco de imágenes de los docentes y través de un script en Python se procesan las imágenes para la creación del respectivo modelo. Las imágenes son capturadas y almacenadas con las siguientes características (Tabla 1):

Tabla 1.- Características de las imágenes almacenadas

Características de las imágenes almacenadas	
Formato	jpg
Tamaño en disco	12.0 KB (por unidad)
Dimensión	150 x 150 pixeles
Resolución	96 x 96 dpi

Todas las imágenes creadas se almacenan y clasifican en diferentes carpetas generadas por el script previamente, creadas con el nombre y código del usuario, separados por un guion bajo. La (Figura 11) ejemplifica esta instrucción.



Figura 11.- Forma de almacenar las imágenes en las carpetas.

Así pues, la computadora principal para la gestión del banco de imágenes almacena toda la información requerida para el sistema. En todo caso se requiere leer o escribir en dicho banco solamente cuando exista un alta/baja de un usuario.

3.3 Requerimiento del modelo de entrenamiento

- El sistema requiere un banco de imágenes.
- El sistema realiza un proceso de reconocimiento detección de rostros.
- El sistema realiza el proceso de reconocimiento facial en las imágenes.
- El sistema requiere, almacenar las características del modelo generado en un archivo xml.

3.4 Requerimientos del dispositivo de RF

Por último, para la etapa de reconocimiento facial a través del dispositivo, se requiere:

- Obtener una imagen de entrada a través de la cámara.
- Realizar un proceso de reconocimiento detección de rostros.
- Realizar un proceso de reconocimiento facial.
- Checar asistencia
- Abrir puerta

3.5 Restricciones de diseño del prototipo físico

El sistema debe ser diseñado e implementado en una Raspberry Pi 4, con su módulo cámara, alimentación y conexión a la cerradura. Todo esto respetando las medidas que se indican en la (Figura 12).

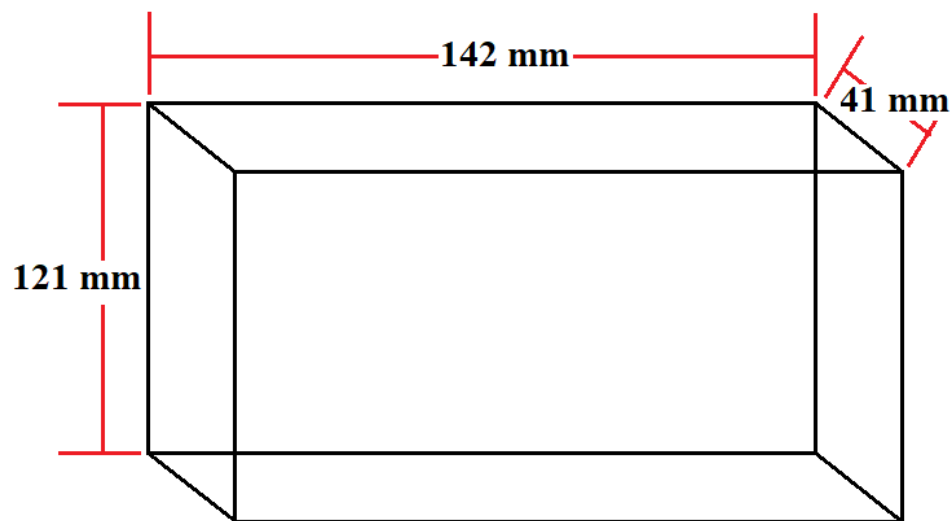


Figura 12.- Dimensiones físicas del sistema

3.5.1 Elicitación de requerimientos

La elicitación de requerimientos es la actividad que se considera como el primer paso en un proceso de ingeniería de requisitos. Debido a que existen muchas técnicas disponibles para elicitar requisitos, es necesario contar con una metodología que sirva de guía para su aplicación, teniendo en cuenta que, cada método tiene fortalezas y debilidades y que además está orientado hacia un dominio específico. Para la elicitación de los requerimientos del sistema, se utilizaron dos técnicas propuestas por Kendall y Kendall [55] que van acorde al proyecto. Estas técnicas se explican en lo adelante.

3.5.1.1 Técnica intrusiva entrevista

Se obtienen los requerimientos, a través de entrevistar a los stakeholders, siguiendo los 5 pasos propuestos en [48], los cuales son:

- Leer antecedentes materiales.
- Establecer los objetivos de la entrevista.
- Decidir a quien se va a entrevistar.
- Preparar la entrevista.
- Decidir el tipo y estructura de las preguntas.

3.5.1.2 Técnica no intrusiva escenarios

Para la elicitación correcta de los requerimientos utilizando esta técnica es necesario pensar en uno o varios escenarios que ayuden a explicar el sistema como si ya estuviese funcionando. Esto se logra a través de la explicación concreta del sistema por parte del director de tesis y un análisis profundo de dicha explicación llevada a cabo por el elicitor para determinar la naturaleza de cada requerimiento encapsulado en dicha explicación [55].

3.6 Especificación de requerimientos (IEEE29148)

Para llevar a cabo la especificación de requerimientos se utiliza como base el estándar IEEE29148 [53]. Aquí se describen y clasifican, a través de diferentes rubros, la forma de redactar todos los requerimientos obtenidos en la elicitación, para facilitar su descripción y manejo en el desarrollo del sistema.

El estándar IEEE29148 sugiere la creación de un documento llamado (*Software Requirement Specifications*) o SRS, el cual contiene el formato sugerido para realizar la correcta documentación de los requerimientos. Este formato se explica a continuación (Tabla 2):

Tabla 2.- Formato de redacción de requerimientos.

ID	Descripción detallada del requerimiento y si existe alguna dependencia con algún otro requisito y resultado esperado.
Título del requerimiento	

3.6.1 Listado general de requerimientos

Los requerimientos obtenidos con las técnicas antes mencionadas se citan a continuación, clasificados según el estándar IEEE29148 [53]. Aquí solo se mencionan de manera general, como se observa en la (Tabla 3) (Véase apéndice C para más información).

Tabla 3.- Listado de requerimientos tabulados.

ID	Tipo	Descripción
SRF-01	Funcional	El sistema tiene 4 posibles estados: apagado, encendido, activo y pasivo.
SRF-02	Funcional	El sistema debe entrar en modo encendido durante la jornada laboral de 07:45 hrs a 18:30 hrs.
SRF-03	Funcional	El sistema debe pasar a modo activo cuando el usuario se acerca, por medio del sensor de proximidad.
SRF-04	Funcional	El sistema debe entrar en modo pasivo pasado 30 segundos de inactividad (no presencia de usuario).
SRF-05	Funcional	El sistema intenta detectar un rostro en la imagen entrante.
SRF-06	Funcional	El sistema debe reconocer al usuario a través de su rostro.
SRF-07	Funcional	El sistema registra o no la asistencia del usuario, según sea el caso.
SRF-08	Funcional	El sistema decide si abre o no la puerta según sea el caso.
SRF-09	Funcional	El sistema comunica su situación al usuario.
SRNF-01	No funcional	Se detecta la presencia de componentes electrónicos previo al encendido.
SRNF-02	No funcional	Uso de librería OpenCV en scripts.
SRNF-03	No funcional	Python como lenguaje principal del proyecto.

SRR-01	Rendimiento	88 % de efectividad en el reconocimiento facial.
SRR-02	Rendimiento	El sistema debe operar en condiciones externas.
SRR-03	Rendimiento	5 segundos como tiempo de respuesta máximo.
SRR-04	Rendimiento	Falsos positivos debe ser menor al 12%.
SyU-01	Usabilidad	Usuario solo requiere acercar su rostro al sistema.
SyU-02	Usabilidad	El dispositivo solo requiere una imagen del usuario.
SyI-01	Interfaz	El dispositivo requiere de una interfaz gráfica en prototipo físico
SyI-02	Interfaz	Interfaz gráfica para administración de los usuarios.
SyH-02	Hardware	El dispositivo requiere como centro de procesamiento central del prototipo una Raspberry Pi 4.
SyH-02	Hardware	El sistema requiere de un sensor de presencia PIR HC-SR505 para operar el requerimiento SRF-03.
SyH-03	Hardware	El sistema requiere de un relevador para operación del requerimiento SyH-04.
SyH-04	Hardware	El sistema requiere de una cerradura electromecánica para operar el requerimiento SRF-08.
SyH-05	Hardware	El sistema requiere de un módulo cámara Raspberry Pi para la satisfacción del requerimiento SRF-05.
SyH-06	Hardware	Raspberry Pi Touch Display 3.5 pulgadas para la operación y satisfacción de los requerimientos SRF-09, SyI-01 y SyI-02.

3.6.1.1 Requerimientos funcionales

Redacción de los requerimientos funcionales según el estándar IEEE29128.

SRF-01	De manera general, el sistema debe estar en 4 posibles estados: apagado, encendido, activo y pasivo.
Estados del sistema	
SRF-02	El sistema debe entrar en modo encendido durante la jornada laboral de 07:45 hrs a 18:30 hrs.
Modo encendido	
SRF-03	El sistema debe entrar en modo activo cuando el usuario se acerca, por medio del sensor de proximidad.
Modo activo	
SRF-04	El sistema debe entrar en modo pasivo pasado 30 segundos de inactividad (no presencia de usuario).
Modo pasivo	
SRF-05	El sistema debe detectar un rostro en la imagen entrante.
Detección de rostro	

SRF-06	Reconocimiento del usuario	El sistema debe reconocer al usuario con su rostro.
SRF-07	Registro de asistencia	El sistema registra o no la asistencia del usuario, según sea el caso.
SRF-08	Decisión acceso al aula	El sistema decide si abre o no la puerta según sea el caso.
SRF-07	Aviso de efectividad	El sistema comunica su situación al usuario.

3.6.1.2 Atributos de software del sistema

Los atributos de software del sistema son:

- **Confiabilidad:** El sistema debe abrir la cerradura de la puerta, si y solo si el usuario es encontrado en el sistema y tiene actividad en dicha aula en tiempo y forma.
- **Disponibilidad:** El sistema debe estar disponible en el horario escolar establecido por el centro universitario; quedando fuera de servicio en horario diferente, excepto en condiciones extraordinarias.
- **Seguridad:** Los resultados obtenidos indican que solo se permita la entrada a usuarios reconocidos. Asimismo, el sistema opera con un archivo *xml* con toda la información procesada por OpenCV. En caso de ser sustraído no es posible obtener su información, al no conocer los demás parámetros de operación.
- **Portabilidad:** En este rubro, el sistema se caracteriza por utilizar hardware y software libre; volviéndose portable ya que puede ser replicado, es escalable y se puede reemplazar cualquier modulo con relativa facilidad.

Capítulo 4

Metodología

Capítulo 4. Metodología

En este capítulo se describe la metodología que fue implementada para el desarrollo de este proyecto, además de hacer una descripción general de las distintas fases planteadas. También, contiene un análisis de los algoritmos a utilizar en las distintas fases del proceso de detección y reconocimiento facial.

4.1 Análisis y comparación de metodologías existentes

Se realizó un análisis de las diferentes metodologías empleadas actualmente en el desarrollo de software (véase Apéndice A); concluyendo, al tomar como punto de partida todas las ventajas y desventajas de cada una de las ellas de que la espiral es la adecuada, debido a que, grosso modo, presenta un grado mayor de flexibilidad, conveniente en el proyecto. Dichos puntos antes mencionados se profundizan en las secciones siguientes.

4.2 Metodología espiral

4.2.1 Metodología en espiral

La metodología que más se ajustó al diseño del producto final, una vez hecho el análisis mencionado anteriormente, es la metodología en espiral, ya que funciona u opera en forma de ciclos o iteraciones que converge en un producto integral incremental, priorizando las siguientes fases en función del análisis de riesgos. Entre sus ventajas se encuentran: promover la entrega de prototipos en cada fase, fomentar el análisis de riesgos e incluso puede ser reemplazado algún requisito importante del sistema, contemplando la importancia de la fase en la que se esté trabajando en ese momento o en cada fase de desarrollo; además de no tener establecida ninguna fase previamente como tal [13].

En la (Figura 13), se plantea la forma de trabajo y las distintas fases propuestas por la metodología; así como los distintos rubros en cada fase. En cada vuelta de la espiral se llevan a cabo reuniones periódicas con el equipo involucrado *stakeholders*; a su vez que se prueba, rectifica, y documenta cada fase una y otra vez, así hasta la entrega del producto prototipo o producto final, con su respectiva documentación.

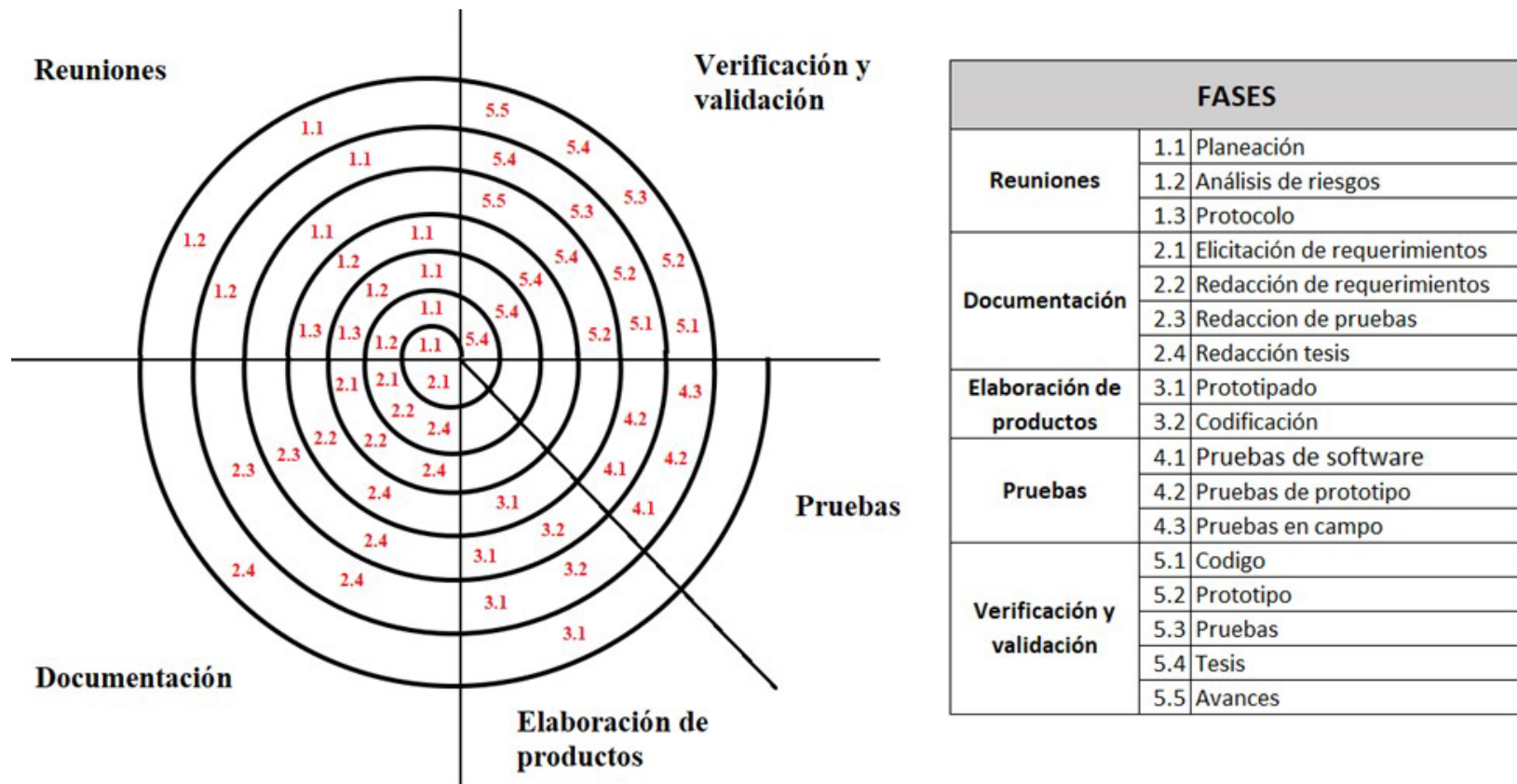


Figura 13.- Ciclo de vida del proyecto y sus fases planteadas

4.2.2 Fases de la metodología en espiral

4.2.2.1 Reuniones

Esta fase propone reuniones periódicas con los interesados de este desarrollo para realizar la planeación y el análisis de riesgos en todo el proyecto; desde la etapa del protocolo hasta la entrega del producto final.

4.2.2.2 Documentación

Todo lo referente a la documentación de los distintos requerimientos y pruebas se realiza en esta fase, siguiendo los estándares como el IEEE29148 [53] y el Código de Ética y Practica Profesional 5.2 [54] ; desde la elicitación y redacción de los requerimientos del sistema, por medio de las técnicas de entrevistas y escenarios, hasta la etapa de pruebas en sus diferentes rubros, los cuales son: pruebas unitarias, integración, sistema y aceptación.

4.2.2.3 Elaboración de productos

Esta sección describe la elaboración de productos, la cual consiste, básicamente en integrar los distintos elementos de hardware y software en un prototipo funcional. Además, contiene la forma de gestionar el versionado de los diferentes scripts de forma incremental, hasta tener un producto funcional final.

4.2.2.4 Pruebas

Las pruebas para considerar son de tres tipos: pruebas de software, de campo y de prototipo, siguiendo lo establecido en el documento de pruebas (Véase Anexo).

4.2.2.5 Verificación y validación

En la etapa de verificación y validación se confirman y aprueban los resultados de las vueltas anteriores para responder las siguientes preguntas: ¿La codificación sigue acorde a lo planeado?, ¿El prototipo funciona y satisface a los requerimientos redactados, después de haber realizado las pruebas correspondientes?, ¿Se cumplió con los objetivos y avances planteados previamente en las reuniones ?, ¿Se ha realizado la documentación acorde a todos los procesos realizados en cada vuelta?

4.3 Herramientas del software desarrollado

Esta sección explica con detalles todas las herramientas de software utilizadas para el desarrollo del producto, a través de diagramas de flujo; además de un análisis en las formas de implementar los distintos algoritmos de la librería OpenCV, para las distintas fases del reconocimiento facial.

4.3.1 Comportamiento de los algoritmos de reconocimiento facial

Los siguientes diagramas mencionan la forma de implementar los algoritmos para gestionar a los usuarios del sistema.

La (Figura 14) está compuesta por 3 secciones, delimitadas por recuadros de colores. En el recuadro verde se describe el proceso de registro de un nuevo usuario al sistema; este es llevado a cabo por las autoridades competentes universitarias. En el rectángulo rojo se puede observar el proceso de reconocimiento facial en usuarios ya registrados. Por último, el recuadro violeta muestra el archivo XML que arroja la etapa de entrenamiento, descrita en la sección 2.3. De manera gráfica, se describe como el sistema opera en dos partes: la administrada por las autoridades y la del reconocimiento facial; unidas por el archivo XML.

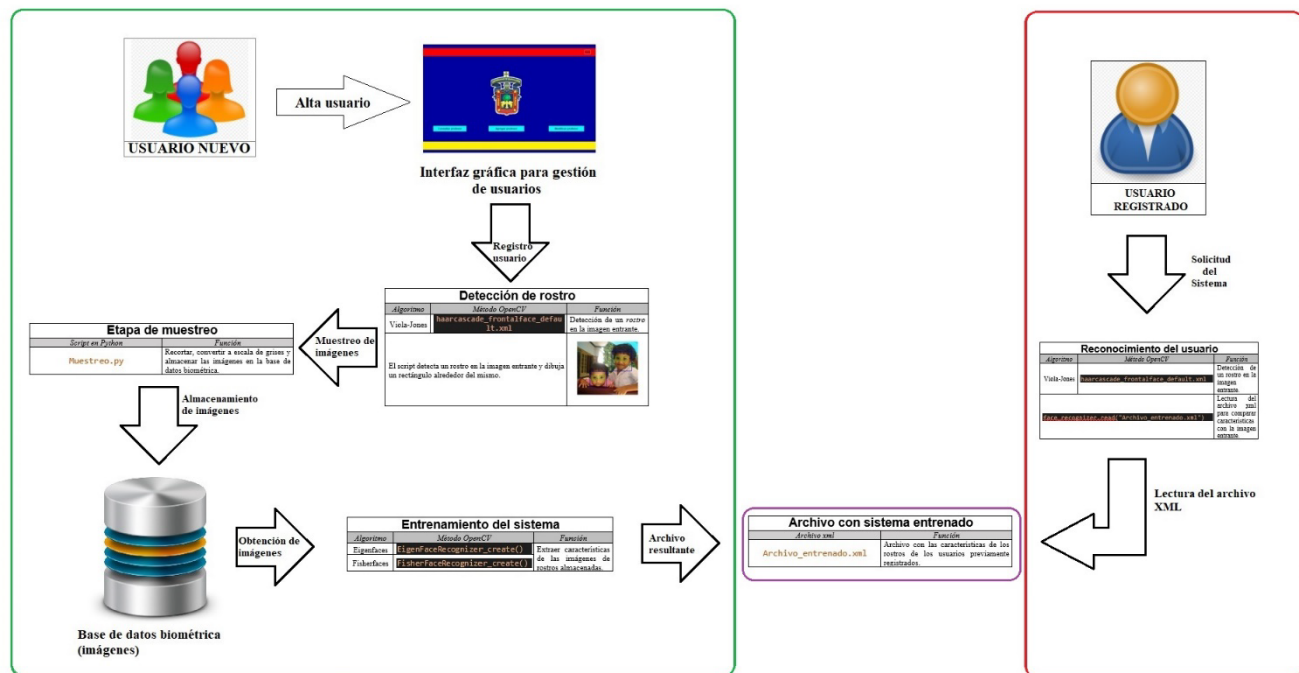


Figura 14.- Explicación gráfica del proceso de gestión de usuario.

