



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Dpto. de Ingeniería de Sistemas y Automática

Sistema empotrado de control de acceso al uso de activos
mediante reconocimiento de caras.

Trabajo Fin de Máster

Máster Universitario en Automática e Informática Industrial

AUTOR/A: Fontalba Roca, Marc

Tutor/a: Simó Ten, José Enrique

CURSO ACADÉMICO: 2023/2024

SISITEMA EMPOTRADO DE CONTROL DE ACCESO AL USO DE ACTIVOS MEDIANTE RECONOCIMIENTO DE CARAS

Autor
Fontalba Roca, Marc

Director: Simó Ten, José Enrique

TRABAJO FIN DE MASTER
MASTER DE AUTOMÁTICA E INFORMÁTICA
INDUSTRIAL
UNIVERSITAT POLITÈCNICA DE VALÈNCIA



**UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA**

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Valencia, 15/07/2024

Dedicatoria

A mis padres y hermanos por haberme apoyado durante tantos años. Y a María José por aguantarme y quererme pese a mis rarezas.

AGRADECIMIENTOS

En primer lugar, me gustaría agradecer a mis padres y hermanos por apoyarme desde que tengo uso de razón. También agradecer a María José por su cariño y comprensión estos cinco años, en los momentos buenos, malos, caóticos, maniacos etc. Además, agradecer a mis compañeros y amigos que han estado ahí tanto en el máster como fuera. Gracias al Kerry, mi pastelito de melocotón y miel, por estos años aprendiendo y haciendo el idiota juntos. Gracias a Abel por enseñarme que con únicamente un boli puedes afrontar cualquier cosa. Y a Polit que bueno ha estado ahí supongo. Por último, gracias a Nena Daconte por su canción “en que estrella estará” que ha sonado en bucle mientras hacia el trabajo. Muchas gracias ... Finalmente solo queda decir que el Máster siempre fue y será los amigos que hacemos por el camino.

TABLA DE CONTENIDO

1.	INTRODUCCION.....	11
2.	JUSTIFICACION.....	12
3.	OBJETIVOS.....	13
3.1	Objetivo general	13
3.2	Objetivos específicos.....	13
4.	ESTADO ACTUAL.....	14
4.1	Fundamentos del reconocimiento facial	14
4.1.1	Introducción a los sistemas de reconocimiento facial	14
4.1.2	Detección facial	15
4.1.3	Extracción de características	16
4.1.4	Verificación e identificación.....	16
4.1.5	Casos de uso y aplicaciones	16
4.2	Sistemas embebidos	17
4.2.1	Introducción a los sistemas embebidos	17
4.2.2	Casos de uso y aplicaciones	17
5.	METODOLOGIA EXPERIMENTAL	18
5.1	Requisitos del sistema.....	18
5.1.1	Requisitos funcionales.....	18
5.1.2	Requisitos no funcionales.....	19
5.2	Diseño general del sistema	19
5.3	Eventualidades	20
5.4	Descripción del hardware	21
5.4.1	Microordenador.....	21
5.4.2	Teclado matricial.....	22
5.4.3	Cámara.....	23
5.4.4	Circuitería	24
5.5	Descripción de herramientas software	24
5.5.1	Lenguaje de programación y justificación	24
5.5.1.1	C++.....	24
5.5.1.2	Python	24
5.5.2	Base de datos MySQL.....	25
5.5.3	Librerías.....	25

5.6	Fases del desarrollo	26
5.6.1	Desarrollo iterativo.....	26
5.6.1.1	Demostrador del motor del sistema.....	26
5.6.1.2	Demostrador del sistema embebido.....	29
5.6.1.3	Demostrador del sistema completo	31
5.6.2	Pruebas unitarias.....	32
5.6.2.1	Desarrollo del módulo clasificador.....	32
5.6.2.2	Desarrollo del módulo de la base de datos.	33
5.6.2.3	Desarrollo del módulo de comunicación por sockets Python y C++.	33
5.6.2.4	Módulo de carga del modelo y reconocimiento en el front-end.	36
5.6.2.5	Módulo de detección de caras.....	36
5.6.2.6	Módulo de extracción de características de las caras.	36
5.6.2.7	Módulo de adquisición de imágenes.	37
5.6.2.8	Módulo de la clase teclado.	38
5.6.2.9	Módulo de la clase led.....	38
6.	RESULTADOS	39
6.1	Arquitectura hardware	39
6.1.1	Diagrama de conexiones	39
6.1.2	Montaje.....	40
6.2	Diagramas UML	45
6.2.1	Diagrama de casos de uso	45
6.2.2	Casos de uso.....	47
6.2.3	Diagrama de interacción.....	54
6.2.4	Diagrama de clases	57
6.3	Instalacion y despliegue	58
6.3.1	Compilación	58
6.3.2	Instalación.....	58
6.3.3	Despliegue.....	59
6.4	Instrucciones de uso.....	59
6.4.1	Manual de USUARIO.....	59
6.4.1.1	Introducción.....	59
6.4.1.2	Descripción del sistema.....	59
6.4.1.3	Operación diaria	60

6.4.1.4	Uso del teclado.....	61
6.4.1.5	Señal de leds.....	61
6.4.1.6	Resolución de problemas comunes.....	62
6.4.2	Manual de ADMINISTRADOR.....	63
6.4.2.1	Introducción.....	63
6.4.2.2	Rutas del sistema.....	63
6.4.2.3	Configuración inicial.....	65
6.4.2.4	Mantenimiento y solución de problemas detallados.....	66
7.	ANALISIS Y DISCUSION DE RESULTADOS.....	68
7.1	Rendimiento.....	68
7.2	Relacion con los requisistos del sistema.....	69
7.3	Limitaciones.....	70
8.	CONCLUSIONES.....	72
9.	RECOMENDACIONES Y TRABAJOS FUTUROS.....	73
10.	REFERENCIAS BIBLIOGRÁFICAS.....	74
11.	ANEXOS.....	76
11.1	Relación del trabajo con los objetivos de desarrollo sostenible de la agenda 2030.....	76
11.2	Planos.....	77
11.3	Presupuestos.....	81
11.4	Codigos.....	84

LISTA DE TABLAS

Tabla 1: Requisitos funcionales del sistema.....	18
Tabla 2: Requisitos no funcionales del sistema.....	19
Tabla 3: Tabla comparativa del microordenador.....	22
Tabla 4: Tabla comparativa de teclados matriciales.....	23
Tabla 5: Tabla comparativa de cámaras.....	23
Tabla 6: Estado de las funcionalidades en el demostrador 1.....	28
Tabla 7: Estado de las funcionalidades del demostrador 2.....	30
Tabla 8: Estado de las funcionalidades en el demostrador 3.....	31
Tabla 9: Ejemplo de tabla users para tres usuarios registrados.....	33
Tabla 10: Ejemplo de la tabla maquinaria para tres máquinas registradas.....	33
Tabla 11: Relación de conexiones en el nodo front-end.....	40
Tabla 12: Caso de uso RF-01, Usuario solicita máquina.....	47
Tabla 13: Caso de uso RF-02, Usuario ingresa PIN.....	47
Tabla 14: Caso de uso RF-03, Administrador realiza el mantenimiento de imágenes.....	48
.....	48
Tabla 15: Caso e uso RF-04, Administrador realiza el mantenimiento de la base de datos.....	48
Tabla 16: Caso de uso RF-05, Computador configura los puertos TCP.....	49
Tabla 17: Caso de uso RF-06, Computador recibe imágenes por el socket TCP2...	49
Tabla 18: Caso de uso RF-07, Computador lanza el entrenamiento de la red.....	50
Tabla 19: Caso de uso RF-08, Raspi lee la solicitud de maquinaria.....	50
Tabla 20: Caso de uso RF-09, Raspi toma una captura de la cara del usuario.....	51
Tabla 21: Caso de uso RF-10, Raspi realiza la autenticación del usuario.....	51
Tabla 22: Caso de uso RF-11, Raspi realiza el reconocimiento facial.....	52
Tabla 23: Caso de uso RF-12, Raspi realiza la lectura del PIN.....	53
Tabla 24: Señales luminosas del prototipo junto con su duración, evento previo y significado.....	61

LISTA DE FIGURAS

Figura 1: Fases del proceso de reconocimiento facial. Extraído de Bakshi, U & Singhal, R. (2014).....	15
Figura 2: Diagrama del diseño del sistema. Elaborado mediante Draw.io.....	20
Figura 3: : Diagrama del diseño del sistema, frente a una eventualidad. Elaborado mediante Draw.io.....	21
Figura 4: Teclados de tipo membrana, tecla y botonera, ordenado de izquierda a derecha. Imágenes extraídas del catálogo de Diatronic (2024) y Microlog (2024).	22
Figura 5: Diagrama de flujo del motor del sistema. Elaboración propia empleando Drwa.io.	27
Figura 6: Diagrama de secuencia de la comunicación en el socket TCP0. Elaboración propia mediante Drwa.io.	34
Figura 7: Diagrama de secuencia de la comunicación en el socket TCP1. Elaboración propia mediante Drwa.io.	34
Figura 8: Diagrama de secuencia de la comunicación en el socket TCP2. Elaboración propia mediante Drwa.io.	35
Figura 9: Diagrama de secuencia de la comunicación en el socket TCP3. Elaboración propia mediante Drwa.io.	35
Figura 10: Etapas del proceso de detección de caras. Elaboración propia mediante Draw.io.	36
Figura 11: Proceso de la extracción de características. Elaboración propia en Drwa.io empleando la imagen de la arquitectura ResNet de Towards Data Science (Yosinski, 2020).	37
Figura 12: Esquema de conexiones en el módulo front-end. Elaboración propia empleado KiCAD.	39
Figura 13: Resultado del paso 1: a la izquierda componentes implicados y a la derecha el resultado de la operación marcando la ubicación y métrica de la tornillería. Elaboración propia empleando Paint.	40
Figura 14: Resultado del paso 2; a la izquierda componentes implicado y a la derecha resultado de la operación. Elaboración propia utilizando Paint.	41
Figura 15: Resultado del paso 3: A la izquierda los componentes implicados en la operación y a la derecha el resultado de esta. Elaboración propia en Paint.	42
Figura 16: Resultado del paso 4: a la izquierda los componentes implicados y a la derecha el resultado de la operación junto con la ubicación de la tornillería y métrica empleada. Elaboración propia usando Paint.	42
Figura 17: Elementos implicados previos al paso 5. Elaboración propia en Paint. ...	43
Figura 18: Resultado de la operación: a la izquierda la cara delantera y a la derecha la cara trasera con la ubicación y métrica de la tornillería empleada. Elaboración propia en Paint.	43
Figura 19: Resultado del paso 6: conexionado de la electrónica de la tapa superior y la inferior. Elaboración en Paint.	44
Figura 20: Prototipo montado al completo con la ubicación de la tornillería y métrica requerida en el paso 7. Elaboración propia en Paint.	45
Figura 21: Diagrama de casos de usos del sistema con actores Administrador, Usuario, Computador, Base de Datos y Raspi. Elaboración propia hecha en Draw.io.	46