4.3.2 Algoritmo de detección

4.3.2.1 Haar Cascade para muestreo de imágenes

El algoritmo *Haar Cascade* se implementa a través de OpenCV; para ello es necesario llamar un archivo con extensión xml. Este archivo contiene la información de características aprendidas durante el entrenamiento para el reconocimiento de rostros. El nombre de este archivo en OpenCV es **haarcascade_frontalface_default.xml** [28]. En la (Figura 15) se muestra un ejemplo funcional de este proceso.

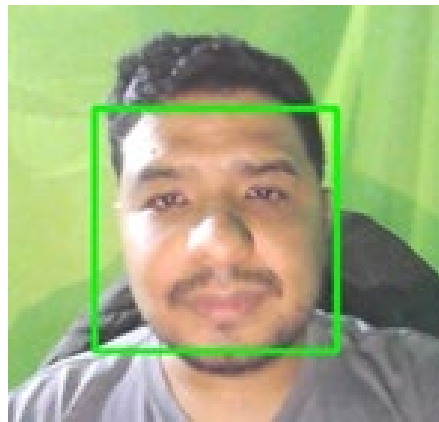


Figura 15.- Detección de rostros con Haar Cascade

Capítulo 5

Diseño

Capítulo 5. Diseño

En este capítulo, se describe como será diseñado el sistema desde distintos ángulos de la ingeniería de software. Utilizando distintos estándares se especifican los tipos de datos que va a utilizar el sistema, como serán las interfaces gráficas, que arquitectura tendrán, cual es el comportamiento esperado explicado en sus diferentes diagramas; todo esto para lograr cumplir con las funcionalidades planteadas en los requerimientos.

5.1 Arquitectura

Una visión general de la forma que tendrá el sistema es analizada en esta sección, vista desde la parte de la arquitectura, descritos en forma de diagramas.

5.1.1 Diagrama general

En esta sección se muestran las diferentes vistas de la arquitectura del sistema. De manera general, el sistema está basado en una Raspberry Pi 3B+, que se interconecta con sus respectivos periféricos y con bases de datos externas; explicado de manera gráfica en la (Figura 16):

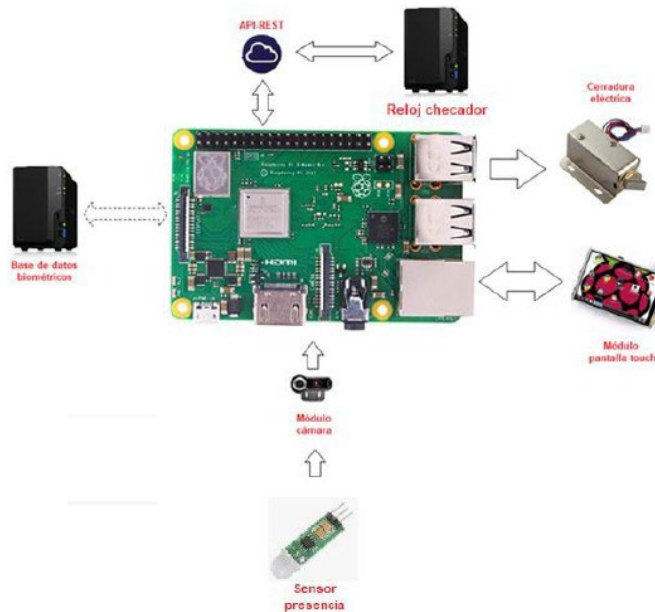


Figura 16.- Arquitectura del sistema basada en Raspberry Pi.

5.1.2 Diagrama de clases

En la (Figura 17) se representan, a través de un diagrama de clases, las distintas funcionalidades del sistema que se propone.

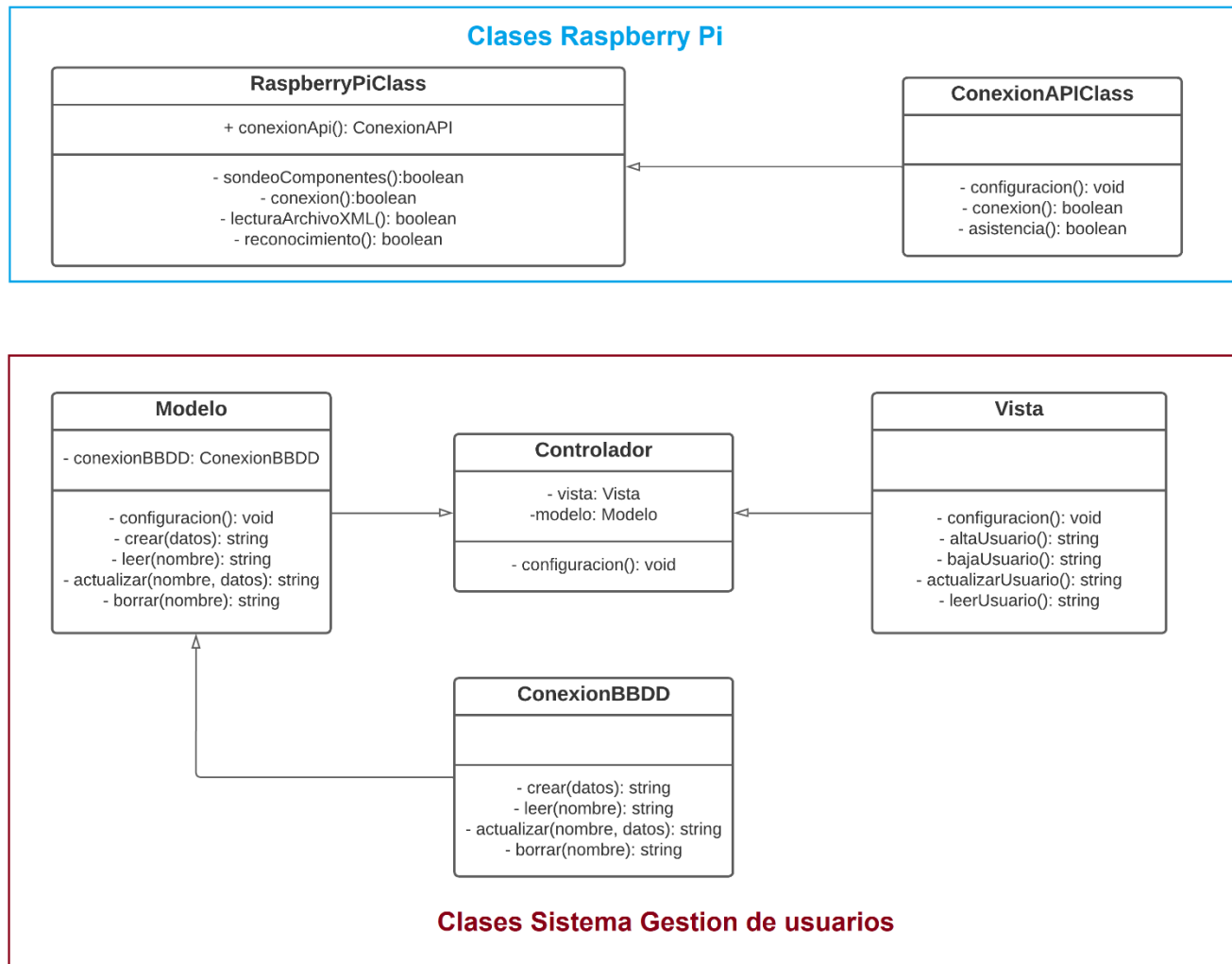


Figura 17.- Diagrama de clases.

5.1.3 Diagrama de casos de uso

5.1.3.1 Caso de uso para alta del usuario

En la (Figura 18) se representa el caso de uso para un usuario nuevo y sus distintas interacciones con el sistema, pasando por IGU y el administrador.

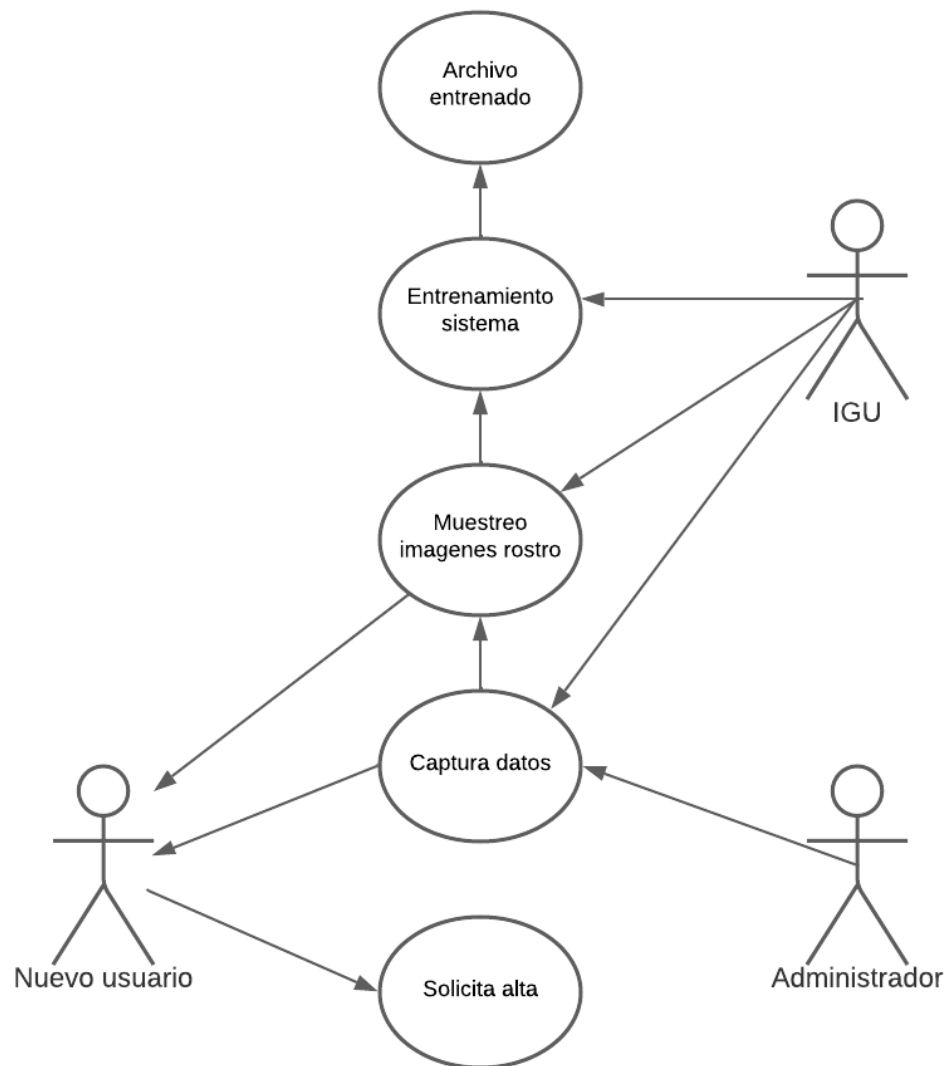


Figura 18.- Diagrama de caso de uso para el usuario nuevo en el sistema.

5.1.3.2 Caso de uso para reconocimiento del usuario registrado.

En la (Figura 19) se representan, a través de un diagrama de casos de uso, las distintas interacciones de un usuario previamente registrado con el sistema.

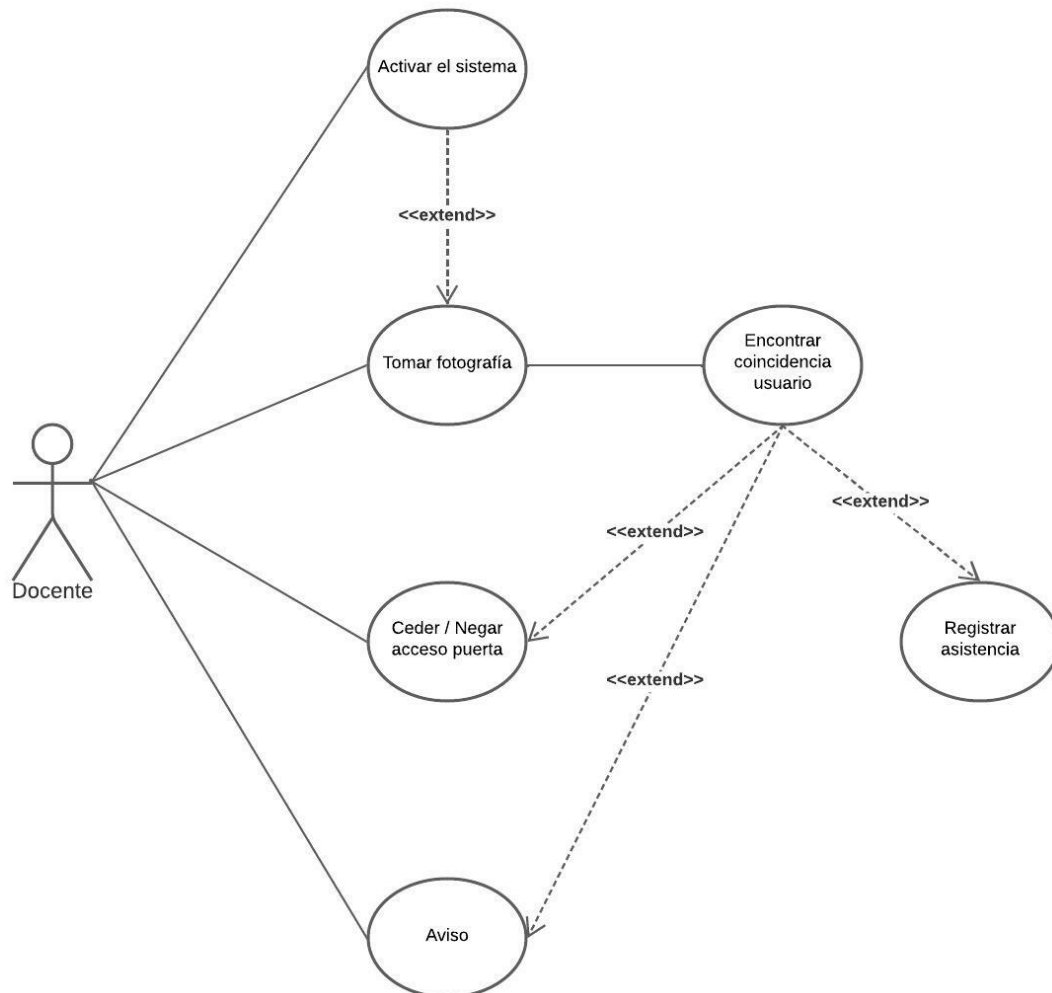


Figura 19.- Diagrama de casos de uso para usuario registrado.

5.2 Vista lógica del sistema

5.2.1 Diagrama de paquetes

Utilizando la arquitectura MVC (Modelo, Vista, Controlador) se describe el funcionamiento del sistema en general. Siguiendo las 3 diferentes capas, donde el modelo contiene los datos, el controlador manipula las distintas operaciones y lógica de negocios y la vista se refiere a la interfaz gráfica con la que el usuario interactúa (Figura 20).



Figura 20.- Diagrama de paquetes

5.3 Vista física del sistema

La parte física del sistema se describe a continuación en forma de módulos y su interacción entre sí, expresados en su diagrama de despliegue.

5.3.1 Diagrama de despliegue

El diagrama de despliegue muestra los componentes físicos de la aplicación. En este caso el usuario inicializa el sistema al acercarse al sensor de proximidad, este ejecuta las funcionalidades y se conecta a través de HTTP local a la base de datos y a la API-Rest (Figura 21).

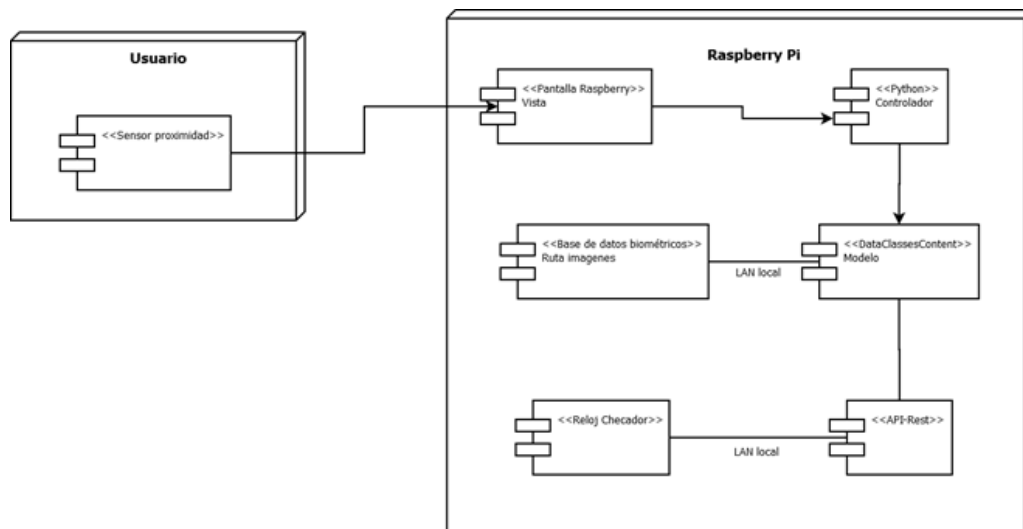


Figura 21.- Diagrama de despliegue.

5.3.2 Diagrama de implementación

El diagrama de implementación muestra los componentes físicos donde el sistema va a habitar. Se puede observar cómo interactúa entre sí en tres partes, como se muestra en la (Figura 22).

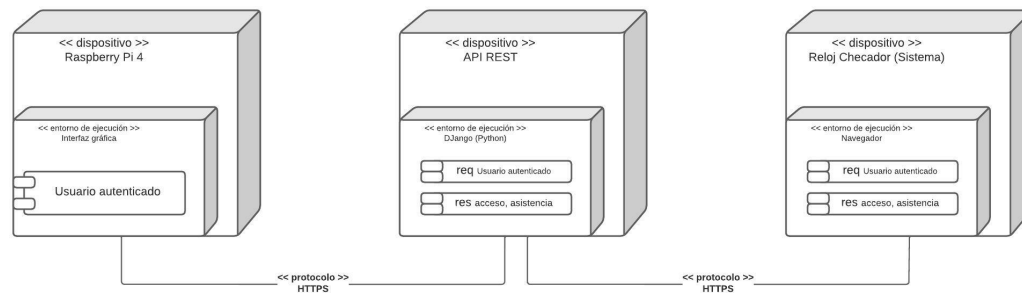


Figura 22.- Diagrama de implementación.

5.4 Vista de procesos del sistema

En esta sección se detalla, a partir de diagramas, las diferentes funcionalidades del sistema y cada uno de los procesos a ejecutar.

5.4.1 Diagrama de actividad

El siguiente diagrama de flujo describe cada proceso del sistema, como se ve en la (Figura 23).