Figura 22: Diagrama de interacción del sistema, parte 1. Elaboración propia mediante Drwa.io.....	54
Figura 23: Diagrama de interacción del sistema, parte 2. Elaboración propia mediante Drwa.io.....	55
Figura 24: Diagrama de interacción del sistema, parte 3. Elaboración propia mediante Drwa.io.....	56
Figura 25: Diagrama de interacción del sistema, parte 4. Elaboración propia mediante Drwa.io.....	57
Figura 26: Diagrama de clases del sistema. Elaboración propia en Draw.io.....	58
Figura 27: Elementos fundamentales del prototipo. Elaboración propia empleado Paint.....	60
Figura 28: Teclado matricial señalizando el rol de los botones. Editado mediante Paint.....	61
Figura 29: Árbol de archivos de la aplicación correspondiente al front-end. Elaboración propia empleando Draw.io.....	64
Figura 30: Árbol de archivos de la aplicación correspondiente al back-end. Elaboración propia empleando Draw.io.....	64
Figura 31 : Ejemplo del archivo Config.txt del back-end. Captura de pantalla archivo usando nano.....	65
Figura 32:: Ejemplo del archivo Config.txt del front-end. Captura de pantalla archivo usando nano.....	66

RESUMEN EXTENDIDO

El siguiente proyecto consiste en la realización, de un sistema empotrado basado en la plataforma RaspberrPi para la identificación y el control de usuarios de maquinaria. La finalidad del sistema desarrollado es el control de acceso a determinados activos, como por ejemplo maquinaria en una línea de producción, en escenarios en los que la seguridad no es un factor crítico pero el registro del acceso, así como la agilidad de este son factores importantes

En este proyecto se han desarrollado las soluciones necesarias tanto software como de hardware que permiten obtener un prototipo final totalmente funcional atendiendo a los requisitos planteados en la fase inicial del proyecto.

El sistema constará de un teclado y una cámara y se comunicará con un servidor que mantendrá la información de acceso a los diferentes activos, así como la información de entrenamiento del reconocimiento de caras. Inicialmente cada usuario tendrá un pin de acceso que lo identificará. Cada vez que el sistema identifique a un usuario, registrará su cara para aprenderla, de manera que llegue un momento en que no sea necesario que el usuario introduzca el PIN para acceder.

Palabras clave: Sistemas empotrados, procesamiento de imágenes, redes neuronales, reconocimiento de caras, control de accesos.

1. INTRODUCCION

Este proyecto Trabajo Fin de Máster surge con la finalidad de expandir los conocimientos adquiridos en el campo del desarrollo de sistemas embebidos, el aprendizaje automático y la programación. Principalmente, debido a que siempre me han resultado de interés los sistemas empotrados y la programación.

El proyecto surge con el objetivo de efectuar el control del acceso a un determinado activo. Así como de mantener un registro del usuario y la maquinaria facilitando la trazabilidad y procesamiento de los datos de la planta.

Por otro lado, para la elaboración del prototipo, se optó por seguir con un desarrollo iterativo en el que se fueron construyendo los diferentes módulos del sistema para finalmente integrarlos todos. De esta forma, se pudieron poner a prueba tanto de manera conjunta como individual eliminando así una mayor cantidad de fallos en el sistema.

El sistema resultante está compuesto por un nodo de Front-end, correspondiente a la Raspberry Pi y el Back-end que corresponde con un computador. El primero, se encarga de la interacción con el usuario. Mientras que el último, actúa como base de datos de los usuarios y maquinaria, además de entrenar la red neuronal encargada del reconocimiento facial.

Finalmente, se ha organizado el contenido del documento iniciando con los objetivos, posteriormente se ha elaborado una explicación del estado del arte, continuando con las metodologías empleadas (entrando en detalle que elementos componen el sistema y que etapas han compuesto su desarrollo), posteriormente se ha realizado la descripción del funcionamiento del sistema empotrado, así como los manuales de usuario y administrador, finalizando con una conclusión donde se evalúan si se han cumplido los objetivos así como posibles mejoras o futuras líneas para ampliar el proyecto.

2. JUSTIFICACION

El presente proyecto tiene por objetivo desarrollar los conocimientos adquiridos en el máster. A su vez, se busca diseñar un sistema capaz de llevar a cabo el control de acceso a un servicio de manera automática. Permitiendo simplificar y mejorar la fluidez para acceder al activo. Así mismo, el sistema mantiene un registro de la información de los accesos y estado de la maquinaria.

Dentro del marco actual de la industria 4.0 donde la digitalización de las industrias esta al orden del día. El proyecto queda justificado, debido a su potencial, ya que presenta una manera de mantener la información de los usuarios y maquinaria, mejorando la trazabilidad de la planta.

Además, el proyecto ofrece una base sólida para un posterior análisis de los datos almacenados. Por ejemplo, relacionando el índice de usuario/producción de máquina, pudiendo determinar que usuario es mejor para cada máquina. Optimizando así la asignación de tareas y manejo de recursos en la producción.

En resumen, el proyecto abre la posibilidad de mejorar la producción gracias al manejo y procesamiento de la información, alineándose con los principios fundamentales de la Industria 4.0.

3. OBJETIVOS

3.1 OBJETIVO GENERAL

El presente proyecto tiene por objetivo el desarrollo e implementación de un prototipo de sistema embebido para el control de accesos a un determinado servicio mediante reconocimiento facial y aprendizaje automático. Además, se tendrá un registro de los accesos y de los usuarios mediante el uso de una base de datos.

3.2 OBJETIVOS ESPECÍFICOS

Para el correcto cumplimiento del proyecto será necesario definir una serie de objetivos específicos:

- Se deberá recopilar información sobre clasificadores, redes neuronales, así como diferentes estrategias de reconocimiento facial, además del software a utilizar. También, se deberá probar diversos clasificadores para un 'dataset' de prueba que emule el que se podría tener en un caso real.
- Se deberá recopilar información sobre las dependencias, librerías y bases de datos que se utilizarán.
- Se deberá realizar una primera etapa de planteamiento del problema y estructuración de los dos nodos que componen el sistema final, así como la comunicación entre ambos.
- Se deberá diseñar el sistema siguiendo con la metodología del desarrollo iterativo empezando con la elaboración de demostradores que represente partes del sistema final. Además, se deberá poner a prueba tanto los demostradores de forma individual como el sistema completo con todas las funcionalidades integradas.

4. ESTADO ACTUAL

4.1 FUNDAMENTOS DEL RECONOCIMIENTO FACIAL

4.1.1 Introducción a los sistemas de reconocimiento facial

Los sistemas de reconocimiento facial quedan incluidos dentro de la categoría de reconocimiento biométrico. Estas metodologías de identificación toman un papel cada vez más significativo en ámbitos relacionados con la seguridad, administración e industrial. En los métodos de biometría, el reconocimiento facial ha atraído un creciente interés debido a que permite identificar y verificar la identidad de individuos de una manera discreta y no intrusiva.

En su análisis crítico, Shepley (2018) realiza un estudio de la evolución y avance de esta tecnología, Atendiendo a la historia del reconocimiento facial, esta tecnología inicia su desarrollo en la década de los 1960 y 1970 con los primeros métodos de reconocimiento basado en la geometría de los rasgos del rostro. Estos primeros acercamientos no tuvieron mucho éxito debido a la dificultad que presentaban la variación en parámetros de la imagen tales como la perspectiva, la rotación, la iluminación algo común en aplicaciones de visión por computador. En los años 80 se consiguen avances con el desarrollo de algoritmos más complejos que aumentaron la precisión en el reconocimiento. Es en la década de 1990 cuando se producen grandes progresos en el campo de la detección y reconocimiento facial, gracias a la aplicación de los Eigenfaces, vectores propios utilizados en el reconocimiento facial. Con la llegada del nuevo milenio esta tecnología fue ganando relevancia en aplicaciones de seguridad impulsando su desarrollo gracias la comercialización de estos sistemas biométricos. Posteriormente con la llegada del desarrollo en el aprendizaje profundo, especialmente en las redes convolucionales se alcanzó un impulso en los sistemas de reconocimiento facial mejorando en gran medida la precisión.

Finalmente, los sistemas de reconocimiento facial podrían ser definidos como sistemas de reconocimiento biométrico que mediante software permiten identificar o verificar sujetos en base al rostro del individuo. Estos sistemas cuentan con una serie de procesos o pasos fundamentales para conseguir resolver el problema de identificación o verificación. Estos pasos, consisten en la detección facial en la imagen, la extracción de características de la cara detectada y el propio reconocimiento. La metodología para resolver el problema de reconocimiento se puede observar en la Figura 1 (Bakshi, U & Singhal, R, 2014).

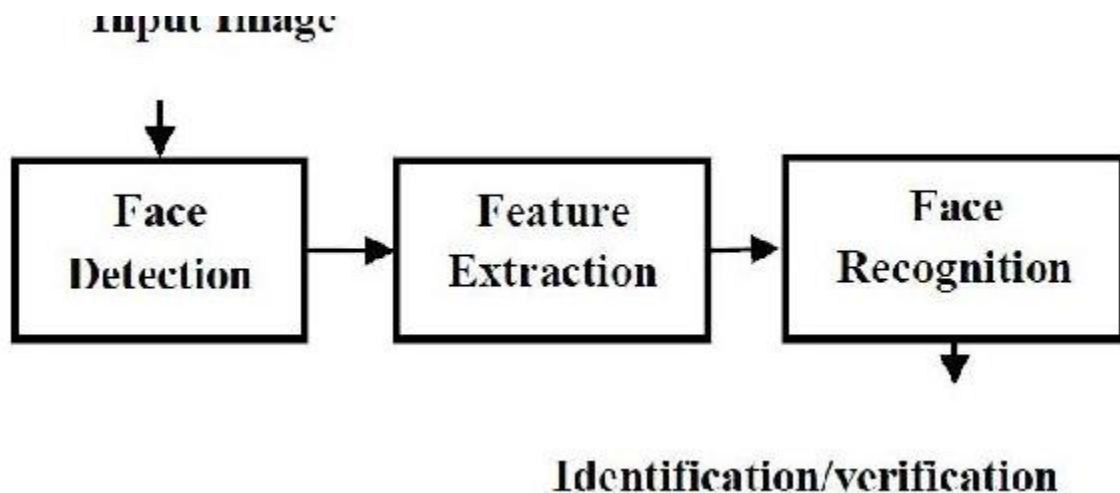


Figura 1: Fases del proceso de reconocimiento facial. Extraído de Bakshi, U & Singhal, R. (2014).

4.1.2 Detección facial

La detección de rostros es un paso fundamental para el reconocimiento y verificación facial. En su análisis crítico, Shepley (2018) realiza un estudio de la evolución de los métodos de detección facial. Inicialmente, este campo se presentaba dificultades para obtener altas precisiones y un buen desempeño de los algoritmos en escenarios no controlados. Un hito en la evolución de esta tecnología se produce con el método de detección de rostros basado en boosting de Viola-Jones, posteriormente con el desarrollo de otras técnicas de extracción de características como el HoGs, SIFT, LBP se consiguió un avance significativo. Sin embargo, estos métodos seguían presentando ciertas limitaciones en entornos no controlados que dieron lugar a otros métodos de detección facial basados en el aprendizaje profundo. Es en este punto, donde las redes neuronales convolucionales presentan un mejor rendimiento en los procesos de detección de rostros en parte gracias al aumento de bases de datos grandes que permitieron mayor desarrollo en este aspecto.

Dentro del problema de la detección facial existen dos acercamientos para llegar a una solución, utilizar métodos basados en regiones y otro en ventanas deslizantes. El primero, resuelve el problema generando propuestas de objetos y posteriormente creando una serie de regiones que pueden incluir rostros, estas son analizadas por un clasificador DCNN para determinar si son o no rostros y devolver sus coordenadas en la imagen. Sin embargo, este método presenta un elevado coste computacional a causa del generador de propuestas. Por otro lado, el método basado en ventanas deslizantes emplea un kernel que realiza un barrido de la imagen y determina mediante un clasificador si existe o no un rostro en la ventana a la vez que se redimensiona la imagen para aumentar la precisión. Este acercamiento presenta un menor coste computacional y mejor escalabilidad.

4.1.3 Extracción de características

El paso posterior a la detección de rostros en el reconocimiento facial, es la extracción de características de la cara. Este es uno de los pasos más importantes puesto que la eficiencia de los sistemas de reconocimiento facial depende en gran medida de la calidad de las características extraídas. Además, se identifican mediante puntos de referencia faciales y puntos fiduciaros. Tradicionalmente, se emplean métodos basados en modelos para obtener y localizar estos puntos. Aunque actualmente se están empleando métodos basados en regresión en cascada que mejoran la precisión frente al enfoque tradicional.

4.1.4 Verificación e identificación

En la última etapa del proceso de reconocimiento facial se suele dividir en dos categorías en función del problema que se afronte, el de identificación y el de verificación. Para la identificación facial, se trata de un método de “uno contra muchos” ya que el sistema busca determinar la identidad del individuo mediante una comparación con el resto de las imágenes en la base de datos. En el caso de la verificación, el problema es del tipo “uno contra uno” ya que se busca si dos imágenes corresponden al mismo individuo. Por esta razón, se realiza la comparación entre la imagen obtenida en la adquisición frente a la imagen de referencia del individuo.

4.1.5 Casos de uso y aplicaciones

En la actualidad, los sistemas de reconocimiento facial son ampliamente utilizados en los ámbitos de seguridad como es el caso de los sistemas de seguridad de los aeropuertos para facilitar el acceso por aduanas y detectar usuarios con pasaportes falsos. Este es uno de los muchos ejemplos que se pueden encontrar en este ámbito.

También encontramos ejemplos en dispositivos de uso diario como pueden ser los teléfonos móviles que presentan técnicas de reconocimiento facial para su desbloqueo o en funciones de enfoque automático a la hora de tomar fotos. Otro ejemplo, relacionado con el uso diario es la funcionalidad de algunas redes sociales que permiten el etiquetado automático de las personas que aparecen en la foto.

Finalmente, también podemos encontrar aplicaciones donde esta tecnología se emplea para agilizar una operación. Como en el caso de algunos establecimientos en China que permiten el pago tomando una foto del rostro del cliente.

4.2 SISTEMAS EMBEBIDOS

4.2.1 Introducción a los sistemas embebidos

Los sistemas embebidos son una combinación de hardware y software especializado para realizar una serie de tareas específicas dentro de un sistema más grande. Esta tecnología se caracteriza por sus limitados recursos, alta eficiencia y emplearse para tareas en tiempo real. Aunque es cierto que con el paso de los años ciertas desventajas como la limitación de recursos se han ido superando con el desarrollo de hardware cada vez más potente.

Los inicios de los sistemas empotrados datan de la década de 1960 y 1970, donde destaca en los ámbitos militar y en sistemas de control industrial. Sin embargo, estos dispositivos eran costosos y voluminosos, pero permitieron un gran avance en los campos de la automatización y control de proceso. Posteriormente en la década de 1980 comienza a aumentar el uso de esta tecnología debido a los avances en que permitieron reducir el coste y tamaño de los microcontroladores y microprocesadores. Finalmente, en el nuevo milenio con el surgimiento de la industria 4.0 y el internet de las cosas (IoT), los sistemas empotrados presentan un crecimiento y desarrollo considerable debido a que toman un papel fundamental en este nuevo paradigma.

4.2.2 Casos de uso y aplicaciones

En la actualidad es muy común encontrar sistemas empotrados en un gran número de ámbitos muy diversos. Entre ellos, en automatización para controlar algunas funciones de los vehículos tales como los frenos ABS o en los sistemas de asistencia al conductor. También se emplean en el ámbito de la aviónica para sistemas de navegación y control de vuelo. Otro ejemplo, sería en la medicina para los dispositivos de monitoreo de la salud del paciente o en los equipos de resonancias. Estos solo son unos pocos de los muchos ejemplos que se pueden encontrar sobre el uso de esta tecnología, que gracias a su versatilidad y utilidad es tan empleada.

5. METODOLOGIA EXPERIMENTAL

En este apartado de la memoria se abordará de manera detallada los requisitos del sistema, así como el hardware y herramientas software empleadas en el desarrollo. Por último, se describen las fases que se han presentado durante la elaboración del proyecto junto con las diferentes demostraciones que se han implementado siguiendo con la metodología en espiral.

5.1 REQUISITOS DEL SISTEMA

A continuación, se presentan los requisitos que deberá cumplir el sistema diseñado. Se ha realizado la distinción entre requisitos funcionales y no funcionales, siendo los primeros aquellos que definen el comportamiento del prototipo. Mientras que los no funcionales describirán características y atributos del sistema.

5.1.1 Requisitos funcionales

Los requisitos funcionales se encuentran agrupados en la siguiente Tabla 1 donde se muestra cada requisito junto con la descripción del funcionamiento esperado.

Tabla 1: Requisitos funcionales del sistema.

Requisito	Descripción
Detección facial	El sistema ha de ser capaz de capturar imágenes y detectar la cara del usuario.
Autenticación facial	El sistema ha de ser capaz de reconocer y diferenciar las caras de los usuarios.
Control de acceso	El sistema ha de ser capaz de permitir o denegar el acceso a la maquinaria en función de la disponibilidad de esta. Además de conceder el acceso únicamente a usuarios registrados y con los permisos suficientes.
Autenticación por PIN	El sistema ha de permitir el registro por pin en el caso en que el reconocimiento facial falle.
Entrenamiento automático	El sistema deberá ser capaz de lanzar el aprendizaje de las caras de manera automática una vez se cumplan con las condiciones establecidas.
Envío de datos	El nodo del sistema empotrado deberá poder enviar las imágenes de las caras de los usuarios al computador. Este último, deberá almacenarlas en carpetas en función del usuario.
Registro de accesos	El sistema deberá mantener un registro temporal de cuando se ha realizado el ultimo acceso de cada usuario.
Acceso a la base de datos	El sistema ha de ser capaz de acceder a la base de datos para actualizar y consultar datos del estado de sistema.
Mantenimiento de la DB	El sistema debe permitir que se pueda modificar la base de datos por el administrador para corregir posibles errores. Así

	como, permitir la modificación para añadir o eliminar usuarios y cambiar permisos.
Mantenimiento de las imágenes	El sistema debe permitir que el administrador pueda corregir errores en el almacenamiento de las imágenes.

5.1.2 Requisitos no funcionales

Los requisitos no funcionales se agrupan en la Tabla 2 donde se muestra cada requisito junto con la descripción de este.

Tabla 2: Requisitos no funcionales del sistema.