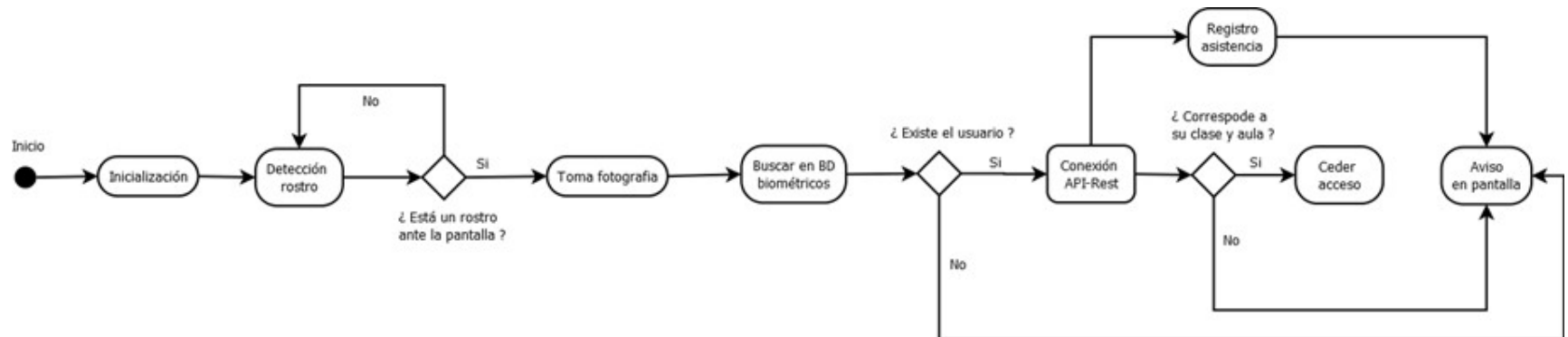


Figura 23.- Diagrama de actividad.

5.5 Comportamiento del sistema

5.5.1 Diagrama de secuencia

A través del diagrama de secuencia como se observa en la (Figura 24), se explica la sucesión de las distintas tareas y procesos principales del sistema.

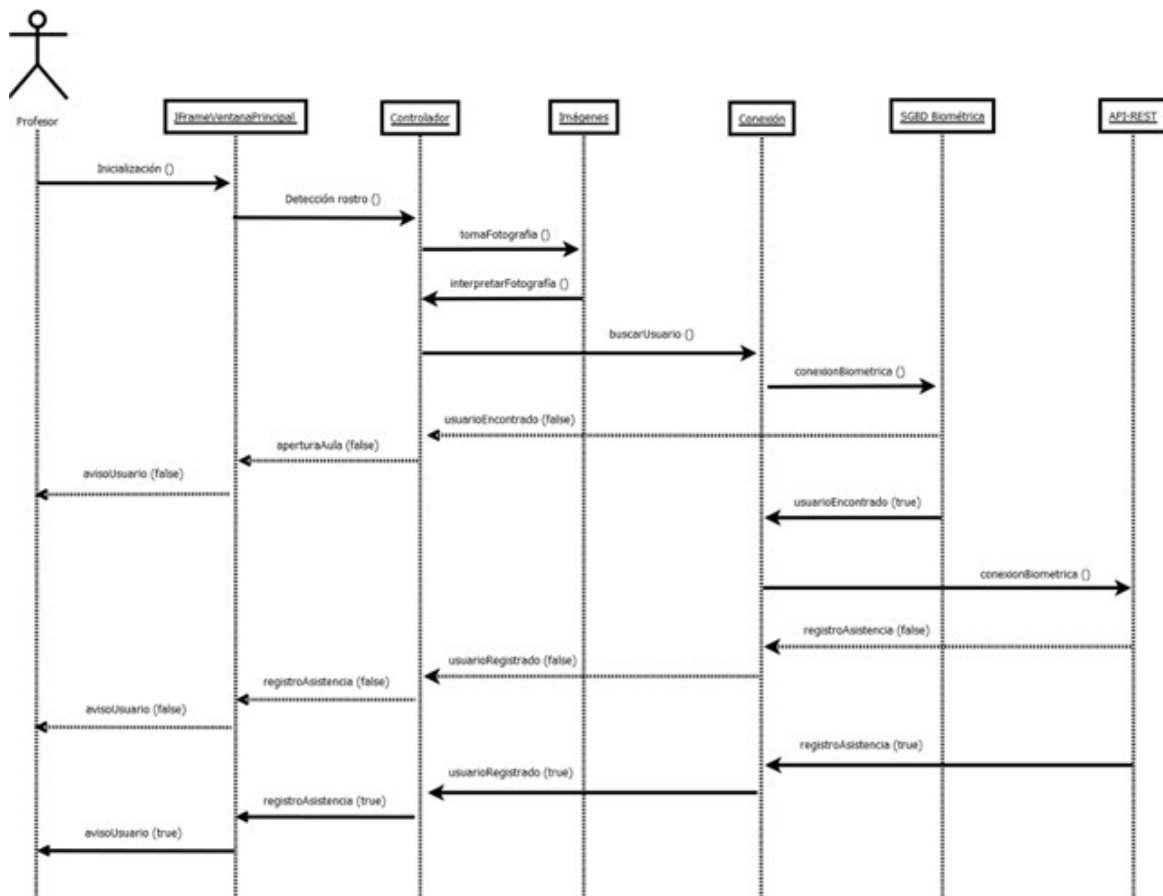


Figura 24.- Diagrama de secuencia.

5.5.2 Modelo entidad-relación

En la (Figura 25) se representa la forma de interactuar del usuario con la base de datos, visto como entidades externas.

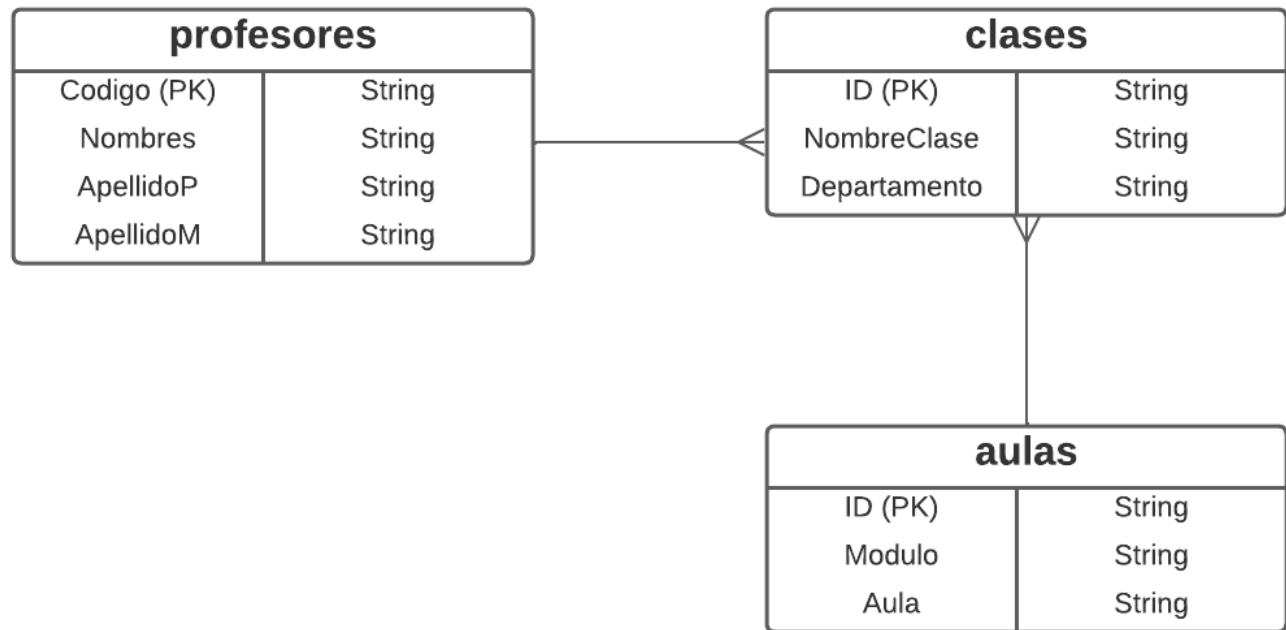


Figura 25.- Modelo entidad-relación (usuarios IGU).

Para la operación de las credenciales del IGU (sección 5.6.1) se diseñó el siguiente modelo, como se puede apreciar en la (Figura 26).

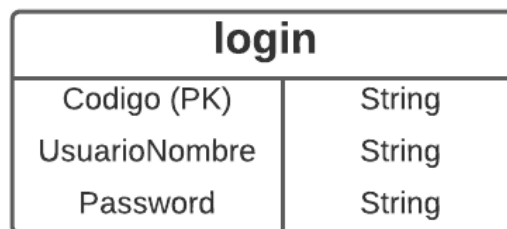


Figura 26.- Modelo entidad-relación (credenciales IGU)

5.6 Interfaces gráficas

El sistema requiere de dos interfaces gráficas para su funcionamiento. Una para la gestión de los usuarios (administrada por las autoridades del CUValles), llamada IGU y otra para la operación de la Raspberry Pi en el producto final.

5.6.1 Interfaz para gestión de usuarios (IGU)

Para llevar a cabo la gestión de los usuarios en el sistema, fue diseñada e implementada una interfaz, llamada *Interfaz de Gestión de Usuario* (IGU). Programada en lenguaje Java y utilizando buenas prácticas como el patrón de diseño Modelo, Vista, Controlador (MVC) permite su escalabilidad con relativa mayor facilidad.

La interfaz para la gestión de usuarios (IGU) es el medio donde las autoridades encargadas administran las imágenes de los profesores en el sistema. Dicha interfaz está compuesta de múltiples ventanas que, a grandes rasgos, operan tres funcionalidades principales: Consultar profesor, Agregar profesor y modificar/Borrar. La (Figura 27) muestra la ventana principal de esta interfaz, observando 3 botones con las funcionalidades antes mencionadas.



Figura 27.- Ventana principal de interfaz para la gestión de usuarios IGU.

5.6.2 Interfaz para Raspberry Pi

La interfaz gráfica de la Raspberry PI es la encargada de comunicar al usuario (profesor) con el sistema. En la (Figura 28), se muestra la ventana del sistema para el reconocimiento facial, la cual será explicada a continuación.

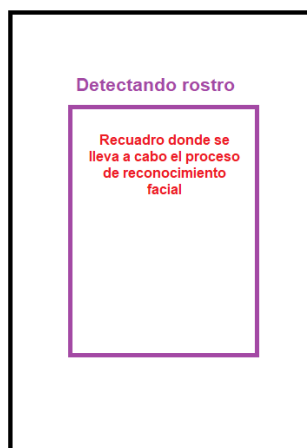


Figura 28.- Ventana al inicializar el sistema.

Una vez que está operando el módulo Pi, al estar diseñado de manera intuitiva, invita al usuario colocar su rostro enfrente del recuadro magenta, para que se detecte su rostro de mejor manera. Si se detecta un rostro, ejecuta el script de reconocimiento para encontrar coincidencia entre la imagen entrante y las entrenadas en el archivo xml. En caso de un reconocimiento negativo, aparece la etiqueta Desconocido en la parte superior derecha, ver (Figura 29).

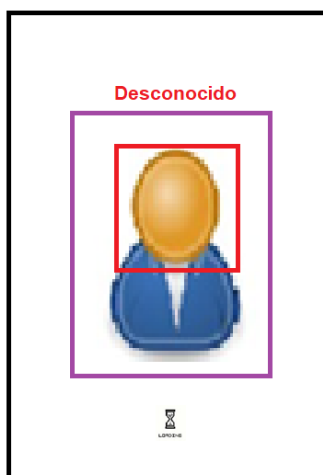


Figura 29.- Ventana al buscar coincidencia.

En caso de un reconocimiento positivo, se despliega la siguiente ventana, ver (Figura 30), con la leyenda “Conocido”, además de la apertura de la puerta y registro de su asistencia.

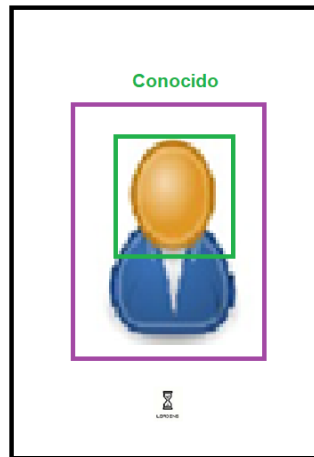


Figura 30.- Ventana al reconocer al usuario.

Capítulo 6

Desarrollo e implementación

Capítulo 6. Desarrollo e implementación

En este capítulo, se detallan las formas de integrar todas las distintas herramientas de software y hardware para realizar el prototipado consecuente, propuesto en las fases de la metodología de desarrollo de software.

6.1 Hardware a utilizar

6.1.1 Raspberry Pi

La Raspberry Pi Foundation es una organización benéfica con sede en el Reino Unido que trabaja para poner el poder de la informática y la creación digital en manos de personas de todo el mundo.

6.1.1.1 Especificaciones técnicas

Las especificaciones técnicas de la Raspberry Pi 4 son las siguientes [8].

Tabla 4.- Especificaciones técnicas Raspberry Pi.

Raspberry Pi 4	
Procesador	SoC de 64 bits Cortex-A72 (ARM v8) de cuatro núcleos a 1,5 GHz.
Memoria	4GB LPDDR4-2400 SDRAM.
Memoria cache	32 KB de datos + 48 KB de caché L1 de instrucciones por núcleo. 1 MB de caché L2.
Multimedia	H.265 (decodificación 4Kp60); H.264 (decodificación 1080p60, codificación 1080p30); OpenGL ES, gráficos 3.0.
Entradas/Salidas	Bus PCIe.
	Puerto Ethernet integrado.
	2 puertos DSI (solo uno expuesto en Raspberry Pi 4B).
	2 puertos CSI (solo uno expuesto en Raspberry Pi 4B).
	Hasta 6 × I2C.
	Hasta 6 × UART (mezclado con I2C).
	Hasta 6 × SPI (solo cinco expuestos en Raspberry Pi 4B).
	Salida de video HDMI dual.
	Salida de video compuesto.

6.1.2 Mini sensor de presencia PIR HC-SR505

6.1.2.1 Especificaciones técnicas

Las especificaciones técnicas del sensor de presencia son las siguientes [56]:

Tabla 5.- Especificaciones técnicas Sensor PIR

PIR HC-SR505	
Tipo:	Mini Sensor PIR.
Modelo	HC-SR505.
Voltaje de entrada	4.5V a 20V DC.
Voltaje de salida	3.3V (Detectado).
Distancia de detección	3m.
Dimensiones	10mm x 24mm.
Tamaño de lente	10 mm (diámetro).
Corriente en reposo	Menor a 60uA.
Tiempo de retardo	8 segundos ($\pm 30\%$ desviación).
Ángulo del sensor	Menor a 100°.
Temperatura de trabajo	-20°C a 70°C.

6.1.3 Cerradura electromecánica

6.1.3.1 Especificaciones técnicas

Por último, las Especificaciones técnicas de la cerradura electromecánica se describen a continuación [57].

Tabla 6.- Especificaciones técnicas Cerradura AX-LOCKR

Cerradura AX-LOCKR	
Material	Acero.
Dirección de apertura	Izquierda.
Tipo de cerradura	Doble cerrojo.
Tiempo de desbloqueo	1 segundo.
Voltaje de operación	12 VDC.

Consumo de corriente	1.5-3 A.
Temperatura de operación	-40° 50°.
Dimensiones	(L X W X D):126 x 102 x 34.5 mm.

6.1.4 Módulo cámara Raspberry Pi

6.1.4.1 Especificaciones técnicas

La (Tabla 7) muestras las especificaciones técnicas del módulo cámara diseñado por Raspberry Pi [11].

Tabla 7.- Especificaciones técnicas módulo cámara

Módulo cámara	
Tamaño	Alrededor de 25 × 24 × 9 mm.
Peso	3g.
Resolución fija	5 megapíxeles.
Modos de video	1080p30, 720p60 y 640 × 480p60 / 90.
Tamaño de pixel	1,4 μm × 1,4 μm.
Tamaño óptico	1/4.
Gama dinámica	67 dB a 8x de ganancia.
Sensibilidad	680 mV / lux-seg.
Profundidad de campo	aprox. 1 m hasta el infinito.
Longitud focal	3,60 mm +/- 0,01.
Campo de visión horizontal	53,50 +/- 0,13 grados.
Campo de visión vertical	41,41 +/- 0,11 grados.
Relación focal (F-Stop)	2.9

6.2 Prototipado

6.2.1 Prototipo 1 (laptop)

6.2.1.1 Etapa de detección de rostros y muestreo

Tomando como muestras, rostros de videos en las distintas plataformas, ejemplificando a usuarios reales, se realizaron muestras de 40 imágenes con los formatos establecidos previamente (Figura 31). En este caso la etapa de muestreo de imágenes y entrenamiento ha sido llevada a cabo en la laptop, relegando solamente la labor del reconocimiento a la Raspberry.

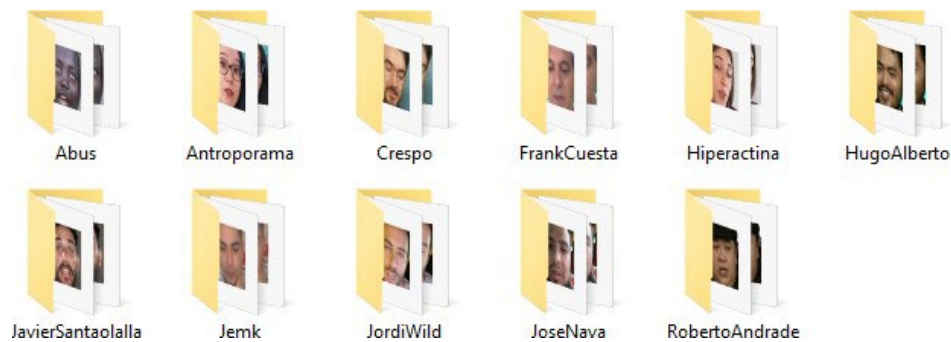


Figura 31.- Carpetas con imágenes muestreadas.

6.2.1.2 Etapa de reconocimiento

En esta etapa de reconocimiento se han corrido pruebas de dos tipos: con videos precargados y con imágenes entrantes por parte de la cámara de la laptop, obteniendo los siguientes resultados:

Para las pruebas con videos se muestran resultados satisfactorios con un 99% de efectividad, cumpliendo con los requerimientos del sistema en lo que se refiere al reconocimiento facial. La (Figura 32) muestra el resultado.

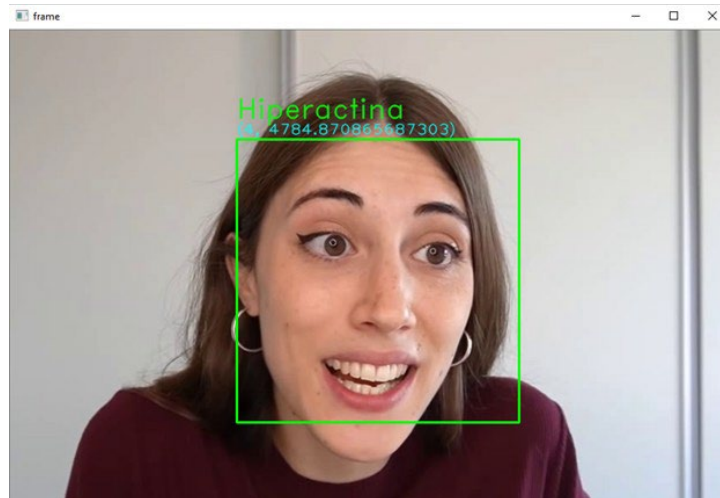


Figura 32.- Reconocimiento facial con video entrante.

Igualmente, la laptop a través de su cámara arrojó resultados satisfactorios, como lo muestra la (Figura 33) (positivo), variando la efectividad en el reconocimiento de manera proporcional bajo condiciones de luz óptimas, como se ejemplifica en la (Figura 34) (negativo).