Requisito	Descripción
Tamaño	El encapsulado del sistema empotrado no deberá superar un tamaño de 170x130x60 mm.
Peso	El conjunto del sistema embebido no deberá ser mayor a 400 gramos.
Formato de imagen	El sistema deberá ser capaz de capturar imágenes en formato .png y una resolución no inferior a 8 megapíxeles.

5.2 DISEÑO GENERAL DEL SISTEMA

En la siguiente Figura 2, se presenta el diagrama general del sistema derivado de los requisitos del sistema. El sistema se compone de dos nodos, uno de front-end y otro de back-end. El primero se encarga de la interacción con el usuario, estas funciones corresponden a leer por teclado, adquirir imágenes de los operarios, dar indicaciones de las operaciones mediante leds y autenticar al usuario por PIN o por reconocimiento facial. Queda incluido en las funciones de este nodo el conceder o denegar el acceso a la maquinaria. Por otro lado, tenemos el back-en que cumple con la función de almacenar los datos de los usuarios y activos en una base MySQL y almacenar las imágenes de los operarios que va capturando el front-end. Además, se encarga de entrenar la red neuronal para el reconocimiento facial. El sistema también cuenta con la comunicación por sockets TCP de los nodos enviando los datos necesarios para el correcto funcionamiento de este.

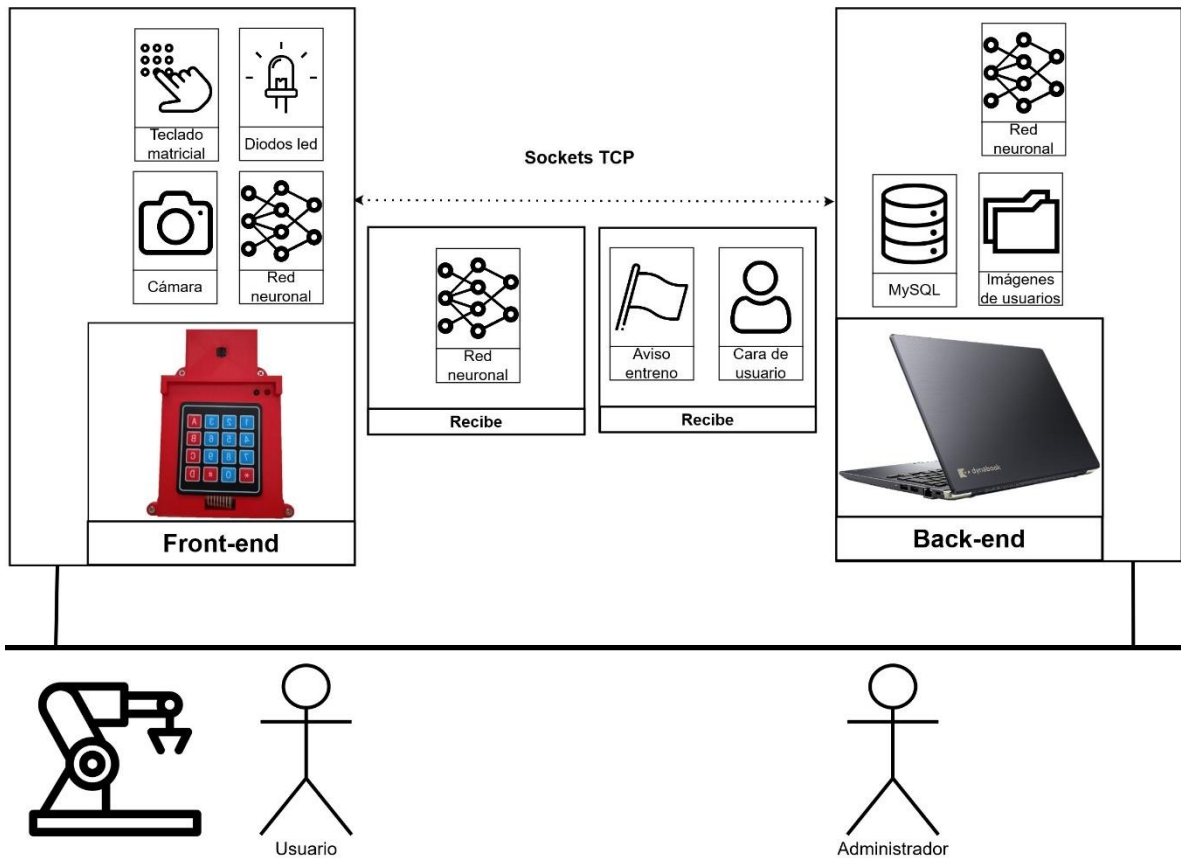


Figura 2: Diagrama del diseño del sistema. Elaborado mediante Draw.io.

5.3 EVENTUALIDADES

En el caso de producirse un error en la comunicación o alguna otra falla en el sistema. El nodo front-end mantendrá el led rojo encendido permanentemente, dejando abierto el acceso para no bloquear el acceso a la maquinaria y señalar que se ha producido un error en el sistema. Tal eventualidad queda reflejada en la Figura 3.

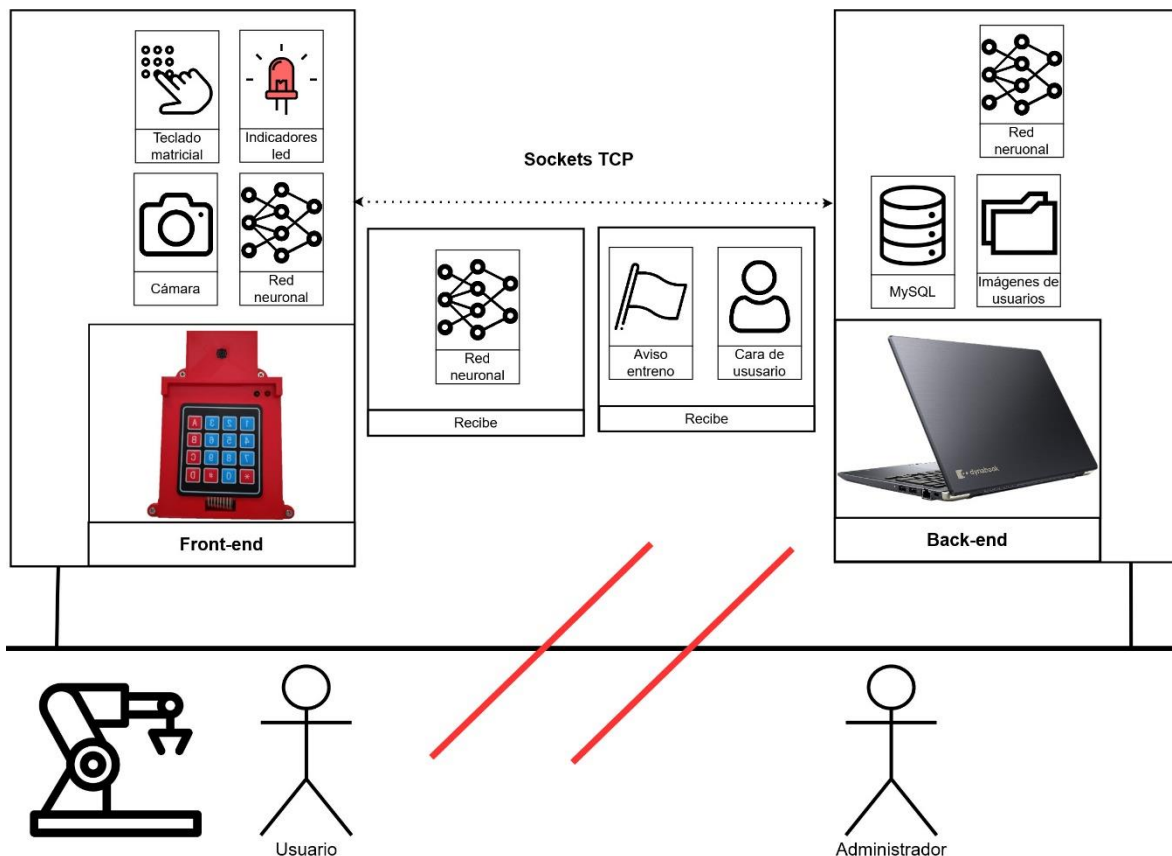


Figura 3: : Diagrama del diseño del sistema, frente a una eventualidad. Elaborado mediante Draw.io.

5.4 DESCRIPCIÓN DEL HARDWARE

En este apartado se abordarán de manera detallada la descripción y justificación de los componentes hardware que constituye al sistema embebido. Así como su función en la aplicación desarrollada. Para analizar los componentes se emplean tablas que puntúan del 1 al 3 cada una de las características consideradas relevantes a la hora de seleccionar el componente, el que mayor puntuación tenga será el elegido para el proyecto.

5.4.1 Microordenador

Este dispositivo se trata de un computador de tamaño reducido que utiliza un microprocesador como unidad central de proceso. Este dispositivo permitirá ejecutar la aplicación desarrollada e interactuar con los sensores y actuadores mediante el uso de sus pines general-purpose input/output (GPIO).

Para valorar la elección del microcomputador se ha tenido en cuenta tanto su coste, como capacidad de procesamiento, si tiene el suficiente número de GPIOs, el tipo de cámara que acepta. El resultado del análisis se muestra en la siguiente Tabla 3.

Tabla 3: Tabla comparativa del microordenador.

Alternativas	Coste	RAM	GPIOs	Compatibilidad con la cámara	Total
Beagle Bone Black	3	2	3	0	7
Raspberry Pi 4	1	3	1	3	8
Raspberry Pi 3	2	1	1	3	7

Se ha optado por una Raspberry Pi 4 Model B, de 8 GB ya que tendrá potencia suficiente como para poder realizar el procesamiento de imagen y utilizar el modelo de red neuronal. Además, pese a su limitado número de GPIOs serán suficientes para interactuar con los sensores y actuadores que presenta el prototipo.

5.4.2 Teclado matricial

Los teclados matriciales son dispositivos compuestos por una matriz de pulsadores organizados en filas y columnas. Por su distribución de las teclas se puede conocer el botón pulsado empleando un menor número de pines que de teclas. Además, cuentan con un tamaño reducido y un bajo coste. En el proyecto, se emplea para introducir el pin del usuario y seleccionar la maquina o servicio a la que el usuario desea acceder.

Dentro de la elección de teclado se tendrá en cuenta el coste, el tamaño y la ergonomía de este. Por otro lado, se seleccionará entre tipos de teclados distinguiendo los de membrana, tecla y botonera.



Figura 4: Teclados de tipo membrana, tecla y botonera, ordenado de izquierda a derecha. Imágenes extraídas del catálogo de Diatronic (2024) y Microlog (2024).

Tabla 4: Tabla comparativa de teclados matriciales.

Alternativas	Coste	Tamaño	Ergonomía	Total
Teclado tipo membrana	2	3	2	7
Teclado tipo tecla	1	1	3	5
Teclado tipo botonera	3	2	1	6

5.4.3 Cámara

Las cámaras son sensores capaces de obtener información mediante la captura de imágenes. Este dispositivo es crucial para la aplicación puesto que permite capturar las caras de los usuarios.

Tabla 5: Tabla comparativa de cámaras.

Alternativas	Coste	Tamaño	Puerto CSI	Calidad de imagen	Total
Raspberry Pi Camera Module 2 NoIR	2	3	3	1	9
Raspberry Pi Camera Module 2 Standard	3	3	3	2	11
Innodisk USB2 Camera Module	1	1	0	3	5

Se ha optado por emplear la cámara Raspberry Pi Camera Module v2 8MP debido a su compatibilidad, fácil configuración y conexionado mediante el puerto CSI de la Raspberry Pi obteniendo altas velocidades en la toma de imágenes. Bajo consumo energético y buena calidad en la adquisición de imágenes. Además, se ha seleccionado la versión estándar del módulo que presenta dos ventajas principales. En primer lugar, el precio y en segundo lugar que la versión sin filtro infrarrojo (NoIR) está diseñada para trabajar en entornos con poca iluminación.

5.4.4 Circuitería

En este apartado quedan incluidos los elementos como las resistencias utilizadas, concretamente se emplean de $1k\Omega$ para limitar las corrientes que transiten por los leds dentro de un rango seguro. También se han necesitado cables macho-hembra para las conexiones entre leds, teclado y la Raspberry Pi. Además de una placa perforada de prototipado para soldar tanto los leds como las resistencias.

Dentro de la circuitería destacan los 2 diodos led, uno verde y otro rojo para señalar al usuario durante la interacción con el prototipo.

5.5 DESCRIPCIÓN DE HERRAMIENTAS SOFTWARE

En este apartado se abordarán de manera detallada la descripción y justificación de las herramientas software utilizadas en el proyecto. Dentro de estas herramientas, presentaremos los lenguajes de programación, las librerías y la base de datos.

5.5.1 Lenguaje de programación y justificación

Respecto a los lenguajes de programación, se ha optado por emplear C++ para la parte correspondiente al conjunto de la Raspberry Pi, mientras que para la computadora se ha empleado Python.

5.5.1.1 C++

Se ha optado por la elección de C++ frente a otros lenguajes de programación debido a que es ampliamente utilizado en el desarrollo de sistemas embebidos. Además, este lenguaje se caracteriza por su rendimiento y eficiencia en el manejo de recursos, así como en su velocidad debido a su condición de lenguaje compilado. Por otro lado, otra de las razones por las que se decidió optar esta solución fue que C++ es similar a C del que ya se tenía experiencia previa.

5.5.1.2 Python

Respecto a Python, se ha optado por esta solución debido a que es un lenguaje de desarrollo sencillo de utilizar y aprender. Además, su uso está muy extendido por lo que abunda su documentación. Por otro lado, presenta un gran número de librerías que se podrán utilizar, destacando potentes librerías para el ámbito científico y en aprendizaje automático.

5.5.2 Base de datos MySQL

En el proyecto se empleó una base de datos para el almacenamiento y gestión de la información de la aplicación. Se optó por una base de datos en un entorno MySQL. Esta solución se escogió debido a su uso extendido. Además, está basado en lenguaje SQL, sencillo de utilizar.

5.5.3 Librerías

Para la realización del proyecto se emplearon librerías sobre las que apoyarse para realizar ciertas funciones de sistema como el aprendizaje del modelo y el procesamiento de imágenes. A continuación, se presenta en un formato de lista las librerías empleadas junto con una descripción de su papel en el proyecto y su función en este.

- Keras: Esta librería de código abierto programada en Python está orientada a la elaboración de redes neuronales. Destaca por ser fácil de utilizar y presenta mucha documentación debido a que es muy utilizada. Otro factor importante es su compatibilidad con la librería Frugally Deep que permite el uso de redes creadas y entrenadas en Keras en un lenguaje C++. En el proyecto se emplea para entrenar el modelo del clasificador que resuelva el problema de identificación del reconocimiento facial.
- Frugally Deep: Esta librería de código abierto permite cargar y utilizar modelos de redes neuronales preentrenadas en Keras en un lenguaje C++. Se caracteriza por su eficiencia en cuanto a memoria y velocidad, además de no requerir dependencias adicionales. Para utilizar la red entrenada en Keras se requiere que previamente se realice la conversión en un formato JSON del modelo. En el proyecto la librería se usa para poder emplear la red entrenada en Keras en la Raspberry Pi donde se trabaja en C++.
- Opencv: Esta librería de código abierto y multiplataforma orientada a la visión artificial y aprendizaje automático. Es ampliamente utilizada y por lo tanto existe mucha documentación. Además, es muy potente y cuenta con una gran cantidad de funcionalidades y algoritmos. En el proyecto se emplea para la adquisición, procesado de imágenes y para la detección de caras.
- Dlib: Esta librería de código abierto está orientada a la visión por computador y el aprendizaje automático. Se puede utilizar tanto en C++ como en Python y cuenta con funcionalidades para procesamiento de imágenes, optimización, aprendizaje profundo y para el reconocimiento facial. En el proyecto se ha empleado para obtener los vectores de características de los rostros de los usuarios.

- MySQL Connector: Esta librería está diseñada para poder acceder a una base de datos MySQL desde aplicaciones en Python y C++ en sus respectivas versiones. En el proyecto se emplea para poder consultar y escribir en la base de datos del sistema.

5.6 FASES DEL DESARROLLO

En este apartado se abordarán las diferentes fases que se han dado en el desarrollo del proyecto. Se describen tanto los demostradores implementados durante el desarrollo iterativo, como las pruebas unitarias realizadas sobre cada uno de los módulos del sistema.

5.6.1 Desarrollo iterativo

Para el desarrollo del sistema empotrado, se elaboraron una serie de demostradores que iban expandiendo la funcionalidad de la anterior hasta alcanzar una versión completa del sistema final. Además, con cada iteración se realizaron pruebas para verificar que el funcionamiento del demostrador era adecuado y por lo tanto se podía pasar a la siguiente fase del desarrollo.

Se realizaron tres demostradores en total, la primera sirvo para realizar un acercamiento inicial al problema de identificación y al flujo de la aplicación. En segundo lugar, el siguiente demostrador tiene por objetivo implementar todas las funcionalidades del primer demostrador entre los dos nodos que componen el sistema y añadir la comunicación entre ambos. Finalmente, el tercer demostrador busca cumplir con todas las funcionalidades del sistema sirviendo como una primera versión del sistema final.

5.6.1.1 Demostrador del motor del sistema

Este demostrador se desarrolló en Python y se ejecutó en el computador, por lo tanto, se sustituyeron las funcionalidades relacionadas con los GPIOs por alternativas similares. El objetivo en el demostrador era simular el funcionamiento básico o motor del sistema tal como el que se muestra en la Figura 5.