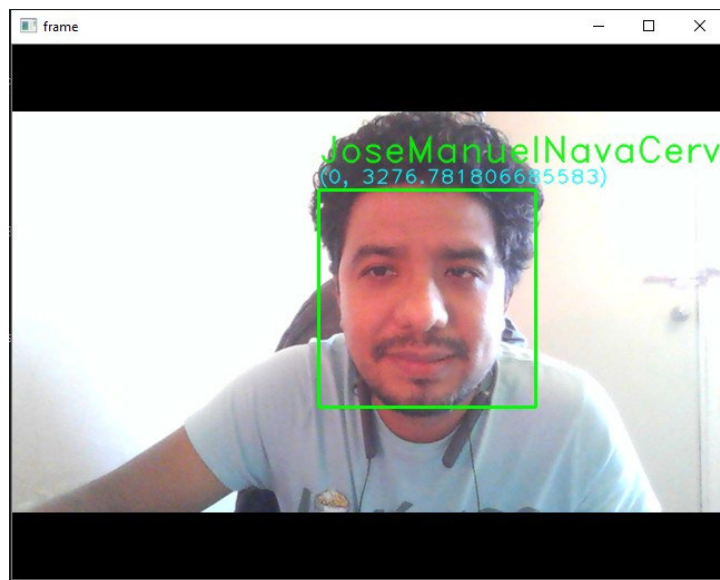


Figura 33.- Reconocimiento facial positivo.

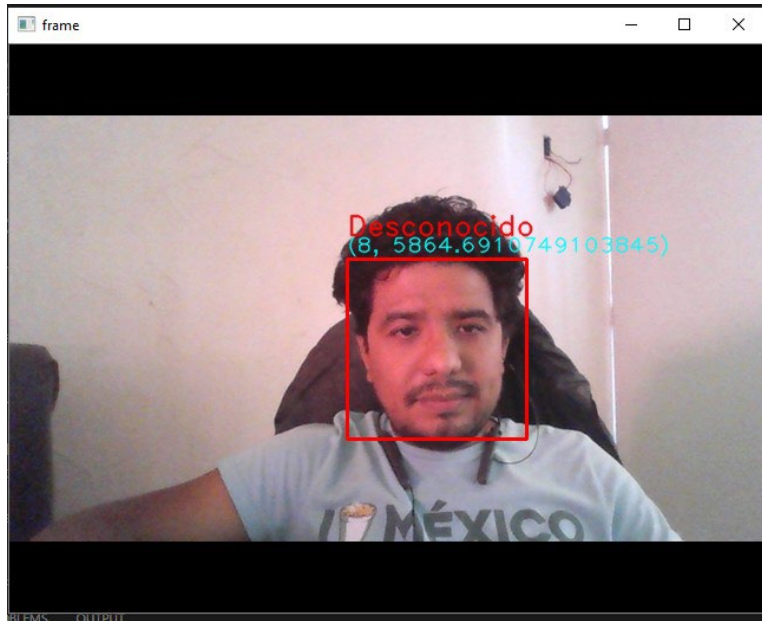


Figura 34.- Reconocimiento facial negativo.

6.2.2 Prototipo 2 (MDF Carcasa)

6.2.2.1 Materiales

Una Raspberry Pi 4, con dos gigas de memoria RAM, memoria SD de 32 gigas y su respectivo modulo cámara; todo esto montado en una caja de material MDF, de manera momentánea; además de un sistema de disipadores-carcasa que cubre toda la placa, como se muestra en la (Figura 35).

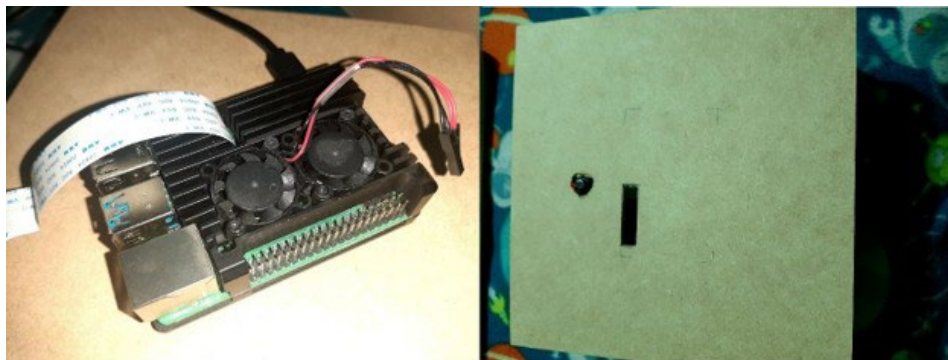


Figura 35.- Prototipado Raspberry Pi.

Para la correcta implementación del primer prototipo, así como sus etapas consecutivas, se utilizó como herramienta principal una laptop con las siguientes características:

- Procesador Intel i5 de 9° generación.
- Memoria RAM de 8 gigas.
- Sistema operativo Windows 10 de 64 bits.
- Tarjeta gráfica NVidia Geforce GTX de 4 gigas.

6.2.2.2 Etapa de detección de rostros y muestreo

La (Figura 36) muestra los resultados obtenidos al ejecutar el programa con el algoritmo de detección de rostros (en laptop). Se puede ver el cuadrado apareciendo en todo momento alrededor del rostro, mientras el programa sigue su ejecución (recortando, convirtiendo a escala de grises y almacenando las imágenes muestreadas, solamente del rostro del usuario).

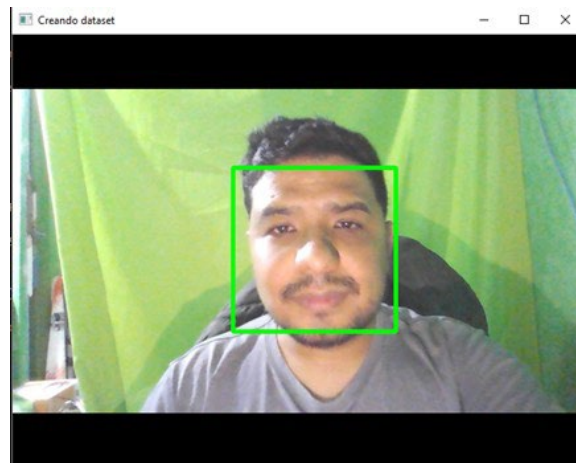


Figura 36.- Script para muestrear imágenes.

Al finalizar el programa anterior, vemos que la carpeta con el nombre del usuario contiene las imágenes, en escala de grises y solo con el rostro como lo demuestra la (Figura 37).



Figura 37.- Imágenes muestreadas.

6.2.2.3 Etapa de entrenamiento

Se realizaron tres entrenamientos con diferentes algoritmos; Eigenfaces, Fisherfaces y LBP; encontrando al primero más conveniente, ya que, éste en tiempo de ejecución, fue el más rápido y en el peso del archivo final, ligeramente menos pesado. De igual manera se tabularon dichos resultados mostrados en la (Tabla 8).

Tabla 8.- Rendimiento en pruebas de algoritmos de entrenamiento

Algoritmo	Cantidad de usuarios	Número de imágenes (por usuario)	Tiempo de ejecución (segundos)	Peso archivo final
Eigenface	11	100	335	669 MB
	11	60	242	394 MB
	11	40	119	265 MB
Fisherface	11	100	350	700 MB
	11	60	250	421 MB
	11	40	120	297 MB
LBP	11	100	342	670 MB
	11	60	242	394 MB
	11	40	125	267 MB

Los resultados hasta este momento han demostrado que las etapas de muestreo y entrenamiento cumplen con los requerimientos establecidos en el SRS en lo que respecta a tiempo de respuesta, confiabilidad y rendimiento, solo en estas etapas.

6.2.2.4 Pruebas en Raspberry Pi

Al ejecutar las pruebas de reconocimiento facial en la Raspberry Pi fueron de manera general satisfactorias, aún con las limitantes de hardware. Esto se puede observar en las (Figuras 38 y 39) donde se muestran dos resultados diferentes al variar las condiciones de luz externas.

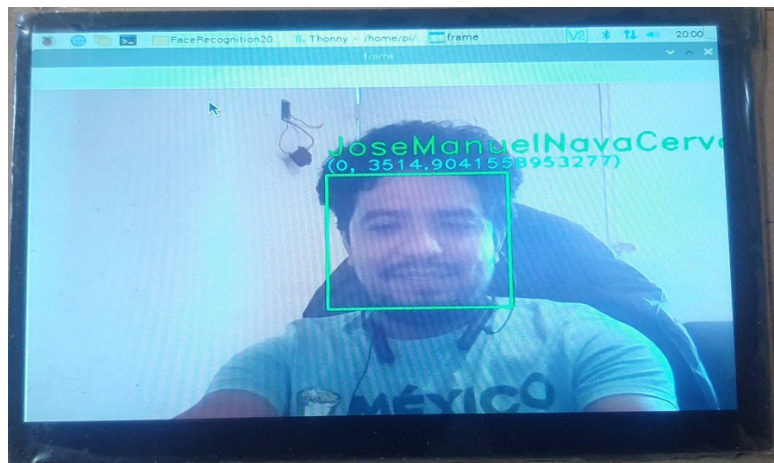


Figura 38.- Reconocimiento facial en Raspberry positivo.

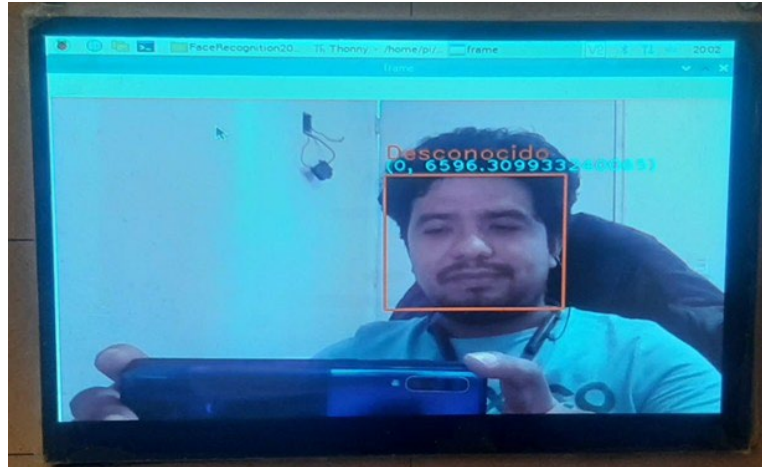


Figura 39.- Reconocimiento facial en Raspberry negativo.

6.2.3 Prototipo 3 (Primera impresión externa)

6.2.3.1 Carcasa impresa en 3D

Como se planteó desde el principio, el sistema requiere de una carcasa impresa en 3D, donde se van a montar todos los diferentes periféricos, así como la Raspberry Pi. Así pues, se diseñó e imprimió, obteniendo los siguientes resultados.

El diseño de la carcasa consta de dos partes principales. La primera, un frente que consiste en una especie de tapa donde se sostienen los siguientes periféricos: el sensor PIR, el módulo cámara, los leds que fungen como flash y la pantalla de 7 pulgadas. Tal como se muestra en la (Figura 40).



Figura 40.- Frente carcasa 3D.

La segunda parte, consta de una especie de caja con diferentes cavidades para la distinta tornillería a utilizar desde el fijamiento en la pared hasta el acople con la primera parte de la impresión; para así lograr cerrar las dos partes y volver un sistema de una sola pieza. En esta parte solo se fija la Raspberry Pi con su respectivo disipador; además de contar con ranuras para posicionar dos ventiladores para disipar el exceso de calor, tal como se muestra en la (Figura 41).

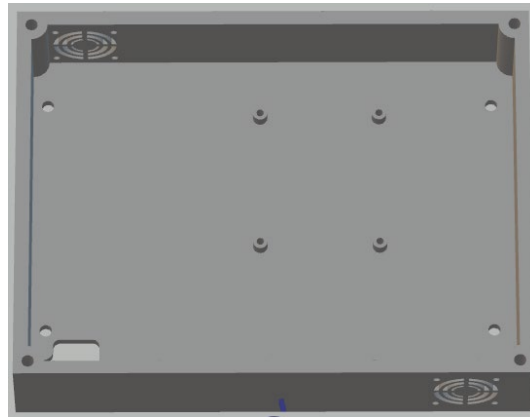


Figura 41.- Caja carcasa 3D.

6.2.3.1.1 Impresión y resultados

La parte de la impresión tuvo algunos inconvenientes no relevantes, que dieron como resultado lo esperado desde el diseño. Las (Figuras 42, 43 y 44) muestran los resultados de la impresión en una impresora de filamento. Aprovechando el tiempo se montaron los diferentes periféricos, antes mencionados que van en cada parte de la impresión.

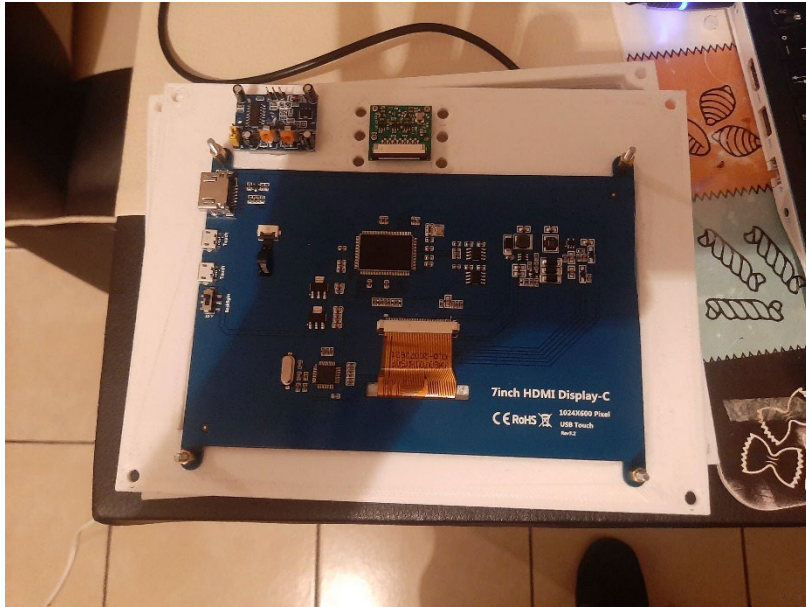


Figura 42.- Tapa impresa parte trasera.



Figura 43.- Tapa impresa parte trasera.



Figura 44.- Carcasa armada.

6.3 Prototipo final

6.3.1 Hardware

A continuación, se mencionan los recursos de software en la elaboración del proyecto:

- Software SolidWorks versión 2021 versión de pago.
- UltimakerCura versión 5.00 versión gratuita.
- Windows 10: Sistema operativo del computador.

6.3.1.1 Recursos externos

Para el proyecto se utilizaron diferentes computadores portátiles de los diseñadores, en el cual estos no fueron considerados dentro de los gastos establecidos. En el apartado de los sistemas operativos uno de ellos era de forma gratuita directamente de proveedor y el otro software utilizado se contó con una licencia para estudiantes, dada por la universidad de procedencia de los estudiantes, por lo que no se tuvieron en cuenta para el cálculo del presupuesto. En la (Tabla 9), se muestra los precios (MXN) de diversos materiales necesarios para la impresión de las armazones o carcasas y para su armado.

Tabla 9.- Tabla de recursos externos usados

Recursos externos			
Recurso	Cantidad	Costo individual	Costo total
Filamento PLA	2	\$ 400	\$ 800
Tornillos M3 con tuerca	10	\$ 7	\$ 70
Tornillos M5 con tuerca	5	\$ 4	\$ 20

6.3.1.2 Simulaciones

Durante el proceso de desarrollo se realizaron pruebas realizando el ensamblaje de los armazones con sus cajas y sus tapas para verificar que se encontraran óptimos para su trabajo. Trabajando con los diferentes prototipos, se hizo un ensamblaje de la tapa y la caja para observar su simetría dejando un pequeño espacio en el ensamblaje para poder visualizar el área de los tonillos que atraviesan el prototipo, Trabajando con el prototipo se hizo el ensamblaje de la tapa y la caja, junto a esto se consiguieron los CADs de los dispositivos que se iban a ocupar para el reconocimiento, esto permitiendo un ensamblaje más completo para su verificación, como se observa en la (Figura 45).



Figura 45.- Ensamblaje realizado en SolidWork para verificación del módulo Pi.

En la (Figura 46), podemos observar la caja base del prototipo la cual fue empotrada a una base de aluminio de la puerta del laboratorio y el apartado de donde se ingresará el cable de alimentación, al igual que en la (Figura 47), podemos observar la tapa del módulo Pi.

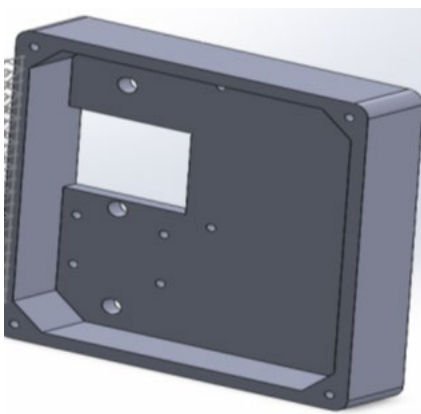


Figura 46.- Carcasa impresa del módulo Pi.

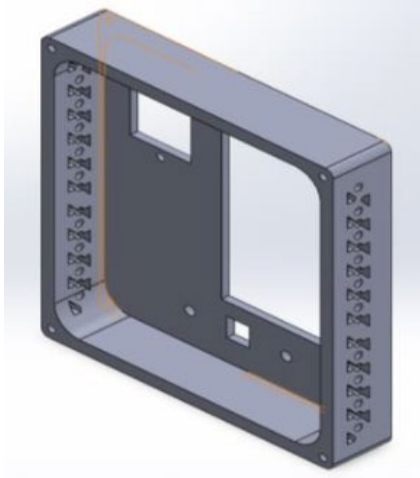


Figura 47.- Tapa correspondiente al módulo Pi

Una vez que se imprimieron los componentes se realizaron las pruebas de montaje de los componentes dentro del armazón como se puede observar en la (Figura 48).



Figura 48.- Montaje de componentes para validación.

Al probar que el área de trabajo era optima, se realizó una prueba de montaje en la puerta designada del edificio *CRECE*, para observar que esta cumpliera con su objetivo como se puede observar en la (Figura 49).



Figura 49.- Prueba de montaje prototipo 2

6.4 Codificación

Los scripts correspondientes a las etapas de muestreo y entrenamiento fueron programados en lenguaje Python y su ejecución fue llevada a cabo en la Laptop.

Para comunicarse con la Raspberry Pi y poder intercambiar información se utiliza el software ssh (implementado por default en la Raspberry Pi).

El código correspondiente, en la etapa de muestreo, utiliza las librerías de OpenCV, para la manipulación de las imágenes, así como los archivos tipo xml de Viola-Jones.

Para la etapa de muestreo, el script creó la carpeta con el nombre del nuevo usuario. como lo muestra la (Figura 50).

```
import cv2
import os
import imutils

personName = 'JoseManuelNavaCervantes_208068119' #Cambie el nombre del nuevo usuario
dataPath = 'C//' #Cambia a la ruta donde hayas almacenado Data
```

```

personPath = dataPath + '\\ ' + personName

if not os.path.exists(personPath):
    print('Carpeta creada: ', personPath)
    os.makedirs(personPath)

cap = cv2.VideoCapture(0, cv2.CAP_DSHOW)

faceClassif =
cv2.CascadeClassifier(cv2.data.haarcascades+'haarcascade_frontalface_default.xml')
count = 0

while True:

    ret, frame = cap.read()
    if ret == False: break
    frame = imutils.resize(frame, width=640)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    auxFrame = frame.copy()

    faces = faceClassif.detectMultiScale(gray, 1.3, 5)

    for (x,y,w,h) in faces:
        cv2.rectangle(frame, (x,y), (x+w,y+h), (0,255,0), 2)
        rostro = auxFrame[y:y+h,x:x+w]
        rostro = cv2.resize(rostro, (150,150), interpolation=cv2.INTER_CUBIC)
        cv2.imwrite(personPath + '_{}.jpg'.format(count), rostro)
        count = count + 1
    cv2.imshow('frame', frame)

    k = cv2.waitKey(1)
    if k == 27 or count >= 100:
        break

cap.release()
cv2.destroyAllWindows()

```

Figura 50.- Código para etapa de muestreo.

Nótese que en la (Figura 50) se utilizó el algoritmo Eigenfaces ya que “El número posible de Eigenfaces es igual al número de imágenes de rostro del grupo de entrenamiento. Sin embargo, los rostros pueden también ser aproximados usando únicamente los mejores Eigenfaces, los que tienen los más grandes eigenvalores y que por consiguiente cuentan para la mayoría de las varianzas dentro del grupo de imágenes de rostro”, por lo tanto “La principal razón para utilizar menos Eigenfaces es por la eficiencia computacional [53]”.