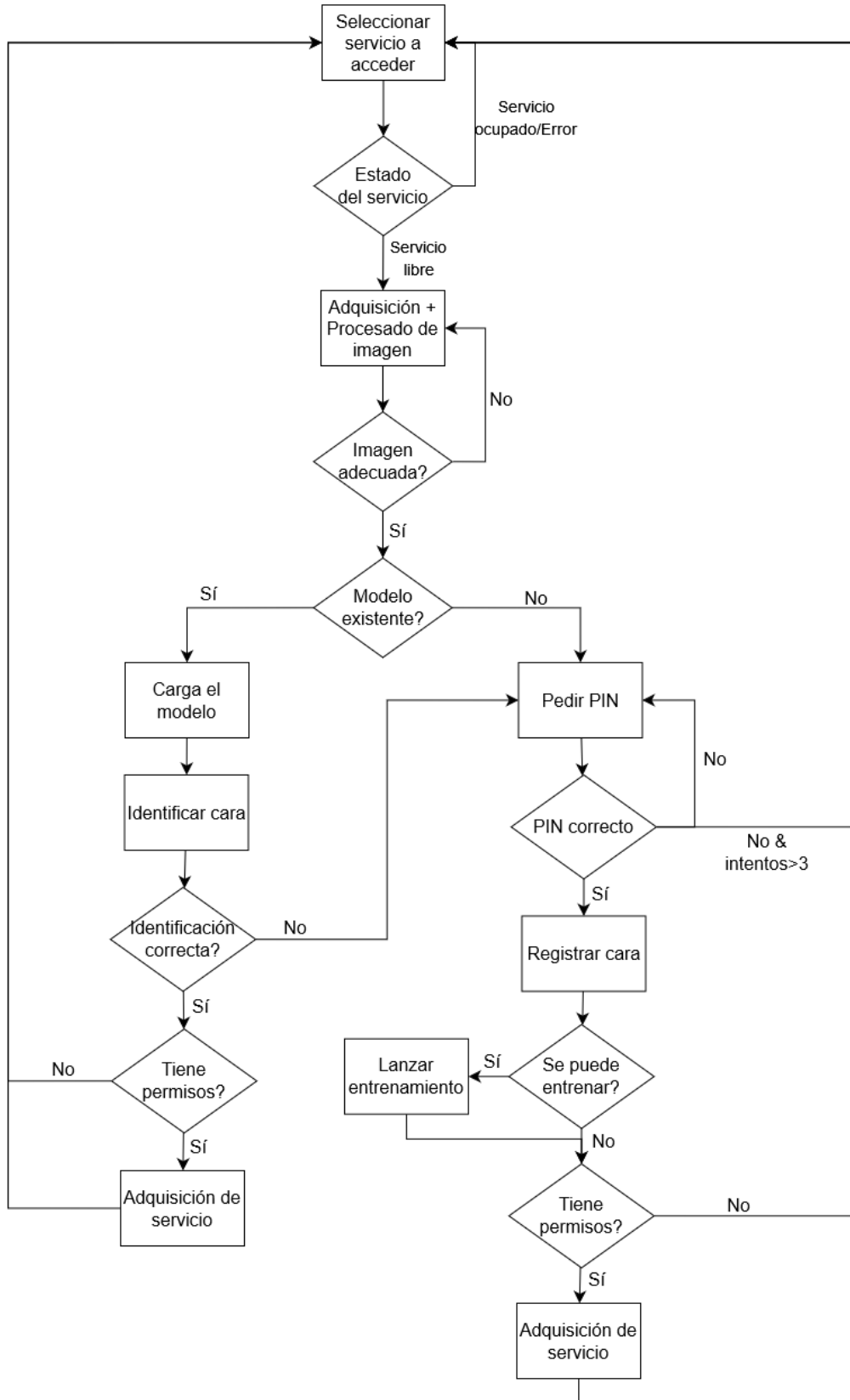


Figura 5: Diagrama de flujo del motor del sistema. Elaboración propia empleando Drwa.io.

Por lo que respecta a las diferentes funcionalidades que presenta el demostrador se muestra la Tabla 6 indicando cuales fueron implementadas, sustituidas y no implementadas. Cabe destacar, que, aunque algunas se sustituyeron el fin último del demostrador era simular el flujo del programa enfatizando la generación del modelo y el paso del registro mediante pin al registro por reconocimiento facial.

Tabla 6: Estado de las funcionalidades en el demostrador 1.

Funcionalidad	Descripción	Implementación
Registro de accesos	Se actualiza la base de datos cada vez que se accede a un servicio o se registra la cara de un usuario no identificado.	Implementada
Control de accesos	Se accede a la base de datos para consultar el estado de la maquina y los permisos del usuario solicitante para aceptar o denegar el acceso.	Implementada
Almacenamiento de caras	Se organizan las caras de los usuarios no identificado en sus respectivas carpetas una vez se registra usando el pin.	Implementada
Aprednizaje de la red	Se realiza el entrenamiento de la red de manera automatica y resulta en un modelo capaz de reconocer la cara de los usuarios.	Implementada
Aprendizaje en el reconocimiento fallido	Se vuelve a lanza el entrenamiento una vez que se ha producido un fallo en el reconocimiento generando un modelo más preciso.	Implementada
Adquisicion de imagen	Se sustituye por una carpeta con un conjunto de imágenes de usuarios. En el demostrador la imagen se selecciona de manera aleatoria y se elimina de la carpeta una vez registrada.	Simulada
Procesamiento de imagen	Extracción de la cara y las caracterisitcas de la misma a partir de la imagen capturada.	Implementada
Ingreso del pin por teclado matricial	Se sustituye utilizando el teclado del computador. No se trabaja con el GPIOs ni con el teclado matricial 4x4.	Simulada
Visualización de los leds	Se sustituye por mensajes que se muestran por pantalla.	Simulada
Comunicación	No se implementa la comunicación entre los nodos del sistema. El demostrador se ejecuta unicamente en el nodo computador.	No impementada

Para dar por válida el demostrador se analizó que se cumplían los siguientes requisitos para poder pasar a la siguiente fase del desarrollo:

- Las operaciones descritas en la tabla de funcionalidades del demostrador se cumplen sin presentar errores.
- El flujo del programa no presenta errores o saltos inesperados entre ramas. Forzando todos los casos posibles del sistema.
- No se presentan errores durante la primera etapa, donde no se ha generado el modelo ni hay suficientes datos almacenados. En esta etapa es necesario utilizar el registro por pin.
- No se presentan errores durante la segunda etapa donde ya se ha generado un modelo de al menos que reconozca al menos dos usuarios. De forma que por pin se registren los usuarios que el modelo no conoce o los que conoce, pero falle en el reconocimiento.
- No se presentan errores durante la última etapa donde ya no es necesario el ingreso por pin y todos los accesos se realizan mediante reconocimiento facial.

5.6.1.2 Demostrador del sistema embebido

En esta segunda demostradora se ha dividido la aplicación entre los dos nodos que componen el sistema, el front-end que corresponde a la Raspberry Pi y el back-end que corresponde al computador. Debido a esta segmentación, se han implementado las comunicaciones entre ambos nodos, funcionalidad que no presenta el primer demostrador.

El objetivo en esta etapa es comprobar que la segmentación del programa entre los nodos no da lugar a errores que alteren el flujo básico del sistema descrito en el demostrador previa. A su vez, también se debe comprobar que la comunicación entre el front-end y el back-end se realiza correctamente.

Atendiendo a las funciones de cada nodo, se destaca que en el back-end se realiza el entrenamiento del modelo, tal como se implementó en el demostrador anterior. Por otro lado, se añade tanto el envío del modelo como la recepción de imágenes mediante sockets TCP. En cambio, en el caso del front-end se realizan el resto de las operaciones del primer demostrador, con la diferencia de que el lenguaje de programación pasa de Python a C++. Al igual que en el back-end, se han añadido las comunicaciones TCP, en este caso para el envío de imágenes y recepción del modelo.

En la siguiente Tabla 7 se muestra las funcionalidades implementadas o simuladas en el demostrador junto con la indicación del nodo donde se ejecuta esta.

Tabla 7: Estado de las funcionalidades del demostrador 2.

Funcionalidad	Descripción	Nodo	Implementación
Registro de accesos	Se actualiza la base de datos cada vez que se accede a un servicio o se registra la cara de un usuario no identificado.	Front-End	Implementada
Control de accesos	Se accede a la base de datos para consultar el estado de la maquina y los permisos del usuario solicitante para aceptar o denegar el acceso.	Front-End	Implementada
Almacenamiento de caras	Se organizan las caras de los usuarios no identificado en sus respectivas carpetas una vez se registra usando el pin.	Back-End	Implementada
Aprednizaje de la red	Se realiza el entrenamiento de la red de manera automatica y resulta en un modelo capaz de reconocer la cara de los usuarios.	Back-End	Implementada
Aprendizaje en el reconocimiento fallido	Se vuelve a lanza el entrenamiento una vez que se ha producido un fallo en el reconocimiento generando un modelo más preciso.	Back-End	Implementada
Adquisicion de imagen	Se sustituye por una carpeta con un conjunto de imágenes de usuarios. En el demostrador la imagen se selecciona de manera aleatoria y se elimina de la carpeta una vez registrada.	Front-End	Simulada
Procesamiento de imagen	Extracción de la cara y las caracterisitcas de la misma a partir de la imagen capturada.	Front-End	Implementada
Ingreso del pin por teclado matricial	Se sustituye utilizando el teclado del computador. No se trabaja con el GPIOs ni con el teclado matricial 4x4.	Front-End	Simulada
Visualización de los leds	Se sustituye por mensajes que se muestran por pantalla.	Front-End	Simulada
Comunicación	Se realiza la comunicación por sockets TCP para enviar imagnes de las caras y el modelo de red.	Ambas	Impementada

Para validar el demostrador se cumplían los siguientes requisitos:

- El programa ejecuta su funcionamiento básico sin presentar errores ni comportamientos que difieran respecto al flujograma planteado en el demostrador anterior.
- La comunicación entre ambos nodos se realiza de manera correcta sin presentar errores.

5.6.1.3 Demostrador del sistema completo

Este demostrador es la extensión de la anterior, a la que se le han añadido las interacciones con el entorno mediante los GPIOs y la cámara. Por lo tanto, en esta iteración se integra la adquisición de imágenes, la lectura de datos de credenciales por teclado matricial 4x4 y la señalización mediante los diodos leds.

El objetivo en esta etapa era conseguir una primera versión del funcionamiento del sistema final. Ya que, en este demostrador, se cubren todas las funcionalidades del sistema en cada uno de los nodos correspondiente.

Como se ha indicado previamente las funcionalidades que cumple el presente demostrador completan toda la tabla de funcionalidades descritas en las etapas anteriores, estas quedan recogidas en la Tabla 8.

Tabla 8: Estado de las funcionalidades en el demostrador 3.

Funcionalidad	Descripción	Nodo	Implementación
Registro de accesos	Se actualiza la base de datos cada vez que se accede a un servicio o se registra la cara de un usuario no identificado.	Front-End	Implementada
Control de accesos	Se accede a la base de datos para consultar el estado de la maquina y los permisos del usuario solicitante para aceptar o denegar el acceso.	Front-End	Implementada
Almacenamiento de caras	Se organizan las caras de los usuarios no identificado en sus respectivas carpetas una vez se registra usando el pin.	Back-End	Implementada
Aprendizaje de la red	Se realiza el entrenamiento de la red de manera automática y resulta en un modelo capaz de reconocer la cara de los usuarios.	Back-End	Implementada
Aprendizaje en el	Se vuelve a lanzar el entrenamiento una vez que se	Back-End	Implementada

reconocimiento fallido	ha producido un fallo en el reconocimiento generando un modelo más preciso.		
Adquisición de imagen	Se capturan las imágenes mediante la cámara conectada al puerto CSI de la Raspberry Pi.	Front-End	Implementada
Procesamiento de imagen	Extracción de la cara y las características de la misma a partir de la imagen capturada.	Front-End	Implementada
Ingreso del pin por teclado matricial	Se utilizando el teclado matricial 4x4 conectado a los GPIOs de la Raspberry Pi.	Front-End	Implementada
Visualización de los leds	Se generan patrones iluminando los diodos led conectados a los GPIOs de la Raspberry Pi con el objetivo de dar indicaciones al usuario.	Front-End	Implementada
Comunicación	Se realiza la comunicación por sockets TCP para enviar imágenes de las caras y el modelo de red.	Ambas	Implementada

Para dar por válida esta primera versión del sistema final, se han comprobado que el programa se ejecuta sin errores ni comportamientos no planificados en el flujograma. Además, se ha puesto especial atención en los aspectos relacionados con las nuevas funciones integradas, concretamente en la adquisición de imágenes y uso de GPIOs.

5.6.2 Pruebas unitarias

De manera concurrente a la programación de los demostradores se desarrollaron una serie de módulos para cubrir con las diferentes partes del sistema. Previamente a la integración de cada módulo en los demostradores se verificó su funcionamiento para poder prevenir algunos errores en los demostradores.

5.6.2.1 Desarrollo del módulo clasificador.

Para la generación del clasificador, se desarrollaron diferentes programas para probar diversas soluciones al problema de clasificación. Se realizaron pruebas con support vector machines, redes convolucionales y finalmente se optó por emplear un perceptrón multicapa. El aprendizaje se programó en Python mediante la librería de Keras debido a su compatibilidad con la librería Frugally Deep de C++.

En la puesta a prueba del módulo, se empleó un 'dataset' con carpetas de validación y entrenamiento, con imágenes de 4 usuarios. El programa procesaba

las imágenes, detectando las caras mediante funcionalidades de la librería Opencv y se extraían vectores de 128 características de estas mediante la librería Dlib. Finalmente se lanzaba el aprendizaje obteniendo un modelo mediante las imágenes del conjunto de entrenamiento y se comprobó el desempeño de la red mediante el conjunto de validación.

5.6.2.2 Desarrollo del módulo de la base de datos.

Este módulo tiene por objetivo realizar las operaciones de lectura y escritura de la base de datos. Se comprobó su funcionamiento mediante la creación de una tabla en MySQL y se realizaron pruebas lanzando operaciones de lectura y escritura de parámetro sobre esta. Este módulo se programó tanto para Python como para C++ mediante las librerías de MySQL Connector y mysqlcppconn para sus respectivos lenguajes de programación. Puesto que, tanto el nodo del front-end como el back-end deben acceder a la base de datos.

Posteriormente se adaptaron las bases de datos y los módulos para las necesidades concretas del sistema y las estructuras de las tablas que se muestran en las siguientes Tabla 9 y 10 ejemplifican las empleadas en el proyecto.

Tabla 9: Ejemplo de tabla users para tres usuarios registrados.

ID (INT)	Nombre (STRING)	Carpeta (STRING)	N_im (INT)	PIN (STRING)	Ingreso (TIMESTAMP)
0	Juan	/home/usuario/Projecto/Caras/Train/Id_0	5	1011	2024-06-12 11:05:10
1	Carlos	/home/usuario/Projecto/Caras/Train/Id_1	1	0000	2024-06-12 11:12:24
2	Eduardo	/home/usuario/Projecto/Caras/Train/Id_2	3	1010	2024-06-12 11:08:46

Tabla 10: Ejemplo de la tabla maquinaria para tres máquinas registradas.

ID_Maquina (STRING)	Max_Tokens (INT)	Curr_Tokens (INT)	Permiso (STRING)	Ingreso (TIMESTAMP)
M1	3	3	111	2024-06-12 11:05:10
M2	5	2	001	2024-06-12 11:08:46
M3	1	0	101	2024-06-11 19:55:28

5.6.2.3 Desarrollo del módulo de comunicación por sockets Python y C++.

Este módulo cumple con la función de gestionar la interacción entre el front-end y el back-end. Esta comunicación se realiza mediante 4 sockets TCP:

1. Socket TCP0: Este gestiona la primera comunicación entre ambos nodos y abre puertos en el back-end y se los indica al front-end. De forma que actúa como un servidor multihilo. Se optó por esta conexión para un posible escalado futuro donde actúen más de un nodo de reconocimiento facial. En la Figura 6 se muestra cómo se realiza la comunicación entre los nodos.

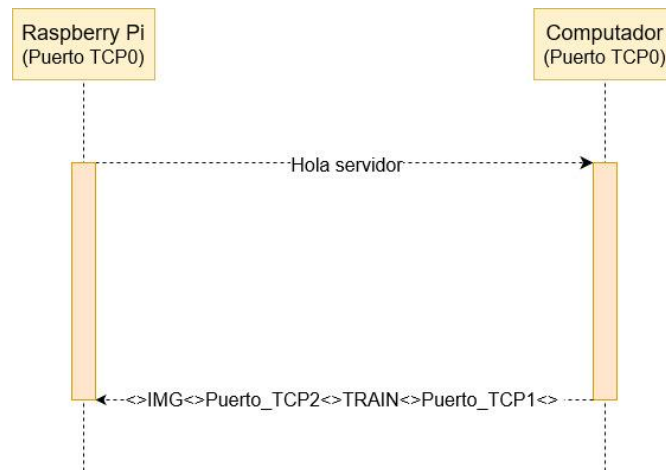


Figura 6: Diagrama de secuencia de la comunicación en el socket TCP0. Elaboración propia mediante Drwa.io.

2. Socket TCP1: En esta comunicación el front-end avisa al back-end de que analice si se cumplen las condiciones para lanzar el entrenamiento. Esto puede dar lugar o no a que se produzca el entrenamiento de la red. En la Figura 7 se muestra cómo se realiza la comunicación entre los nodos.

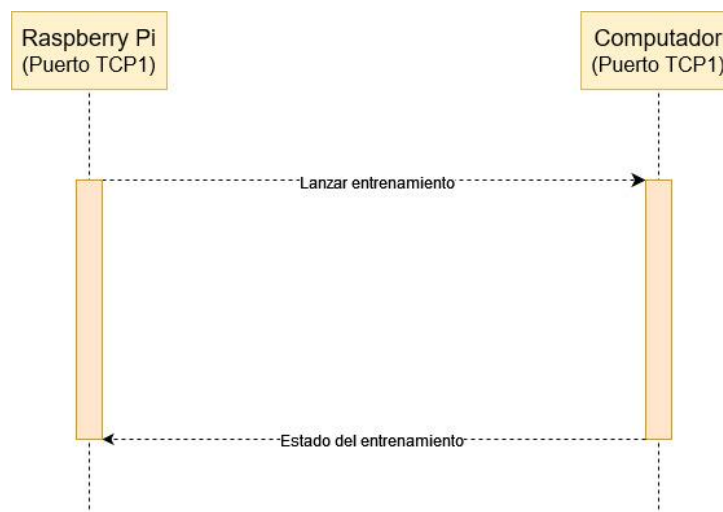


Figura 7: Diagrama de secuencia de la comunicación en el socket TCP1. Elaboración propia mediante Drwa.io.

3. Socket TCP2: Este gestiona el envío de imágenes desde la Raspberry Pi al computador indicando a que usuario corresponde la imagen. En la Figura 8 se muestra cómo se realiza la comunicación entre los nodos

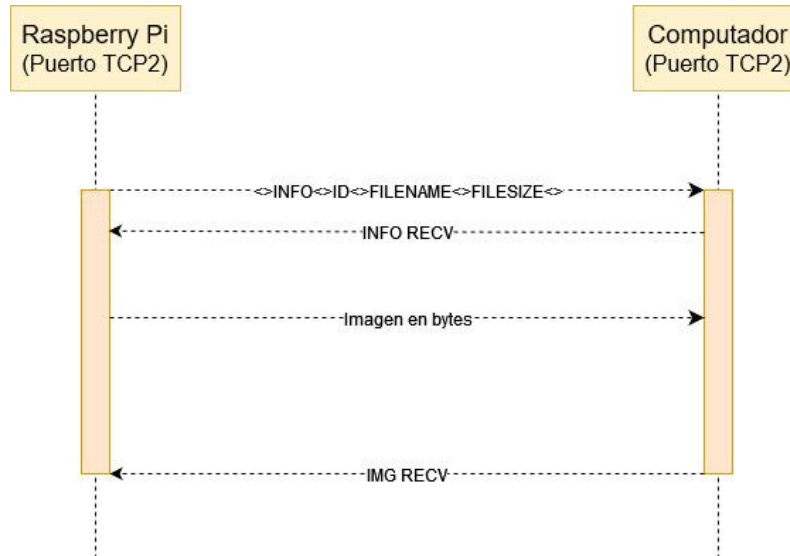


Figura 8: Diagrama de secuencia de la comunicación en el socket TCP2. Elaboración propia mediante Drwa.io.

4. Socket TCP3: En esta comunicación el back-end envía al front-end el modelo de red neuronal convertida en formato JSON. En la Figura 9 se muestra cómo se realiza la comunicación entre los nodos

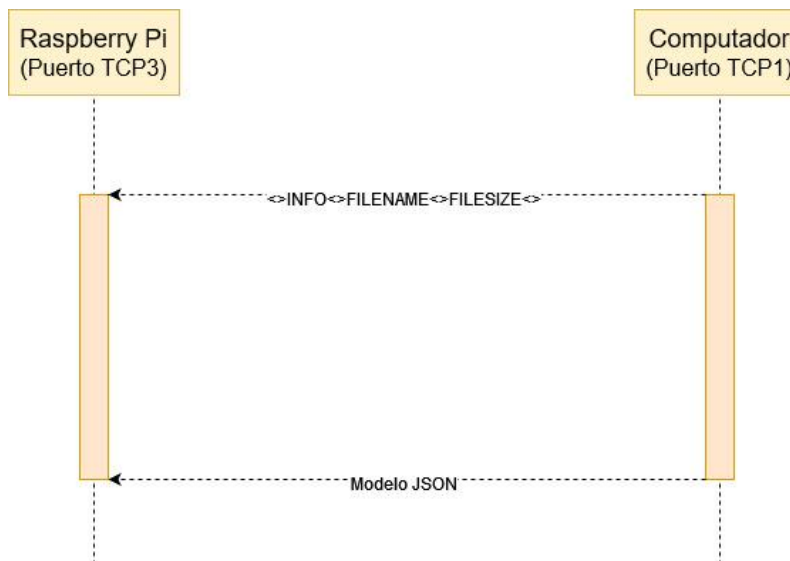


Figura 9: Diagrama de secuencia de la comunicación en el socket TCP3. Elaboración propia mediante Drwa.io.