La etapa de entrenamiento fue realizada con ayuda de dos métodos principales bajo OpenCV; el primero llamado *EigenFaceRecognizer*. El segundo método utilizado fue el *train* que, a grandes rasgos, entrena el sistema a partir de características en las imágenes, a modo de parámetros. Al finalizar este proceso, arroja un archivo con extensión xml, este código es mostrado en la (Figura 51). Igualmente, para la implementación de los algoritmos Fisherfaces y LBPH solamente es necesario reemplazar el método *EigenFaceRecognizer.create()* por *FisherFaceRecognizer.create()* o *LBPHFaceRecognizer.create()* [31].

```
import cv2
import os
import numpy as np

dataPath = 'C:/Users'
peopleList = os.listdir(dataPath)

labels = []
facesData = []
label = 0

#Lectura de todas las imágenes almacenadas
for nameDir in peopleList:
    PersonPath = dataPath + '/' + nameDir
    print('Leyendo las imagenes')

    for fileName in os.listdir(PersonPath):
        labels.append(label)
        facesData.append(cv2.imread(PersonPath + '/' + fileName, 0))
        image = cv2.imread(PersonPath+'/'+fileName, 0)
        label= label + 1

#EigenFaces
face_recognizer = cv2.face.EigenFaceRecognizer.create()
print('Entrenando el reconocedor de rostros...')
```

```

#Entrenando el modelo
face_recognizer.train(facesData, np.array(labels))

#Almacenando el modelo obtenido
face_recognizer.write('modeloEigenFaces_postEntrenamiento.xml')
print('Modelo almacenado correctamente')

```

Figura 51.- Código para etapa de entrenamiento.

Por último, el código para la etapa de reconocimiento facial (ejecutada en módulo Pi) funciona a través de determinar parámetros de coincidencia entre la imagen entrante y las entrenadas previamente, que se encuentran en el archivo xml resultante de la etapa anterior, mostrando en pantalla, la variación de coincidencia y el nombre del usuario encontrado; en caso de sobrepasar dicha variación (result), cuenta como usuario desconocido, ver (Figura 52).

```

import cv2
import os

dataPath = 'C:/Usuarios'
imagePath = os.listdir(dataPath)

face_recognizer = cv2.face.EigenFaceRecognizer_create()

#Leyendo el modelo
face_recognizer.read('modeloEigenFaces_postEntrenamiento.xml')

cap = cv2.VideoCapture(0)

faceClassif =
cv2.CascadeClassifier(cv2.data.harcascades+'haarcascade_frontalface_default.xml')

while(True):
    ret,frame = cap.read()
    if ret == False:break
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    auxFrame = gray.copy()

    face = faceClassif.detectMultiScale(gray, 1.1, 5)

```

```

for (x,y,w,h) in faces:
    rostro = auxFrame[y:y+h, x:x+w]
    rostro = cv2.resize(rostro, (150,150), interpolation=cv2.INTER_CUBIC)
    result = face_recognizer.predict(rostro)

    cv2.putText(frame, '{}'.format(result),(x,y-
5),1,1.3,(255,255,0),1,cv2.LINE_AA)

    if result[1] < 5500:
        cv2.putText(frame, '{}'.format(imagePath[result[0]]), (x,y-
25),2,1.1,(0,255,0),1,cv2.LINE_AA)
        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,255,0),2)
    else:
        cv2.putText(frame, 'Desconocido', (x,y-20),2,0.8,(0,0,255),1,cv2.LINE_AA)
        cv2.rectangle(frame,(x,y),(x+w,y+h),(0,0,255),2)

cv2.imshow('reconociendo', frame)
k = cv2.waitKey(1)
if k == 27:
    break

cap.release()
cv2.destroyAllWindows()

```

Figura 52.- Código para etapa de reconocimiento facial.

Capítulo 7

Pruebas

Capítulo 7. Pruebas

En esta parte de la investigación se plantean las diferentes pruebas funcionales de los requerimientos del sistema previamente establecidos en el documento de pruebas. Siguiendo dicho documento se planean los resultados esperados en los diferentes rubros: rendimiento, funcionalidad, portabilidad y precisión.

7.1 Conceptos importantes

- **Falla:** Una falla se presenta cuando un requerimiento dado no es cubierto totalmente. Es una discrepancia entre la funcionalidad actual del sistema y la funcionalidad esperada.
- **Error:** Modificar u omitir un módulo o requerimiento, por parte del programador, ya sea por olvido o de manera intencionada.
- **Defecto:** Proviene de un error de programación y solo pueden hacerse visibles cuando el software es ejecutado.

7.2 Casos de prueba

La redacción de las pruebas está basada en el documento “Ingeniería de Software Código de Ética y Practica Profesional 5.2 [47]” y en el estándar IEEE29148 [46]. Se redactaron 40 diferentes pruebas, en sus cuatro grandes grupos, que son: unitarias, integración, sistema y aceptación; a manera de casos de prueba, siguiendo el formato de la (Tabla 10). En este caso solo se menciona el formato a seguir para el diseño, implementación y mantenimiento de las pruebas (Véase Apéndice B para más información).

Tabla 10.- Formato de redacción de pruebas.

Columna	Instrucciones
Id	Número de identificación del caso de uso
Caso de Prueba	Nombre del caso de prueba
Descripción	Aquí se describe el requerimiento
Objetivo	Que obtendremos al ejecutar esta prueba.
Fecha	Fecha de elaboración de la prueba.
Funcionalidad / Característica	Aquí se describe el funcionamiento de la prueba.
Datos / Acciones de Entrada	<u>Que</u> tipo de dato o acción se requiere para ejecutar la prueba.
Resultado Esperado	Descripción a profundidad de que comportamiento esperar al ejecutar la prueba.
Procedimientos especiales requeridos	En caso de requerir algo especial, mencionarlo aquí.
Dependencias con otros casos de Prueba	En caso de depender de alguna prueba directamente, mencionarlo aquí.
Información para el seguimiento	
Resultado Obtenido	Que arrojo la prueba al ser ejecutada bajo las condiciones mencionadas anteriormente.
Estado	Actualmente como se encuentra la prueba.
Última Fecha de Estado	Cuando se ejecutó la prueba anteriormente
Observaciones	Algo que el tester necesite o guste mencionar sobre lo anterior.

7.2.1 Listado general de documentación de pruebas

Igualmente se elaboró una tabla donde se describen, de manera general las 40 pruebas planteadas en el documento de pruebas, como lo explica la (Tabla 11).

Tabla 11.- Tabla de Casos de Prueba

Número	ID	Categoría	Título	Descripción
1	PU-01	Unitarias	Cantidad de imágenes	Captura de imágenes al detectar un rostro.
2	PU-02		Factor luz	Variación de la luz externa al capturar imágenes.
3	PU-03		Accesorios rostro	Captura de imágenes con lentes.
4	PU-04		Accesorios rostro	Captura de imágenes con barba.
5	PU-05		Falsos positivos	Captura de imágenes falsos rostros.
6	PU-06		Aviso visual positivo	Respuesta del aviso visual (positivo).
7	PU-07		Aviso visual negativo	Respuesta del aviso visual (negativo).
8	PU-08		Aviso sonoro positivo	Respuesta del aviso sonoro (positivo).
9	PU-09		Aviso sonoro negativo	Respuesta del aviso sonoro (negativo).
10	PI-01	Integración	Presencia de periféricos	DESHABILITAR COMPONENTES INTERCALADOS DE MANERA INTENCIONADA (puerta, sensor, api, camara)
11	PI-02		Presencia de periféricos	DESHABILITAR COMPONENTES INTERCALADOS DE MANERA INTENCIONADA (puerta, sensor, api)
12	PI-03		Presencia de periféricos	DESHABILITAR COMPONENTES INTERCALADOS DE MANERA INTENCIONADA (puerta, sensor, camara)
13	PI-04		Presencia de periféricos	DESHABILITAR COMPONENTES INTERCALADOS DE MANERA INTENCIONADA (puerta, sensor)
14	PI-05		Presencia de periféricos	DESHABILITAR COMPONENTES INTERCALADOS DE MANERA INTENCIONADA (puerta, api, camara)
15	PI-06		Presencia de periféricos	DESHABILITAR COMPONENTES INTERCALADOS DE MANERA INTENCIONADA (puerta, api)
16	PI-07		Presencia de periféricos	DESHABILITAR COMPONENTES INTERCALADOS DE MANERA INTENCIONADA (puerta, camara)
17	PI-08		Presencia de periféricos	DESHABILITAR COMPONENTES INTERCALADOS DE MANERA INTENCIONADA (puerta)
18	PI-09		Presencia de periféricos	DESHABILITAR COMPONENTES INTERCALADOS DE MANERA INTENCIONADA (sensor, api, camara)
19	PI-10		Presencia de periféricos	DESHABILITAR COMPONENTES INTERCALADOS DE MANERA INTENCIONADA (sensor, api)
20	PI-11		Presencia de periféricos	DESHABILITAR COMPONENTES INTERCALADOS DE MANERA INTENCIONADA (sensor, camara)
21	PI-12		Presencia de periféricos	DESHABILITAR COMPONENTES INTERCALADOS DE MANERA INTENCIONADA (sensor)
22	PI-13		Presencia de periféricos	DESHABILITAR COMPONENTES INTERCALADOS DE MANERA INTENCIONADA (api, camara)
23	PI-14		Presencia de periféricos	DESHABILITAR COMPONENTES INTERCALADOS DE MANERA INTENCIONADA (api)
24	PI-15		Presencia de periféricos	DESHABILITAR COMPONENTES INTERCALADOS DE MANERA INTENCIONADA (camara)
25	PI-16		Presencia de periféricos	DESHABILITAR COMPONENTES INTERCALADOS DE MANERA INTENCIONADA ()
26	PI-17		Compatibilidad entre datos	Lectura de datos de internet.
27	PI-18		Compatibilidad entre datos	Escritura de los datos entre módulos.
28	PI-19	Sistema	Flujo de información	Como fluye los datos entre módulos.Funcionamiento entre la API-REST y el módulo registro de asistencia (valida).
29	PI-20		Flujo de información	Como fluye los datos entre módulos.Funcionamiento entre la API-REST y el módulo registro de asistencia (invalida).
30	PS-01		Apertura puerta	Abrir la puerta según sea el caso (negativo).
31	PS-02		Apertura puerta	Abrir la puerta según sea el caso (positivo).
32	PS-03		Tiempo de respuesta	Cronometrar el tiempo de respuesta desde el sensor hasta la respuesta final
33	PS-04		Vulneración	Vulnerar el sistema con fotografías
34	PS-05	Aceptación	Vulneración	Vulnerar el sistema con celular
35	PS-06		Vulneración	Vulnerar el sistema con Tablet
36	PA-01	Aceptación	Horario	Checar el sistema fuera de horario.
37	PA-02		Fecha	Checar el sistema fuera de fecha.
38	PA-03		Rendimiento	Confiabilidad en la apertura de la puerta con 50 usuarios
39	PA-04		Aviso asistencia	Registro efectivo de asistencia, a través de un aviso sonoro
40	PA-05		Aviso asistencia	Registro efectivo de asistencia, a través de un aviso visual

7.2.2 Resultados

Para la captura de los resultados, en lo referente a las pruebas previamente planteadas, se muestran solamente 10 pruebas (para fines prácticos) y sus resultados de ejecución en el producto final (Véase Apéndice B para más información).

7.2.2.1 Unitarias

Tabla 12.- Prueba unitaria PU-01.

Columna	Instrucciones
Id	PU-01
Caso de Prueba	Cantidad de imágenes
Descripción	Captura de imágenes al detectar un rostro.
Objetivo	Checar como la cantidad de imágenes influye en la detección de caras.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Variar la cantidad de imágenes capturadas desde 50 hasta 200 para determinar la cantidad óptima.
Datos / Acciones de Entrada	Que el usuario posicione su rostro a 30 cm de la cámara y en posición frontal.
Resultado Esperado	Obtener la cantidad de imágenes solicitadas.
Procedimientos especiales requeridos	No aplica
Dependencias con otros casos de Prueba	No aplica
Información para el seguimiento	
Resultado Obtenido	La cantidad óptima de imágenes de caras encontrada fue de 60 imágenes por usuario.
Estado	Ejecutada
Última Fecha de Estado	12/2022
Observaciones	Resultado obtenido al encontrar la mínima cantidad de imágenes requeridas para satisfacer los requerimientos del sistema con 10 usuarios .

Tabla 13.- Prueba unitaria PU-05

Columna	Instrucciones
Id	PU-05
Caso de Prueba	Falsos positivos.
Descripción	Captura de imágenes falsos rostros.
Objetivo	Detectar cuantas imágenes falsas de rostros se detectan y almacenan.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Revisar la cantidad de falsos rostros en las imágenes capturadas y almacenadas en el banco de imágenes.
Datos / Acciones de Entrada	Que el usuario posicione su rostro a 30 cm de la cámara y en posición frontal.
Resultado Esperado	Que la cantidad de falsos positivos en la detección de rostros sea menor al 3%.
Procedimientos especiales requeridos	No aplica
Dependencias con otros casos de Prueba	PU-01
Información para el seguimiento	
Resultado Obtenido	Solamente una imagen diferente de un rostro fue detectada y almacenada por cada prueba con 60 imágenes.
Estado	Ejecutada
Última Fecha de Estado	12/2022
Observaciones	Esta prueba fue ejecutada con 10 usuarios encontrando en promedio 11 imágenes de falso positivo, respecto con las 600 imágenes capturadas; aproximadamente el 2%

7.2.2.2 Integración

Tabla 14.- Prueba de integración PI-01.

Columna	Instrucciones
Id	PI-01
Caso de Prueba	Presencia de periféricos (deshabilitados).
Descripción	Deshabilitar todos los componentes periféricos.
Objetivo	Checar comportamiento del sistema al desconectar periféricos.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el sistema imprima en pantalla el (los) componentes desconectados.
Datos / Acciones de Entrada	Desconectar los 4 periféricos: puerta, sensor proximidad, cámara y API-REST
Resultado Esperado	Que la pantalla imprima todos los componentes no detectados.
Procedimientos especiales requeridos	Desconectar todos los periféricos
Dependencias con otros casos de Prueba	No aplica
Información para el seguimiento	
Resultado Obtenido	De manera general, al inicializar el sistema, todos los componentes muestran como estado false .
Estado	Ejecutada
Última Fecha de Estado	12/2022
Observaciones	Esta prueba agregó etiquetas en la pantalla de la Raspberry Pi por medio del script, pero solo para pruebas. Al momento de operar no se muestran.

Tabla 15.- Prueba de integración PI-02

Columna	Instrucciones
Id	PI-02
Caso de Prueba	Presencia de periféricos (habilitados)
Descripción	Habilitar todos los componentes periféricos.
Objetivo	Checar comportamiento del sistema al conectar periféricos.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el sistema imprima en pantalla el (los) componentes conectados.
Datos / Acciones de Entrada	Conectar los 4 periféricos: puerta, sensor proximidad, cámara y API-REST
Resultado Esperado	Que la pantalla imprima todos los componentes detectados.
Procedimientos especiales requeridos	Conectar todos los periféricos
Dependencias con otros casos de Prueba	No aplica
Información para el seguimiento	
Resultado Obtenido	De manera general, al inicializar el sistema, todos los componentes muestran como estado true .
Estado	Ejecutada
Última Fecha de Estado	12/2022
Observaciones	Esta prueba agregó etiquetas en la pantalla de la Raspberry Pi por medio del script, pero solo para pruebas. Al momento de operar no se muestran.

Tabla 16.- Prueba de integración PI-17

Columna	Instrucciones
Id	PI-17
Caso de Prueba	Compatibilidad entre datos
Descripción	Lectura de los datos entre módulos.
Objetivo	Checar la correcta lectura de la asistencia del usuario a través de la API.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Leer la fecha en el formato DIA / MES / AÑO y la hora en 24 horas, con sus minutos solamente.
Datos / Acciones de Entrada	Correcta detección de un usuario del sistema.
Resultado Esperado	Impresión en pantalla de la fecha y hora con un mensaje del estado de la solicitud: "201"
Procedimientos especiales requeridos	Una base de datos de prueba, para probar la API-Rest.
Dependencias con otros casos de Prueba	PA-03
Información para el seguimiento	
Resultado Obtenido	Respuesta de estatus 200 por parte de la API. Resultado satisfactorio al revisar la lectura en la base de datos.
Estado	Ejecutada
Última Fecha de Estado	12/2022
Observaciones	API simulado con el software POSTMAN.

Tabla 17.- Prueba de integración PI-18.

Columna	Instrucciones
Id	PI-18
Caso de Prueba	Compatibilidad entre datos
Descripción	Escritura de los datos entre módulos.
Objetivo	Checar la correcta escritura de la asistencia del usuario a través de la API.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Escribir la fecha en el formato DIA / MES / AÑO y la hora en 24 horas, con sus minutos solamente.
Datos / Acciones de Entrada	Captar fecha y hora desde el dispositivo. Correcta detección de un usuario del sistema.
Resultado Esperado	Impresión en pantalla de la fecha y hora con un mensaje del estado de la solicitud: "201"
Procedimientos especiales requeridos	Una base de datos de prueba, para probar la API-REST.
Dependencias con otros casos de Prueba	PA-03
Información para el seguimiento	
Resultado Obtenido	Respuesta 200 por parte de la API. Resultado satisfactorio al revisar la escritura en la base de datos.
Estado	Ejecutada
Última Fecha de Estado	12/2022
Observaciones	API simulado con el software POSTMAN.

7.2.2.3 Sistema

Tabla 18.- Prueba de sistema PS-01.

Columna	Instrucciones
Id	PS-01
Caso de Prueba	Apertura puerta
Descripción	Abrir la puerta según sea el caso.
Objetivo	Validar la correcta apertura de la puerta.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Ceder / Negar accesos en la puerta principal.
Datos / Acciones de Entrada	Imagen capturada, con un usuario del sistema.
Resultado Esperado	Apertura de la puerta a través de la cerradura y el relevador; solo en caso de detección efectiva.
Procedimientos especiales requeridos	Iluminación externa.
Dependencias con otros casos de Prueba	No aplica.
Información para el seguimiento	
Resultado Obtenido	Al ejecutar la prueba con 5 usuarios, se obtuvo 4 aperturas positivas por una falsa. Un 80% de efectividad.
Estado	Ejecutada
Última Fecha de Estado	12/2022
Observaciones	Pruebas ejecutadas en condiciones de luz natural de mediana-baja eficiencia.

Tabla 19.- Prueba de sistema PS-04.

Columna	Instrucciones
Id	PS-04
Caso de Prueba	Vulneración
Descripción	Vulnerar el sistema mostrando en la pantalla una fotografía con el rostro de un usuario previamente registrado.
Objetivo	Validar el requisito de confiabilidad, establecido en SRS.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Seguridad en el sistema.
Datos / Acciones de Entrada	Colocar en la cámara una fotografía de un usuario registrado.
Resultado Esperado	Muestra del usuario en estatus Desconocido y la no apertura de la puerta.
Procedimientos especiales requeridos	Luz externa.
Dependencias con otros casos de Prueba	No aplica.
Información para el seguimiento	
Resultado Obtenido	Al ejecutar la prueba con 5 usuarios, se obtuvieron 4 reconocimientos en estatus Desconocido y uno Conocido. Eficiencia del 80%
Estado	Ejecutada.
Última Fecha de Estado	12/2022
Observaciones	Pruebas ejecutadas en condiciones de luz natural de mediana-baja eficiencia.

7.2.2.4 Aceptación

Tabla 20.- Prueba de aceptación PA-01.

Columna	Instrucciones
Id	PA-01
Caso de Prueba	Horario
Descripción	Checar el sistema fuera de horario.
Objetivo	Validar la confiabilidad en el sistema.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Comportamiento el sistema fuera de horario.
Datos / Acciones de Entrada	Ingresar hora diferente a la estipulada, de manera manual.
Resultado Esperado	No apertura de la puerta.
Procedimientos especiales requeridos	Ingresar horario diferente al establecido (clase correspondiente).
Dependencias con otros casos de Prueba	No aplica
Información para el seguimiento	
Resultado Obtenido	Respuesta de la API con estatus 401.
Estado	Ejecutada.
Última Fecha de Estado	12/2022
Observaciones	Se ingresó el horario manualmente en el script, diferente a la correspondiente (hora clase simulada).

Tabla 21.- Prueba de aceptación PA-01.

Columna	Instrucciones
Id	PA-02
Caso de Prueba	Fecha
Descripción	Checar el sistema fuera de horario.
Objetivo	Validar la confiabilidad en el sistema.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Comportamiento el sistema fuera de horario.
Datos / Acciones de Entrada	Ingresar fecha diferente a la estipulada, de manera manual.
Resultado Esperado	No apertura de la puerta.
Procedimientos especiales requeridos	Ingresar fecha fuera del horario establecido.
Dependencias con otros casos de Prueba	No aplica
Información para el seguimiento	
Resultado Obtenido	Respuesta de la API con estatus 401.
Estado	Ejecutada.
Última Fecha de Estado	12/2022
Observaciones	Se ingresó la fecha manualmente en el script, diferente a la correspondiente (fecha clase simulada).

Capítulo 8

Resultados, conclusiones y trabajos futuros

Capítulo 8. Resultados, conclusiones y trabajos futuros

Este capítulo se compone de 3 secciones que son: a) los resultados obtenidos con el prototipo final, b) conclusiones generales y particulares obtenidas en el desarrollo de este proyecto y c) las proyecciones futuras referentes a la escalabilidad del sistema.

8.1 Resultados

Esta sección menciona los resultados obtenidos, una vez que se obtuvo el prototipo final, que como ya se mencionó anteriormente, se compone de dos secciones la IGU y el módulo PI.

8.1.1 Software para gestión de usuarios

Como se menciona en la sección 5.6 se diseñó e implementó un software IGU, encargado de gestionar a los usuarios (Figura 53). Cabe mencionar su funcionamiento, de manera general, como un sistema tipo CREATE, READ, UPDATE, DELETE (CRUD, por sus siglas en inglés) administrado por las autoridades (sección 5.5.2).

La operación del IGU, se explica a continuación conforme a sus respectivos botones.

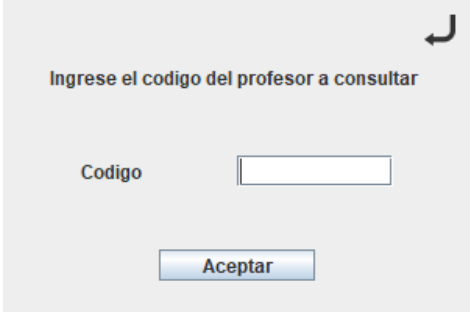


Figura 53.- Ventana principal de interfaz para la gestión de usuarios.

8.1.1.1 Botón *Consultar profesor*

8.1.1.1.1 Ventana consultar profesor

Al presionar el botón Consultar profesor se oculta la ventana principal (Figura 53) y se despliega la siguiente ventana (Figura 54), para realizar la búsqueda del profesor a través de su código.

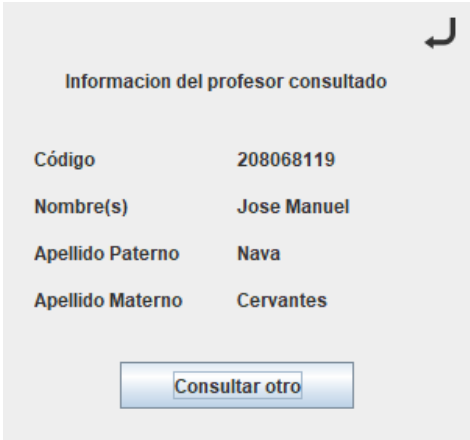
A screenshot of a software window titled 'Ingrese el codigo del profesor a consultar'. It features a text input field labeled 'Codigo' and a blue button labeled 'Aceptar'. A back arrow icon is in the top right corner.

Ingrese el codigo del profesor a consultar	
Codigo	
Aceptar	

Figura 54.- Ventana consultar profesor.

8.1.1.1.2 Ventana profesor consultado

La (Figura 55), muestra la ventana con la información del usuario.

A screenshot of a software window titled 'Informacion del profesor consultado'. It displays a table with the following data: Código (208068119), Nombre(s) (Jose Manuel), Apellido Paterno (Nava), and Apellido Materno (Cervantes). At the bottom is a blue button labeled 'Consultar otro'. A back arrow icon is in the top right corner.

Informacion del profesor consultado	
Código	208068119
Nombre(s)	Jose Manuel
Apellido Paterno	Nava
Apellido Materno	Cervantes
Consultar otro	

Figura 55.- Ventana profesor consultado.

8.1.1.2 Botón Agregar profesor

8.1.1.2.1 Ventana *Login*

Al presionar el botón Agregar profesor en la ventana principal, se despliega la ventana siguiente como se observa en la (Figura 56). En este caso aparecen dos campos los cuales son requeridos para otorgar privilegios de administrador en el sistema; usuario y contraseña. Estos campos utilizan una encriptación de tipo MD5 para protección.

A screenshot of a login window titled 'Ingrese sus datos'. It features a back arrow in the top right corner. Below the title, there are two input fields: 'Usuario' with the text 'admin' and 'Contraseña' with five dots. A blue 'Aceptar' button is positioned at the bottom center.

Figura 56.- Ventana Log In.

8.1.1.2.2 Ventana alta usuario (login negativo)

En el caso de que el administrador se halla equivocado o alguien ajeno intente acceder a esta funcionalidad, se despliega la siguiente ventana, ver (Figura 57).

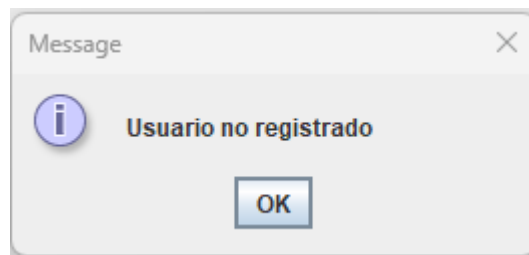


Figura 57.- Mensaje login negativo

8.1.1.2.3 Ventana alta usuario (login positivo)

Después de haberse identificado satisfactoriamente, el administrador procede a capturar los datos del nuevo usuario, por medio de los campos que se encuentran en la siguiente ventana, como se observa en la (Figura 58).

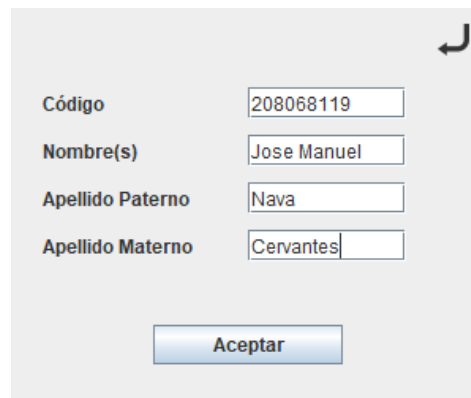
Una ventana de software con un fondo gris claro. En la esquina superior derecha hay un icono de flecha curva hacia arriba y a la izquierda. Hay cuatro campos de texto con etiquetas a la izquierda: 'Código' con el valor '208068119', 'Nombre(s)' con 'Jose Manuel', 'Apellido Paterno' con 'Nava', y 'Apellido Materno' con 'Cervantes'. Debajo de estos campos hay un botón rectangular con el texto 'Aceptar'.

Figura 58.- Ventana alta usuario.

Cuando todos los datos ingresados previamente se hallan registrado, se emite el siguiente mensaje como se observa en la (Figura 59).

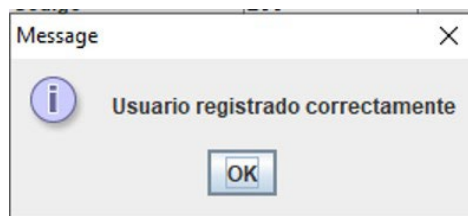


Figura 59.- Ventana de confirmación Alta usuario.

8.1.1.2.4 Script para muestreo de imágenes.

En el IGU, al momento de capturar la información del nuevo usuario en la base de datos, se ejecuta el script de Python referente a la etapa de captura y muestreo de las imágenes para el sistema (sección 6.4). Esto se muestra en la siguiente (Figura 60).

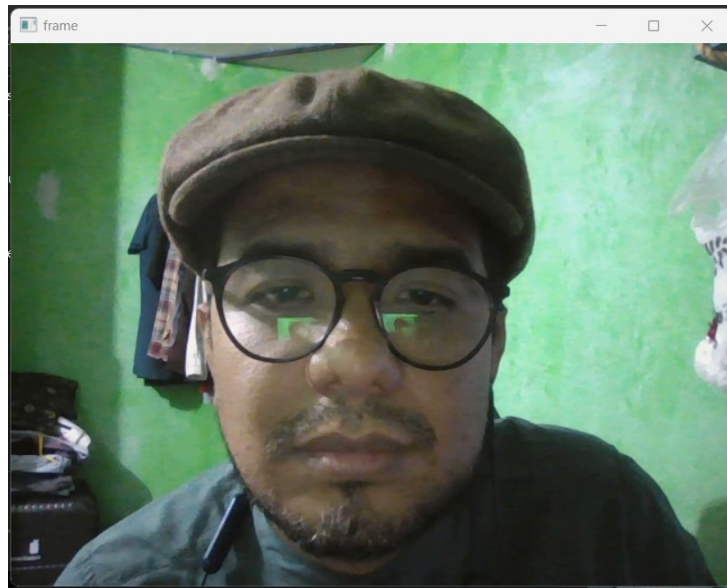


Figura 60.- Script Python para muestreo de imágenes.

Al finalizar la tarea, se crea una nueva carpeta que contiene las respectivas 100 imágenes del rostro del nuevo usuario nombradas y enumeradas. La (Figura 61) ejemplifica este complejo proceso.

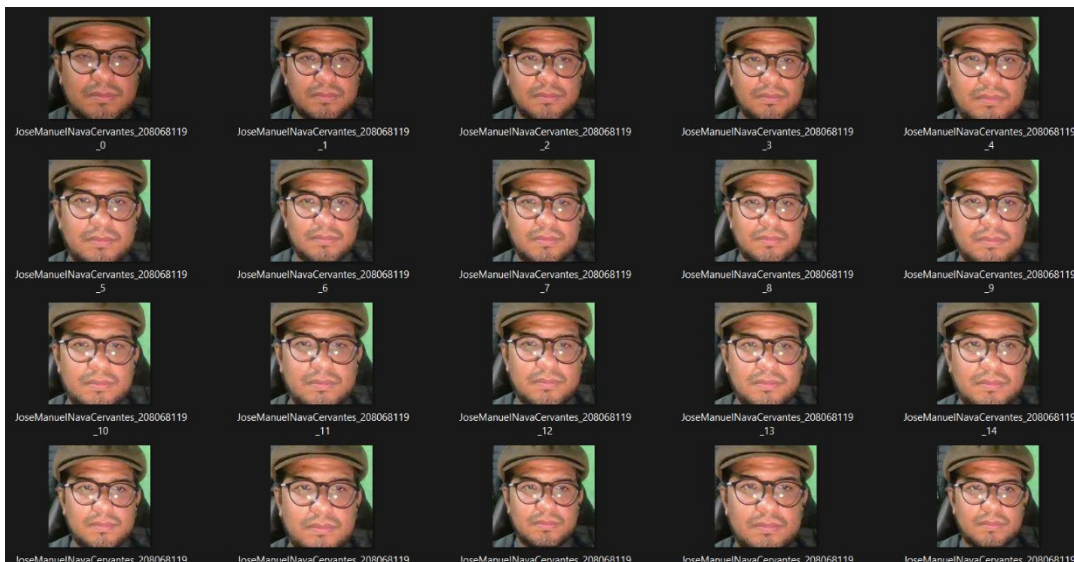


Figura 61.- Carpeta con imágenes del rostro del nuevo usuario.

8.1.1.2.5 Botón Modificar/Borrar

De igual manera que en la sección Ventana login (8.1.1.2.1) es requerido que el administrador se identifique; en caso de que el proceso de login sea negativo, se repite la etapa mencionada en la sección 8.1.1.2.2. El resultado de presionar el botón Modificar/Borrar y de un login positivo, logra el despliegue de la ventana, como se ve en la (Figura 62), que contiene dos botones: modificar y borrar; al presionar el primero, actualiza los datos a través de los campos con la información del usuario. En el caso de presionar el botón *Eliminar*, borra los datos del usuario en la base de datos y la carpeta con las imágenes de su rostro.



Código	100
Nombre(s)	Roberto
Apellido Paterno	Rodríguez
Apellido Materno	Rodríguez
Modificar Borrar	

Figura 62.- Ventana Modificar / Borrar usuario.

8.1.2 Sistema módulo Pi

Estos fueron los resultados al montar el módulo Pi en el marco de la puerta que proporciona el acceso al laboratorio de innovación y desarrollo tecnológico. Internamente en la carcasa se encuentran conectados los respectivos componentes, dejando a la vista solo su parte operativa como lo muestra la (Figura 63). Se puede observar la distribución de los periféricos de la siguiente manera: sensor PIR del lado superior izquierdo, el módulo cámara rodeado por dos leds ultrabrillantes que fungen de iluminación artificial externa y el display en forma vertical.



Figura 63.- Montura del Módulo Pi.

Al energizar el módulo Pi, se carga correctamente el sistema operativo de la Raspberry y se ejecuta el script del sistema RF, para en un tiempo de 30 segundos, entrar en modo espera, tal como se observa en las (Figuras 64 y 65).



Figura 64.- Inicialización correcta del Módulo Pi.



Figura 65.- Sistema en modo de espera.

Para activar el reconocimiento facial en el módulo Pi, basta con acercarse a escasos centímetros, donde el sensor PIR ejecuta el script encargado y activa los leds, como se observa en la (Figura 66).

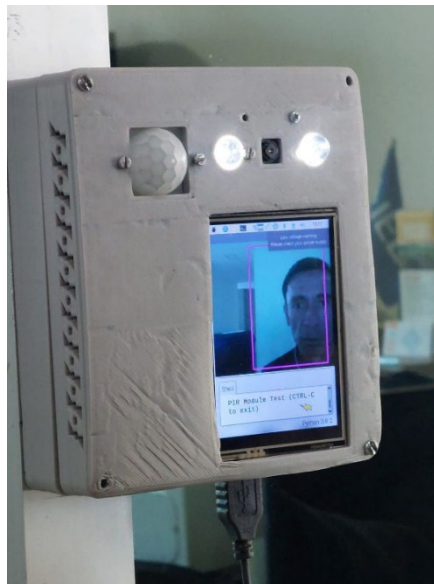


Figura 66.- Sistema en modo de detección de rostros en la imagen entrante.

Por último, la etapa de reconocimiento facial muestra las dos situaciones de los usuarios en el sistema: Conocido (Figura 67) y Desconocido (Figura 68).

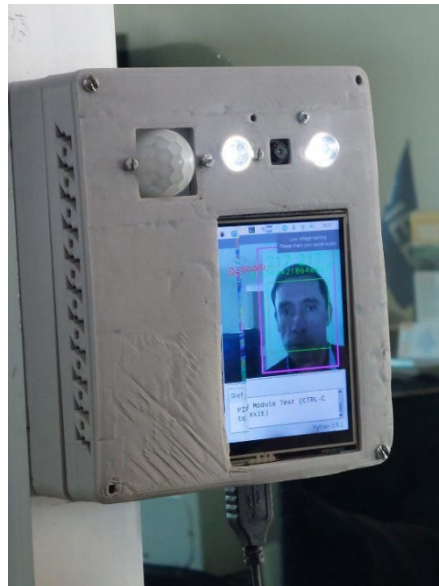


Figura 67.- Reconocimiento facial con estatus “Conocido”

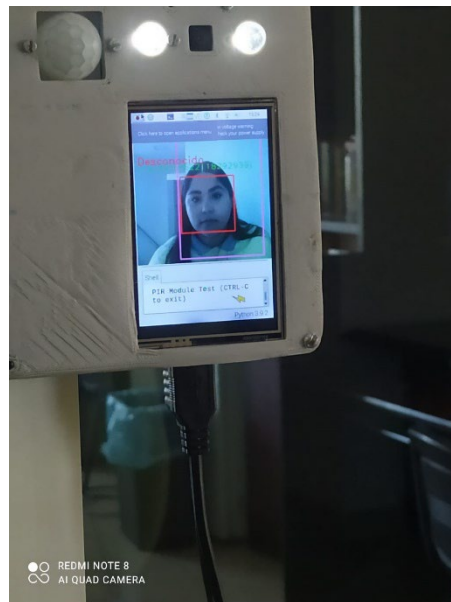


Figura 68.- Reconocimiento facial con estatus “Desconocido”.

8.1.3 Comparativa entre algoritmos de reconocimiento

Se llevo a cabo la etapa del reconocimiento facial, por cada algoritmo (EigenFaces, FisherFaces y LBPH) en su archivo XML, para determinar el tiempo y la tasa de reconocimiento del usuario; arrojando los siguientes resultados en las (Tablas 22 y 23) .

Tabla 22.- Comparativa de reconocimiento con 50 imágenes.

Parámetro	EigenFaces	FisherFaces	LBPH
Número de imágenes entrenadas	50	50	50
Tiempo de respuesta	12 s	7 s	69 s
Tasa de reconocimiento positivo	85 %	80 %	83%
Tasa de falsos positivos	15 %	20 %	17 %

Tabla 23.- Comparativa de reconocimiento con 100 imágenes.

Parámetro	EigenFaces	FisherFaces	LBPH
Número de imágenes entrenadas	100	100	100
Tiempo de respuesta	16 s	9 s	9 s
Tasa de reconocimiento positivo	88 %	82 %	86 %
Tasa de falsos positivos	12 %	18 %	14 %

8.1.4 API Reloj Checador

Para la creación de la API que simula el servicio de reloj checador, se utilizó un framework llamado NestJs, que grosso modo puede describirse como la implementación de buenas prácticas con el respaldo de lenguajes como Javascript y Typescript, para la creación de servidores altamente escalables [58]. Debido a su relativa facilidad de implementación se pudo diseñar un backend de prueba con un endpoint que funciona de la siguiente manera:

Una dirección local que recibe como parámetro el código del profesor, el cual fue previamente reconocido en el módulo Pi, tal como lo describe la (Figura 69).

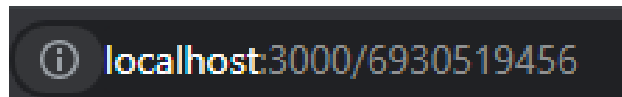


Figura 69.- Endpoint de prueba para registro asistencia.

Al realizar la solicitud GET en dicho endpoint, nos regresa un estatus 201 con información del profesor en cuestión. La (Figura 70), representa la respuesta del backend.

```
{
  "id": "ef2df6fb-0df5-4728-bca3-5280007eaa57",
  "status": "201",
  "asistencia": "Se registró la asistencia del profesor Daren correctamente",
  "fecha": "Martes 16 de Enero del 2023",
  "hora": "15:34 hrs."
}
```

Figura 70.- Respuesta a la solicitud del módulo Pi..

Con estos datos, fue posible tomar la decisión de permitir o negar el acceso al laboratorio, a través del relevador.

8.2 Conclusiones

Las diferentes fuentes citadas en el estado del arte proyectan al reconocimiento facial como el principal mecanismo biométrico de identificación; siendo equiparable en estos momentos a otro mecanismo como la huella dactilar, en lo que respecta a confiabilidad y tasa de casos positivos.

El haber sustentado este proyecto en estándares como el IEEE29148 y el código de ética y práctica profesional de la ingeniería de software, aseguran un software de calidad en sus distintas etapas de diseño y arquitectura; así como una documentación legible y detallada.

La tasa obtenida en la etapa de muestreo (detección de rostros) cumple los resultados esperados desde la fase de requerimientos, en lo referente al rendimiento, tiempo de respuesta y eficiencia, al presentar un número mínimo de falsos positivos equivalente al 1%, respecto de la muestra total.

El haber realizado una comparativa entre los diferentes algoritmos (Eigenfaces, Fisherfaces y LBPH) en la etapa de entrenamiento y la de reconocimiento, arroja un resultado confiable al contar con más del 80% de casos positivos en el reconocimiento de los usuarios que fueron probados, permite tomar la decisión respecto al algoritmo óptimo entre las diferentes variables contempladas, como son el tiempo de cómputo requerido, peso final del archivo y tasa de casos positivos; siendo elegido EigenFaces al arrojar mayor índice de casos positivos, bajo las condiciones actuales en las que está instalado el equipo.

Se obtuvo una significativa mejora al implementar el algoritmo Viola-Jones en un área específica de las imágenes entrantes (recuadro magenta).

En la fase de reconocimiento, la tasa de casos positivos satisface los requerimientos mínimos, aun con las limitantes del hardware e iluminación externa, al obtener una tasa del 88%.

Una de las ventajas que presenta el módulo Pi es su bajo costo. Permitiendo que su fabricación en serie y/o el reemplazo de alguno de sus periféricos, en caso de ser requerido, sea mucho más conveniente al estar diseñado sobre componentes de fácil obtención y precio accesible.

Gracias a la metodología empleada, se le dio al sistema un enfoque de producto escalable, debido al uso de buenas prácticas desde el principio. En caso de necesitar agregar alguna funcionalidad, es necesario solamente trabajar sobre un nuevo ciclo en la espiral.

Para el diseño de la API de prueba, se utilizaron solamente tablas y relaciones que simulan el comportamiento de la base de datos real, arrojando resultados satisfactorios.

Cabe mencionar que este proyecto surgió en el auge de la contingencia sanitaria por el COVID-19, la cual inspiró su diseño como un sistema intuitivo y de baja intrusión para el usuario, al requerir únicamente de su rostro para ser usado.

8.3 Trabajos futuros

Gracias a la escalabilidad mencionada anteriormente en el documento, se pueden agregar y/o modificar funcionalidades y módulos en el sistema. Debido a esto se proyecta la adición de las siguientes tareas o funcionalidades para su continuación:

- Sustitución de los algoritmos convencionales por técnicas de aprendizaje profundo.
- Registro de asistencia de los alumnos.
- Control de los aires acondicionados e iluminación en cada aula.

Referencias

Referencias

- [1] M. A.-C. M. S. a. S. J. C. S. J. Johnston, "Applicability," *IEEE*, pp. 141-146, 2016.
- [2] X. M. S. L. a. H. P. L. Li, "A review of face recognition technology," *IEEE Access*, vol. 8, 2020.
- [3] K. a. O. H. C. R. a. H. M. a. Y. S. N. a. S. V. P. Lai, "Risk, Trust, and Bias: Causal Regulators of Biometric-Enabled Decision Support," *IEEE Access*, vol. 8, 2020.
- [4] N. B. Sukhai, "Access Control & Biometrics," *Association for Computing Machinery*, pp. 124-127, 2004.
- [5] C. S. a. R. P.-G. a. R. C. a. J. L. a. M. W. G. D. a. F. A. a. O. Pascalis, "Face recognition is shaped by the use of sign language," *J. Deaf Stud. Deaf Educ.*, vol. 23, pp. 1-9, 2018.
- [6] H. a. W. X. a. Y. D. a. L. Z. a. Z. X. a. L. S. Z. Shi, "Cross-Modality Face Recognition via Heterogeneous Joint Bayesian," *IEEE Signal Processing Letters*, vol. 24, pp. 81-85, 2017.
- [7] H. a. H. X. Yang, "Face Recognition Attendance System Based on Real-Time Video Processing," *IEEE Access*, vol. 8, 2020.
- [8] R. Pi, "Raspberry," Raspberry, 2022. [Online]. Available: <https://www.raspberrypi.com/products/raspberry-pi-4-model-b/>.
- [9] N. S. { a. S. Yamanoor, "High quality, low cost education with the Raspberry Pi," in *2017 IEEE Global Humanitarian Technology Conference (GHTC)*, 10.1109/GHTC.2017.8239274}, 2015.
- [10] R. P. Foundation, "About us," Raspberry Pi Foundation, 2022. [Online]. Available: <https://www.raspberrypi.org/about/>.
- [11] R. Pi, "Módulo Cámara," Raspberry Pi, 2022. [Online]. Available: <https://www.raspberrypi.com/documentation/accessories/camera.html#camera-modules>.
- [12] lcdwiki, "lcdwiki.com," 2022. [Online]. Available: http://www.lcdwiki.com/3.5inch_RPi_Display.
- [13] B. W. Boehm, "A spiral model of software development and enhancement," *Computer*, vol. 21, pp. 61-72, 1988.
- [14] J. L. a. S. Huan, "The implementation of the visitor access control system for the senior citizen based on the LBP face recognition," 2017.
- [15] E. L. Aldea, "Raspberry Pi," in *Raspberry Pi Fundamentos y aplicaciones*, 1ra ed., Madrid, 2017, p. 66.
- [16] B. A. Moises, "Internet de las cosas," in *Internet de las cosas*, Madrid, España, 2018, p. 19.
- [17] P. A. López, "Seguridad informática," in *Seguridad informática*, Madrid, España, 2010, p. 62.
- [18] merriam-webster, "Latencia definición," [Online]. Available: <https://www.merriam-webster.com/dictionary/latent>.
- [19] M. C. Huguet, Administración de sistemas operativos en red, Barcelona, España: <https://books.google.com.mx/books?id=w4utLelkYgkC\pg=PA144\dq=protocolos+IP\hl=es-419\sa=X\ved=2ahUKEwjHsvHyvMbrAhUBMqwKHxNVATAQ6wEwBHoECAIQAQ\&v=onepage\&q=protocolos\20IP\&f=false>, 2008, p. 144.
- [20] Oracle, "¿Qué es Java?," 2021. [Online]. Available: https://www.java.com/es/about/whatis_java.jsp. [Accessed 06 04 2021].

- [21] R. G. Duque, "Python para todos," in *Python para todos*, España, 1ra Edición, 1991, p. 12.
- [22] V. D. H. a. A. P. a. M. Puglia, "RFID-A guide to radio frequency identification," John Wiley \ Sons, Inc, 2007, p. 1.
- [23] Intel, "About OpenCV," [Online]. Available: <https://opencv.org/about/>. [Accessed 10 09 2022].
- [24] F. d. A. L. Fuentes, *Sistemas Distribuidos, Una década de la Unidad Cuajimalpa de la Universidad Autónoma Metropolitana*, 2015, pp. 63-64.
- [25] J. M. O. Candel, *Tecnologías para arquitecturas basadas en microservicios: Patrones y soluciones para aplicaciones desplegadas en contenedores*, Google Play Books, 2020, p. 27.
- [26] S. Domínguez, "Reconocimiento facial mediante el análisis de componentes principales (PCA)," Sevilla, España, 2017.
- [27] P. V. a. M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001.
- [28] Google, "Haar cascade classifier," [Online]. Available: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html. [Accessed 10 09 2022].
- [29] Google, "Boosting," [Online]. Available: <https://docs.opencv.org/2.4.13.7/modules/ml/doc/boosting.html>. [Accessed 10 09 2022].
- [30] I. a. M. D. a. C. F. a. V. P. Granja, "¿ Qué es LDA ?," in *Procesamiento De Imágenes Para La Identificación De Personas Como Sistema De Seguridad En Zonas Domiciliarias*, 2020, p. 164.186.
- [31] Google, "Face Recognition with OpenCV," [Online]. Available: https://docs.opencv.org/3.4/da/d60/tutorial_face_main.html. [Accessed 10 09 2022].
- [32] Google, "docs.opencv," [Online]. Available: https://docs.opencv.org/3.4/da/d60/tutorial_face_main.html. [Accessed 10 09 2022].
- [33] J. H. D. K. P.N. Belhumeur, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, July 1997.
- [34] L. S. a. M. Kirby, "Low-dimensional procedure for the characterization of human faces," *Journal of the Optical Society of America A*, vol. 4, no. 3, pp. 519-524, 1987.
- [35] A. H. M. P. Timo Ahonen, "Face Recognition with Local Binary Patterns," *Lecture Notes in Computer Science*, vol. 3021, pp. 469-481, 2004.
- [36] OpenCV, "opencv.org," Face Recognition with OpenCV, [Online]. Available: https://docs.opencv.org/3.4/da/d60/tutorial_face_main.html. [Accessed 15 01 2023].
- [37] S. P. a. V. K. a. S. A. a. S. Sahu, "Smart Home Security Using IoT and Face Recognition," in *Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, Pune, India, 2018.
- [38] F. F. a. S. A. Hossain, "Smart Security System Using Face Recognition on Raspberry Pi," in *13th International Conference on Software, Knowledge, Information Management and Applications (SKIMA)*, Island of Ulkulhas, Maldives, 2019.
- [39] E. R. M. A. F. R. Arturo Zúñiga-López and Juan Villegas Cortez and Carlos Avilés Cruz, "Reconocimiento de rostros por medio de Openface en una Raspberry Pi," Ciudad de México

México, 2018.

- [40] J. L. a. X. F. a. T. Zhang, "A smart system for face detection with spatial correlation improvement in IoT environment," in *2017 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computed, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation*, 2017.
- [41] S. a. S. S. K. a. K. K. Chakraborty, "Facial Biometric System for Recognition Using Extended LGHP Algorithm on Raspberry Pi," *IEEE Sensors Journal*, vol. 20, pp. 8117-8127, 2020.
- [42] M. M. a. K. G. D. a. C. N. A. a. A. J. M. a. I. M. M. Hasan Moon, "Design and Implementation of a Vault Security System," in *2019 1st International Conference on Advances in Science, Engineering and Robotics Technology (ICASERT)*, 2019.
- [43] T. F. Gabor Arva, "Embedded Video Processing on Raspberry Pi," *IEEE*, vol. 10, 2017.
- [44] "Development of a Secured Door Lock System Based on Face Recognition using Raspberry Pi and GSM Module," *IJTSRD*, 2019.
- [45] D. K. S. Syed fasiuddin, "Real Time Application of Vehicle Anti Theft," *IEEE*, 2020.
- [46] L. P. C. P. T. L. Ngo Tung Son, "Combination of Facial Recognition and Interaction with Academic Portal in Automatic Attendance System," in *ICSCA '19: Proceedings of the 2019 8th International Conference on Software and Computer Applications*, Vietnam, 2019.
- [47] T. D. Adnan Firoze, "Face Recognition Time Reduction Based on Partitioned Faces without Compromising Accuracy and a Review of state-of-the-art Face Recognition Approaches," in *Proceedings of the 2018 International Conference on Image and Graphics Processing*, Hong Kong, Hong Kong , 2018.
- [48] V. P. C. K. S. D. Ishita Gupta, "Face detection and recognition using Raspberry Pi," in *EEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*, Pune, India, 2016.
- [49] R. E. Ahmed El-Mahdy, "A Large-Scale Mobile Facial Recognition System Using Embedded GPUs," Computer Science and Engineering Department, Egypt-Japan, 2015.
- [50] N. D. A. S. Sudhir Chitnis, "An investigative study for smart home security: Issues, challenges and countermeasure," *Wireless Sensor Network*, vol. 8, no. 4, 04 2016.
- [51] Techground, "Techground.es," US: E Smart lock, nueva cerradura inteligente con desbloqueo facial, [Online]. Available: <https://techground.es/us-e-smart-lock-cerradura-inteligente-desbloqueo-facial/>. [Accessed 14 01 2023].
- [52] M. Libre, "Mercadolibre.com.mx," Cerradura Wifi Reconocimiento Facial 6 Modos De Apertura, [Online]. Available: https://articulo.mercadolibre.com.mx/MLM-1508250955-cerradura-wifi-reconocimiento-facial-6-modos-de-apertura-_JM?matt_tool=17264456&matt_word=&matt_source=google&matt_campaign_id=15700894061&matt_ad_group_id=130350624206&matt_match_type=&matt_network=g&ma. [Accessed 14 01 2023].
- [53] ISO/IEC/IEEE, "ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering," *ISO/IEC/IEEE 29148:2018(E)*, pp. 1-104, 2018.
- [54] E. T. S. University, "Software Engineering Code of Ethics and Professional Practice," 2001. [Online]. Available: <https://experts.arizona.edu/en/publications/software-engineering-code-of-ethics-and-professional-practice>. [Accessed 14 01 2023].

- [55] K. K. a. J. Kendall, "Elicitacion de requerimientos," in *Análisis y diseño de sistemas*, Pearson, Ed., 2011, p. 104.
- [56] Alldatasheet, "Especificaciones técnicas Sensor PIR," [Online]. Available: <https://www.alldatasheet.com/datasheet-pdf/pdf/1284342/ETC1/HC-SR505.html>. [Accessed 14 01 2023].
- [57] Tecnosinergia, "CERRADURA ELECTROMECANICA AXCEZE AX-LOCKR ORIENTACION DERECHA INSTALACION SOBRE PONER APERTURA INTERNA POR BOTON FABRICADA EN ACERO," [Online]. Available: <https://tecnosinergia.com/ax-lockr.html>. [Accessed 14 01 2023].
- [58] MIT, "NestJs," [Online]. Available: <https://docs.nestjs.com/>. [Accessed 16 01 2023].
- [59] G. a. D. J. Praveen, Face Recognition: Challenges and Issues in Smart City/Environments, Vols. 791-793, 2020.

Apéndices

Apéndice A

Metodología	Ventajas	Desventajas	
CASCADA	Alta organización Ideal para proyectos pequeños Metas claras Ideal para proyectos rígidos	No hay prototipos Errores se visualizan hasta el ultimo Bastante tardado Operación hasta finalizado el proyecto	Descartada ya que no hay prototipos, es muy tardado, y no permite cambios.
ESPIRAL	No están predefinidas ninguna fase Altamente adaptable y flexible a los proyectos Genera continuos prototipos. Control máximo de errores y bugs	No se aconseja utilizarlo en pequeños sistemas Genera mucho tiempo en el desarrollo del sistema Modelo costoso Requiere experiencia en la identificación de riesgos	Descartada por problema de tiempo; de prototipado y desarrollo; además de no existir fase predefinida y altos costos.
PROTOTIPOS	Reduce el riesgo de construir productos que no satisfagan las necesidades de los usuarios Reduce costo y aumenta la probabilidad de éxito Exige disponer de las herramientas adecuadas.	El usuario ve que el prototipo funciona piensa que este es el producto terminado y no entienden que recién se va a desarrollar el software. Se puede caer en la tentación de ampliar el prototipo para construir el sistema final	Descartada solo por baja demanda en mercado laboral
DRA	Comprar puede ahorrar dinero en comparación con construir Visibilidad temprana Mayor involucramiento de los usuarios Menor codificación manual Ciclos de desarrollo más pequeños Mayor flexibilidad	Progreso más difícil de medir Menor precisión científica Más fallas (por síndrome de "codificar a lo bestia") Costo de herramientas integradas y equipo necesario Dependencia en componentes de terceros: funcionalidad de más o de menos, problemas legales	Descartada por tema de costos y menor precisión en temas de investigación; además de la codificación es de menor prioridad para mis fines.
XP	Menos, es más. Simplicidad en el diseño Combina las que han demostrado ser las mejores prácticas para desarrollar software, y las lleva al extremo. Retroalimentación concreta y frecuente del cliente, del equipo y de los usuarios finales Programación organizada Menor tasa de errores	Dificultad para organizar equipos Alta necesidad del recurso humano Es recomendable emplearlo solo en proyectos a corto plazo Altas comisiones en caso de fallar. Alta organización y normas codificación estrictas	Descartado pues no requiero de tanta inmediatez, ni tan estricta codificación.
SCRUM	Muy fácil de aprender, objetivos muy relacionados a nuestra manera diaria de trabajar El cliente puede comenzar a usar su producto rápidamente Se agiliza el proceso, ya que la entrega de valor es muy frecuente. Menor probabilidad de imprevistos, el cliente ve frecuentemente el proyecto.	Aunque sea fácil de aprender, es muy difícil poder implementarlo. Cambio profundo de cultura de la organización que debe ir desde los altos mandos hasta los clientes. Tener equipos multidisciplinarios puede ser un problema, puede tender a realizar el camino más corto para conseguir el objetivo de un Sprint, el cual no siempre es el de mayor calidad	Adaptable a mi proyecto de tesis, necesito un producto funcional inmediato para realizar pruebas continuas, se enfoca en el software antes que la documentación, se agiliza el proceso, comunicación diaria con director. Alta demanda en el mercado laboral.

Apéndice B

Columna	Instrucciones
Id	PU-01
Caso de Prueba	Cantidad de imagenes
Descripción	Captura de imágenes al detectar un rostro.
Objetivo	Checar como la cantidad de imagenes influye en la detección de caras.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Variar la cantidad de imágenes capturadas desde 50 hasta 200 para determinar la cantidad óptima.
Datos / Acciones de Entrada	Que el usuario posicione su rostro a 30 cm de la cámara y en posición frontal.
Resultado Esperado	Obtener la cantidad de imágenes solicitadas.
Procedimientos especiales requeridos	No aplica
Dependencias con otros casos de Prueba	PS-10
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PU-02
Caso de Prueba	Factor Luz
Descripción	Variación de la luz externa al capturar imágenes.
Objetivo	Checar como influye la luz externa en la detección del usuario.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Colocar luz artificial externa, al momento de detectar al usuario.
Datos / Acciones de Entrada	Que el usuario posicione su rostro a 30 cm de la cámara y en posición frontal, ahora con luz artificial.
Resultado Esperado	Aumento de la coincidencia en el reconocimiento facial.
Procedimientos especiales requeridos	Luz artificial externa.
Dependencias con otros casos de Prueba	No aplica
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PU-03
Caso de Prueba	Accesorios rostro.
Descripción	Captura de imágenes con lentes.
Objetivo	Checar como influye el uso de lentes en la detección del usuario.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el usuario porte lentes al momento de la detección.
Datos / Acciones de Entrada	Que el usuario posicione su rostro a 30 cm de la cámara y en posición frontal, usando lentes.
Resultado Esperado	Que no disminuya o se mantenga la coincidencia en la detección
Procedimientos especiales requeridos	Que la captura de imágenes previa y su entrenamiento, haya sido sin lentes.
Dependencias con otros casos de Prueba	No aplica
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PU-04
Caso de Prueba	Accesorios rostro.
Descripción	Captura de imágenes con barba.
Objetivo	Checar como influye la barba en la detección del usuario.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el usuario porte barba al momento de la detección.
Datos / Acciones de Entrada	Que el usuario posicione su rostro a 30 cm de la cámara y en posición frontal, usando barba.
Resultado Esperado	Que no disminuya o se mantenga la coincidencia en la detección
Procedimientos especiales requeridos	Que la captura de imágenes previa y su entrenamiento, haya sido sin barba.
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera.
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PU-05
Caso de Prueba	Falsos positivos.
Descripción	Captura de imágenes falsos rostros.
Objetivo	Detectar cuantas imágenes falsas de rostros se detectan y almacenan.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Revisar la cantidad de falsos rostros en las imágenes capturadas y almacenadas en el banco de imágenes.
Datos / Acciones de Entrada	Que el usuario posicione su rostro a 30 cm de la cámara y en posición frontal.
Resultado Esperado	Que la cantidad de falsos positivos en la detección de rostros sea menor al 3%.
Procedimientos especiales requeridos	No aplica
Dependencias con otros casos de Prueba	PU-01
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PU-06
Caso de Prueba	Aviso visual positivo
Descripción	Respuesta del aviso visual.
Objetivo	Checar el comportamiento del aviso visual en la pantalla.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que, en la pantalla de la Raspberry, se imprima el mensaje visual.
Datos / Acciones de Entrada	Ingresar valor positivo en la detección.
Resultado Esperado	Que el mensaje impreso en la pantalla sea "Usuario encontrado".
Procedimientos especiales requeridos	
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PU-07
Caso de Prueba	Aviso visual negativo
Descripción	Respuesta del aviso visual.
Objetivo	Checar el comportamiento del aviso visual en la pantalla.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que, en la pantalla de la Raspberry, se imprima el mensaje visual.
Datos / Acciones de Entrada	Ingresar valor negativo en la detección.
Resultado Esperado	Que el mensaje impreso en la pantalla sea "Usuario no encontrado".
Procedimientos especiales requeridos	
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PU-08
Caso de Prueba	Aviso sonoro positivo.
Descripción	Respuesta del aviso sonoro.
Objetivo	Checar el comportamiento del aviso sonoro en el parlante.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que, en el parlante de la Raspberry, se emita el mensaje sonoro.
Datos / Acciones de Entrada	Ingresar valor positivo en la detección.
Resultado Esperado	Que el mensaje sonoro en el parlante sea "Usuario encontrado".
Procedimientos especiales requeridos	
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PU-09
Caso de Prueba	Aviso sonoro negativo.
Descripción	Respuesta del aviso sonoro.
Objetivo	Checar el comportamiento del aviso sonoro en el parlante.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que, en el parlante de la Raspberry, se emita el mensaje sonoro.
Datos / Acciones de Entrada	Ingresar valor negativo en la detección.
Resultado Esperado	Que el mensaje sonoro en el parlante sea "Usuario no encontrado".
Procedimientos especiales requeridos	
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-01
Caso de Prueba	Presencia de periféricos
Descripción	Deshabilitar componentes intercalados.
Objetivo	Checar comportamiento del sistema al desconectar periféricos.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el sistema imprima en pantalla el (los) componentes desconectados.
Datos / Acciones de Entrada	Conectar los 4 periféricos: puerta, sensor proximidad, cámara y API-REST
Resultado Esperado	Que la pantalla imprima "Todos los componentes detectados"
Procedimientos especiales requeridos	Conectar todos los periféricos
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-02
Caso de Prueba	Presencia de periféricos
Descripción	Deshabilitar componentes intercalados.
Objetivo	Checar comportamiento del sistema al desconectar periféricos.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el sistema imprima en pantalla el (los) componentes desconectados.
Datos / Acciones de Entrada	Conectar 3 periféricos solamente: puerta, sensor proximidad y API-REST.
Resultado Esperado	Que la pantalla imprima "Cámara no detectada"
Procedimientos especiales requeridos	Desconectar solo la cámara
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-03
Caso de Prueba	Presencia de periféricos.
Descripción	Deshabilitar componentes intercalados.
Objetivo	Checar comportamiento del sistema al desconectar periféricos.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el sistema imprima en pantalla el (los) componentes desconectados.
Datos / Acciones de Entrada	Conectar 3 periféricos solamente: puerta, sensor proximidad y cámara.
Resultado Esperado	Que la pantalla imprima "Error de conexión en API-REST"
Procedimientos especiales requeridos	Inhabilitar la conexión a la API-REST
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-04
Caso de Prueba	Presencia de periféricos.
Descripción	Deshabilitar componentes intercalados.
Objetivo	Checar comportamiento del sistema al desconectar periféricos.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el sistema imprima en pantalla el (los) componentes desconectados.
Datos / Acciones de Entrada	Conectar 2 periféricos solamente: puerta y sensor proximidad.
Resultado Esperado	Que la pantalla imprima: "Error de conexión en API-REST", "Cámara no detectada"
Procedimientos especiales requeridos	Inhabilitar la conexión a la API-REST y desconectar la cámara.
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-05
Caso de Prueba	Presencia de periféricos.
Descripción	Deshabilitar componentes intercalados.
Objetivo	Checar comportamiento del sistema al desconectar periféricos.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el sistema imprima en pantalla el (los) componentes desconectados.
Datos / Acciones de Entrada	Conectar 3 periféricos solamente: puerta, API-REST y cámara.
Resultado Esperado	Que la pantalla imprima: "Sensor de proximidad no detectado"
Procedimientos especiales requeridos	Desconectar el sensor de proximidad.
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-06
Caso de Prueba	Presencia de periféricos
Descripción	Deshabilitar componentes intercalados.
Objetivo	Checar comportamiento del sistema al desconectar periféricos.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el sistema imprima en pantalla el (los) componentes desconectados.
Datos / Acciones de Entrada	Conectar 2 periféricos solamente: puerta y API-REST.
Resultado Esperado	Que la pantalla imprima "Sensor de proximidad no detectado", "Cámara no detectada"
Procedimientos especiales requeridos	Desconectar sensor de proximidad y cámara.
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-07
Caso de Prueba	Presencia de periféricos
Descripción	Deshabilitar componentes intercalados.
Objetivo	Checar comportamiento del sistema al desconectar periféricos.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el sistema imprima en pantalla el (los) componentes desconectados.
Datos / Acciones de Entrada	Conectar 2 periféricos solamente: puerta y cámara.
Resultado Esperado	Que la pantalla imprima "Sensor de proximidad no detectado", "Error de conexión en API-REST"
Procedimientos especiales requeridos	Desconectar sensor de proximidad y deshabilitar conexión en API-REST.
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-08
Caso de Prueba	Presencia de periféricos
Descripción	Deshabilitar componentes intercalados.
Objetivo	Checar comportamiento del sistema al desconectar periféricos.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el sistema imprima en pantalla el (los) componentes desconectados.
Datos / Acciones de Entrada	Conectar solamente la puerta.
Resultado Esperado	Que la pantalla imprima "Sensor de proximidad no detectado", "Error de conexión en API-REST", "Cámara no detectada".
Procedimientos especiales requeridos	Desconectar sensor de proximidad, deshabilitar conexión en API-REST y desconectar la cámara.
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-09
Caso de Prueba	Presencia de periféricos
Descripción	Deshabilitar componentes intercalados.
Objetivo	Checar comportamiento del sistema al desconectar periféricos.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el sistema imprima en pantalla el (los) componentes desconectados.
Datos / Acciones de Entrada	Conectar 3 periféricos solamente: sensor de proximidad, API-REST y cámara.
Resultado Esperado	Que la pantalla imprima "Puerta no detectada".
Procedimientos especiales requeridos	Desconectar puerta.
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-10
Caso de Prueba	Presencia de periféricos
Descripción	Deshabilitar componentes intercalados.
Objetivo	Checar comportamiento del sistema al desconectar periféricos.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el sistema imprima en pantalla el (los) componentes desconectados.
Datos / Acciones de Entrada	Conectar 2 periféricos solamente: sensor de proximidad y API-REST.
Resultado Esperado	Que la pantalla imprima "Puerta no detectada", "Cámara no detectada"
Procedimientos especiales requeridos	Desconectar puerta y cámara.
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-11
Caso de Prueba	Presencia de periféricos
Descripción	Deshabilitar componentes intercalados.
Objetivo	Checar comportamiento del sistema al desconectar periféricos.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el sistema imprima en pantalla el (los) componentes desconectados.
Datos / Acciones de Entrada	Conectar 2 periféricos solamente: sensor de proximidad y cámara.
Resultado Esperado	Que la pantalla imprima "Puerta no detectada", "Error de conexión en API-REST"
Procedimientos especiales requeridos	Desconectar puerta y deshabilitar la conexión de la API-REST.
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-12
Caso de Prueba	Presencia de periféricos
Descripción	Deshabilitar componentes intercalados.
Objetivo	Checar comportamiento del sistema al desconectar periféricos.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el sistema imprima en pantalla el (los) componentes desconectados.
Datos / Acciones de Entrada	Conectar solamente el sensor de proximidad.
Resultado Esperado	Que la pantalla imprima "Puerta no detectada", "Error de conexión en API-REST", "Cámara no detectada".
Procedimientos especiales requeridos	Desconectar puerta, cámara y deshabilitar la conexión de la API-REST.
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-13
Caso de Prueba	Presencia de periféricos
Descripción	Deshabilitar componentes intercalados.
Objetivo	Checar comportamiento del sistema al desconectar periféricos.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el sistema imprima en pantalla el (los) componentes desconectados.
Datos / Acciones de Entrada	Conectar 2 periféricos solamente: cámara y API-REST.
Resultado Esperado	Que la pantalla imprima "Puerta no detectada", "Sensor de proximidad no detectado"
Procedimientos especiales requeridos	Desconectar puerta y sensor de proximidad.
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-14
Caso de Prueba	Presencia de periféricos
Descripción	Deshabilitar componentes intercalados.
Objetivo	Checar comportamiento del sistema al desconectar periféricos.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el sistema imprima en pantalla el (los) componentes desconectados.
Datos / Acciones de Entrada	Conectarse solamente a la API-REST.
Resultado Esperado	Que la pantalla imprima "Puerta no detectada", "Sensor de proximidad no detectado", "Cámara no detectada".
Procedimientos especiales requeridos	Desconectar puerta, cámara y sensor de proximidad.
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-15
Caso de Prueba	Presencia de periféricos
Descripción	Deshabilitar componentes intercalados.
Objetivo	Checar comportamiento del sistema al desconectar periféricos.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el sistema imprima en pantalla el (los) componentes desconectados.
Datos / Acciones de Entrada	Conectar solamente la cámara.
Resultado Esperado	Que la pantalla imprima "Puerta no detectada", "Sensor de proximidad no detectado", "Error de conexión en la API-REST".
Procedimientos especiales requeridos	Desconectar puerta, sensor de proximidad y deshabilitar la conexión en la API-REST..
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-16
Caso de Prueba	Presencia de periféricos
Descripción	Deshabilitar componentes intercalados.
Objetivo	Checar comportamiento del sistema al desconectar periféricos.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Que el sistema imprima en pantalla el (los) componentes desconectados.
Datos / Acciones de Entrada	No conectar ningún periférico.
Resultado Esperado	Que la pantalla imprima "Puerta no detectada", "Sensor de proximidad no detectado", "Cámara no detectada", "Error de conexión en la API-REST".
Procedimientos especiales requeridos	Desconectar todos los periféricos.
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-17
Caso de Prueba	Compatibilidad entre datos
Descripción	Lectura de datos de internet.
Objetivo	Correcta lectura de fecha y hora desde internet.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Leer la fecha en el formato DIA / MES / AÑO y la hora en 24 horas, con sus minutos solamente.
Datos / Acciones de Entrada	Captar fecha y hora de internet.
Resultado Esperado	Impresión en pantalla de la fecha y hora captadas de internet.
Procedimientos especiales requeridos	Conexión a internet.
Dependencias con otros casos de Prueba	No aplica.
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-18
Caso de Prueba	Compatibilidad entre datos
Descripción	Escritura de los datos entre módulos.
Objetivo	Checar compatibilidad de formatos entre la API-REST y el módulo registro de asistencia.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Escribir y leer las fechas en el formato DIA / MES / AÑO y la hora en 24 horas, con sus minutos solamente.
Datos / Acciones de Entrada	Captar fecha y hora de internet.
Resultado Esperado	Impresión en pantalla de la fecha y hora captadas de internet y un mensaje en pantalla "Formato compatible"
Procedimientos especiales requeridos	Una base de datos de prueba con una tabla relacional, para probar la conexión de la API-REST.
Dependencias con otros casos de Prueba	PI-17
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-19
Caso de Prueba	Flujo de información.
Descripción	Como fluye los datos entre módulos.
Objetivo	Comprobar el correcto funcionamiento entre la API-REST y el módulo registro de asistencia.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Comparar las fechas y tiempos entre módulos.
Datos / Acciones de Entrada	Fecha valida ingresada de manera manual con el formato día/mes/año
Resultado Esperado	Mensaje impreso en la pantalla "Fecha acorde"
Procedimientos especiales requeridos	En el módulo registro de asistencia, previamente establecido, una fecha y hora idénticas a las ingresadas en las acciones de entrada.
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PI-20
Caso de Prueba	Flujo de información.
Descripción	Como fluye los datos entre módulos.
Objetivo	Comprobar el correcto funcionamiento entre la API-REST y el módulo registro de asistencia.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Comparar las fechas y tiempos entre módulos.
Datos / Acciones de Entrada	Fecha incorrecta, ingresada de manera manual con el formato día/mes/año
Resultado Esperado	Mensaje impreso en la pantalla "Fuera de tiempo"
Procedimientos especiales requeridos	En el módulo registro de asistencia, previamente establecido, una fecha y hora diferentes a las ingresadas en las acciones de entrada.
Dependencias con otros casos de Prueba	
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PS-01
Caso de Prueba	Apertura puerta
Descripción	Abrir la puerta según sea el caso.
Objetivo	Validar la correcta apertura de la puerta.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Ceder / Negar accesos en la puerta principal.
Datos / Acciones de Entrada	Imagen capturada, con un usuario ajeno al sistema.
Resultado Esperado	Mensaje impreso y sonoro en la pantalla "Usuario no encontrado". No ceder el acceso de la puerta.
Procedimientos especiales requeridos	Iluminación externa.
Dependencias con otros casos de Prueba	No aplica.
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera.
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PS-02
Caso de Prueba	Apertura puerta
Descripción	Abrir la puerta según sea el caso.
Objetivo	Validar la correcta apertura de la puerta.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Ceder / Negar accesos en la puerta principal.
Datos / Acciones de Entrada	Imagen capturada, con un usuario del sistema.
Resultado Esperado	Mensaje impreso y sonoro en la pantalla "Bienvenido + el nombre del usuario encontrado". Ceder el acceso de la puerta.
Procedimientos especiales requeridos	Iluminación externa.
Dependencias con otros casos de Prueba	No aplica.
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera.
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PS-03
Caso de Prueba	Tiempo de respuesta.
Descripción	Cuanto tiempo tarda el sistema.
Objetivo	Validar el requisito de tiempo establecido en SRS.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Cronometrar el tiempo de respuesta desde el sensor hasta la respuesta final.
Datos / Acciones de Entrada	Un usuario del sistema se acerca para su activación.
Resultado Esperado	Mensaje visual en la pantalla con el tiempo cronometrado.
Procedimientos especiales requeridos	Una función de cronómetro en el sistema que se active al activar el sensor de proximidad.
Dependencias con otros casos de Prueba	No aplica.
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera.
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PS-04
Caso de Prueba	Vulneración
Descripción	Vulnerar el sistema con fotografías.
Objetivo	Validar el requisito de confiabilidad, establecido en SRS.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Seguridad en el sistema.
Datos / Acciones de Entrada	Inicializar el sistema al acercarse, colocar una fotografía en la cámara.
Resultado Esperado	En la pantalla se muestre un rostro detectado, pero no tome la fotografía.
Procedimientos especiales requeridos	Luz externa.
Dependencias con otros casos de Prueba	No aplica.
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera.
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PS-05
Caso de Prueba	Vulneración
Descripción	Vulnerar el sistema con una imagen en un smartphone.
Objetivo	Validar el requisito de confiabilidad, establecido en SRS.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Seguridad en el sistema.
Datos / Acciones de Entrada	Inicializar el sistema al acercarse, colocar un smartphone con una imagen de un usuario registrado.
Resultado Esperado	En la pantalla se muestre un rostro detectado, pero no tome la fotografía.
Procedimientos especiales requeridos	Luz externa.
Dependencias con otros casos de Prueba	No aplica.
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera.
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PS-06
Caso de Prueba	Vulneración
Descripción	Vulnerar el sistema con una imagen en una tablet.
Objetivo	Validar el requisito de confiabilidad, establecido en SRS.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Seguridad en el sistema.
Datos / Acciones de Entrada	Inicializar el sistema al acercarse, colocar una Tablet con una imagen de un usuario registrado.
Resultado Esperado	En la pantalla se muestre un rostro detectado, pero no tome la fotografía.
Procedimientos especiales requeridos	Luz externa.
Dependencias con otros casos de Prueba	No aplica.
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera.
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PA-01
Caso de Prueba	Horario
Descripción	Checar el sistema fuera de horario.
Objetivo	Validar la confiabilidad en el sistema.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Comportamiento el sistema fuera de horario.
Datos / Acciones de Entrada	Ingresar hora diferente a la estipulada, de manera manual.
Resultado Esperado	No apertura de la puerta. Mensaje visual y sonoro "No coincide su clase + Nombre del usuario", "Registro de asistencia exitoso"
Procedimientos especiales requeridos	Captar la hora de otra fuente diferente a internet.
Dependencias con otros casos de Prueba	No aplica
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera.
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PA-02
Caso de Prueba	Fecha
Descripción	Checar el sistema fuera de fecha.
Objetivo	Validar la confiabilidad en el sistema.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Comportamiento el sistema fuera de fecha.
Datos / Acciones de Entrada	Ingresar fecha diferente a la estipulada, de manera manual.
Resultado Esperado	No apertura de la puerta. Mensaje visual y sonoro "No coincide su clase + Nombre del usuario", "Registro de asistencia exitoso"
Procedimientos especiales requeridos	Captar la fecha de otra fuente diferente a internet.
Dependencias con otros casos de Prueba	No aplica
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera.
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PA-03
Caso de Prueba	Rendimiento
Descripción	Rendimiento del sistema con 50 usuarios.
Objetivo	Validar el rendimiento del sistema
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Conocer la tasa de rendimiento del sistema.
Datos / Acciones de Entrada	Un usuario del sistema se acerca, lo inicializa y espera. Así 50 veces con diferentes usuarios.
Resultado Esperado	Apertura de la puerta. Mensaje visual y sonoro "Bienvenido + nombre del usuario". Sumar usuario a la función SumarUsuario
Procedimientos especiales requeridos	Función SumarUsuario, para sumar cada usuario aceptado y mostrar la información.
Dependencias con otros casos de Prueba	No aplica
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera.
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PA-04
Caso de Prueba	Aviso asistencia
Descripción	Registro efectivo de asistencia, a través de un aviso sonoro.
Objetivo	Validar el aviso sonoro efectivo al registrar asistencia.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Avisar al usuario si se registró su asistencia o no, de manera sonora.
Datos / Acciones de Entrada	Un usuario al acercarse al sistema.
Resultado Esperado	Registro efectivo de asistencia del usuario. Mensaje sonoro "Asistencia registrada correctamente"
Procedimientos especiales requeridos	Que el módulo de reconocimiento facial envíe un valor positivo.
Dependencias con otros casos de Prueba	No aplica
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera.
Última Fecha de Estado	
Observaciones	

Columna	Instrucciones
Id	PA-06
Caso de Prueba	Aviso asistencia.
Descripción	Registro efectivo de asistencia, a través de un aviso visual.
Objetivo	Validar el aviso visual efectivo al registrar asistencia.
Fecha	10 / 12 / 2020
Funcionalidad / Característica	Avisar al usuario si se registró su asistencia o no, de manera visual.
Datos / Acciones de Entrada	Un usuario al acercarse al sistema.
Resultado Esperado	Registro efectivo de asistencia del usuario. Mensaje sonoro "Asistencia registrada correctamente"
Procedimientos especiales requeridos	Que el módulo de reconocimiento facial envíe un valor positivo.
Dependencias con otros casos de Prueba	No aplica
Información para el seguimiento	
Resultado Obtenido	
Estado	En espera.
Última Fecha de Estado	
Observaciones	

Apéndice C

Requerimientos

SRF-01	El usuario al acercarse a 50cm del sistema como mínimo, el sistema inicializará mediante el encendido de la pantalla. Caso contrario el sistema se mantiene en modo de espera.
Inicialización por aproximación	

SRF-02	Después de SRF-01, el sistema debe detectar un rostro a través de la cámara.
Detección de rostro	

SRF-03	Después de SRF-02, el sistema debe tomar una fotografía al usuario, por medio de la cámara.
Captura imagen	

SRF-04	Después de SRF-03, se debe de reconocer, si el usuario es profesor en el sistema, por medio del procesamiento de la imagen.
Reconocimiento del usuario	

SRF-05	Después de SRF-04, el sistema debe de registrar la asistencia del usuario a través de una API-REST, en otro sistema llamado “Reloj Checador”, según corresponda a su aula en tiempo y forma.
Registro de asistencia	

SRF-06	Después de SRF-04 y SRF-05, el sistema determina si abrir o no la puerta a través de la cerradura.
Decisión acceso al aula	

SRF-07	Después de SRF-06, el sistema emitirá un mensaje sonoro y visual de la situación del usuario en el proceso.
Aviso de efectividad	

SRNF-01	El sistema debe detectar la presencia e interconexión de los distintos componentes y partes del sistema (cámara, cerradura, Raspberry y API-REST) al inicializar. Cualquier anomalía encontrada en el sistema avisará a través de la pantalla, mencionando cual dispositivo no se encuentra, de manera visual.
Debug interno	

SRNF-02	El sistema utilizará este lenguaje de programación como base en los distintos algoritmos y funcionalidades de software en el sistema.
Librería OpenCV	

SRNF-03	El sistema de debe de desarrollar en lenguaje Python.
Lenguaje Python	

SRR-01	El sistema deberá alcanzar como mínimo un 86 % de efectividad en el proceso de reconocimiento facial. Cualquier tasa por debajo de este umbral no permitirá la apertura de la puerta ni el registro de asistencia.
Efectividad	

SRR-02	El sistema deberá operar en condiciones externas; contemplando agentes climatológicos en su diseño e implementación, tales como protección a lluvia, sol, etc.
Intemperie	

SRR-03	El sistema dará respuesta (desde la toma de la fotografía, hasta la apertura de la cerradura) con un periodo no máximo de 5 segundos.
Respuesta	

SyU-01	El sistema debe inicializar al detectar la presencia del usuario por medio del sensor de proximidad.
Activación sistema	

SyU-02	El usuario solamente debe permanecer de frente a la cámara para lograr fidelidad en la fotografía.
Toma de fotografía	

SyHw-04	El sistema requiere una pantalla de 3.5 pulgadas para comunicarse con el usuario.
Pantalla Raspberry	