Se realizaron pruebas de todas las comunicaciones previamente a su inclusión en los demostradores.

5.6.2.4 Módulo de carga del modelo y reconocimiento en el front-end.

Este módulo programado en C++, cumple con el objetivo de cargar el modelo desarrollado en Keras y recibido por el socket TCP3 en formato JSON en la Raspberry Pi. Para ponerlo a prueba se cargó el modelo mediante la librería Frugally-Deep y posteriormente se procedió a reconocer diferentes caras obtenido tanto la clase perteneciente como la probabilidad de que dicha cara corresponda a la clase.

5.6.2.5 Módulo de detección de caras.

Este módulo tiene como finalidad detectar la cara del individuo y extraer una imagen reducida de un rectángulo alrededor de la cara detectada. Se puso a prueba pasando diferentes imágenes de personas y filtrando la cara con mayor área puesto que es la que se considera más cercana a la cámara. Posteriormente a la detección se mostró por pantalla la región de interés (ROI) de la imagen para corroborar que corresponde con la cara del usuario validando el funcionamiento de la detección. En la siguiente Figura 10 se observa el proceso de detección.

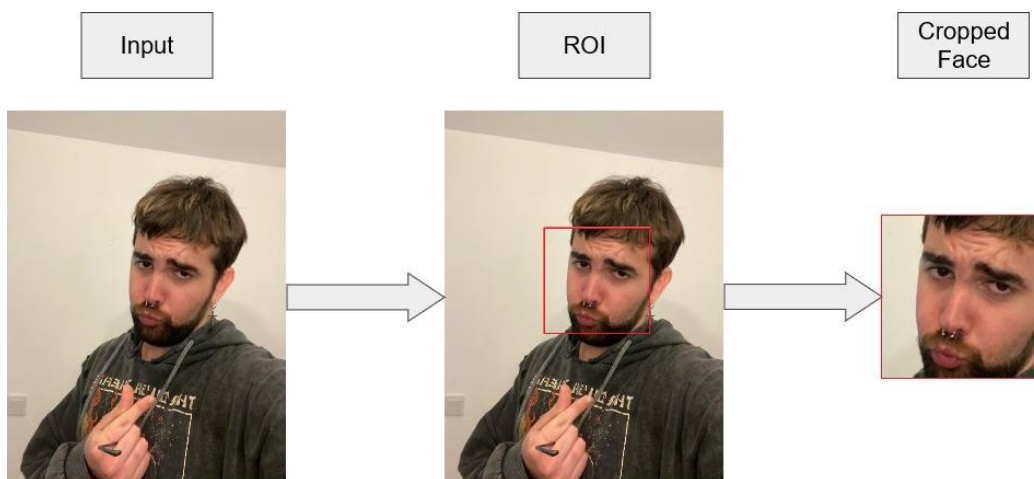


Figura 10: Etapas del proceso de detección de caras. Elaboración propia mediante Draw.io.

5.6.2.6 Módulo de extracción de características de las caras.

Este módulo emplea la librería Dlib de C++ y Python para extraer un vector de 128 características que definen una cara. La librería realiza la extracción de características de la siguiente manera, en primer lugar, detecta los puntos clave de la cara mediante un predictor, posteriormente se alinea el rostro para eliminar variaciones en las características debido a la orientación, finalmente se extrae las

características mediante una red neuronal ResNet que nos proporciona la propia librería. El módulo se puso a prueba extrayendo los vectores de varias caras y comparando la similitud de los vectores entre caras del mismo usuario. En la Figura 11 se pueden observar las etapas del proceso de extracción de características.

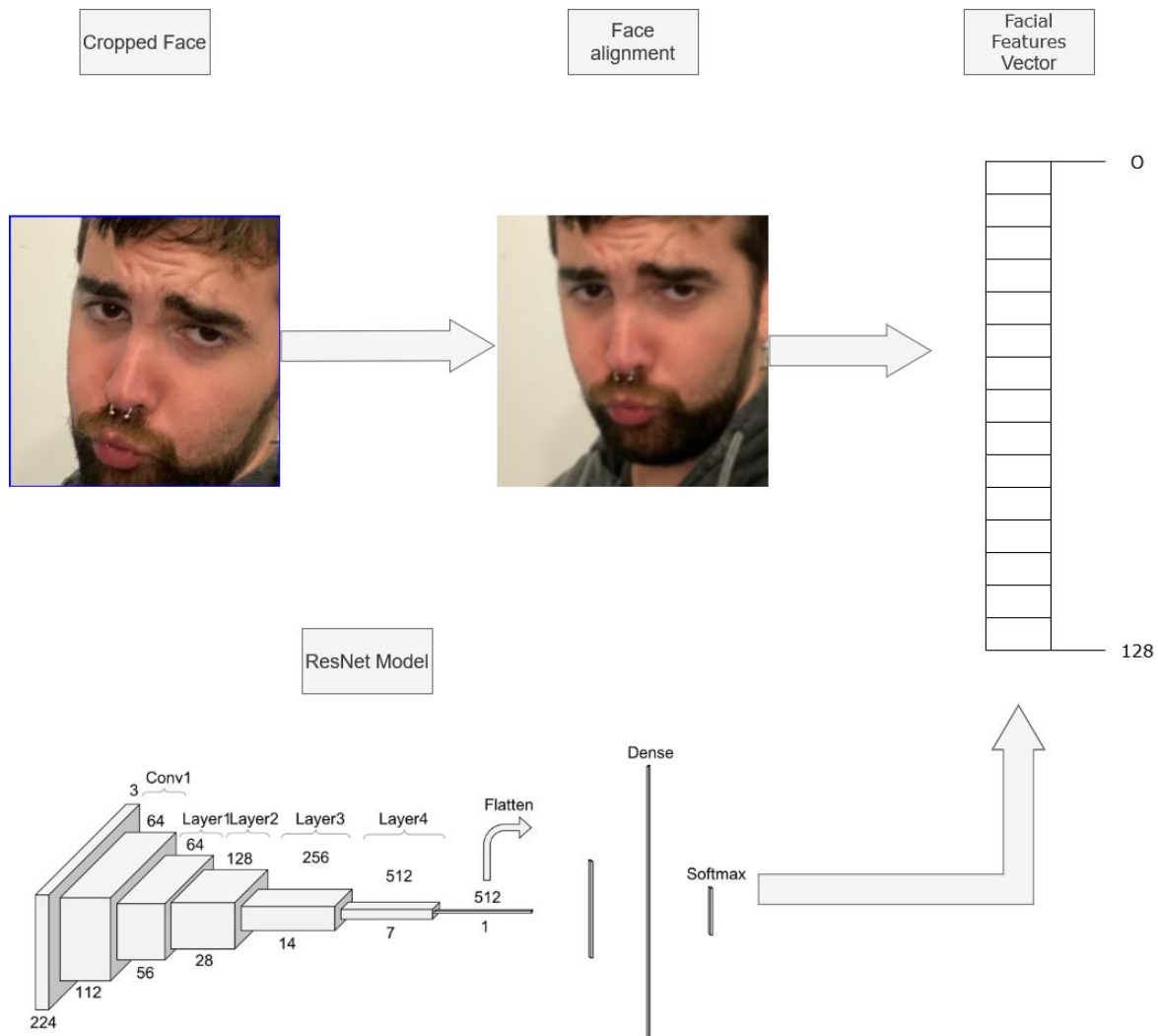


Figura 11: Proceso de la extracción de características. Elaboración propia en Drwaw.io empleando la imagen de la arquitectura ResNet de Towards Data Science (Yosinski, 2020).

5.6.2.7 Módulo de adquisición de imágenes.

Este módulo cumple con la función de capturar imágenes mediante a la cámara conectada al puerto CSI de la Raspberry Pi y se procesa la imagen mediante el módulo de detección de caras. Se puso a prueba tomado fotos y comprobando que la inclusión de la detección no produce errores.

5.6.2.8 Módulo de la clase teclado.

Este módulo se encarga del control y lectura del teclado matricial 4x4, así como los procesos de adquisición del pin o identificador de la maquinaria. Se realizaron diversas pruebas para comprobar que la lectura por el teclado se realiza de manera correcta leyendo las teclas e imprimiendo por pantalla la tecla pulsada. Además, se comprobó que no se pudiese introducir pines que superasen más de los dígitos máximos admitidos o menores a 0 caracteres. A si mismo, se hizo la misma comprobación para el caso de los identificadores.

5.6.2.9 Módulo de la clase led.

Este módulo cumple con la función de controlar los diodos leds permitiendo configurar el tiempo de encendido y apagado de los mismo.

6. RESULTADOS

En este apartado de la memoria se abordará de manera detallada el diseño y funcionamiento del sistema. Atendiendo a como se estructura el hardware y el software del prototipo, así como los manuales de uso para manejar el sistema dese un rol usuario y otro rol administrado.

6.1 ARQUITECTURA HARDWARE

A continuación, se presenta el diagrama de conexiones hardware del front-end y los planos del encapsulado del mismo.

6.1.1 Diagrama de conexiones

En este apartado se presenta el diagrama de conexiones Figura 12 correspondientes al sistema embebido de forma que quede bien definido como esta conectados los sensores y actuadores a la Raspberry Pi. Además, se han agrupado en la Tabla 11 la relación entre pines.

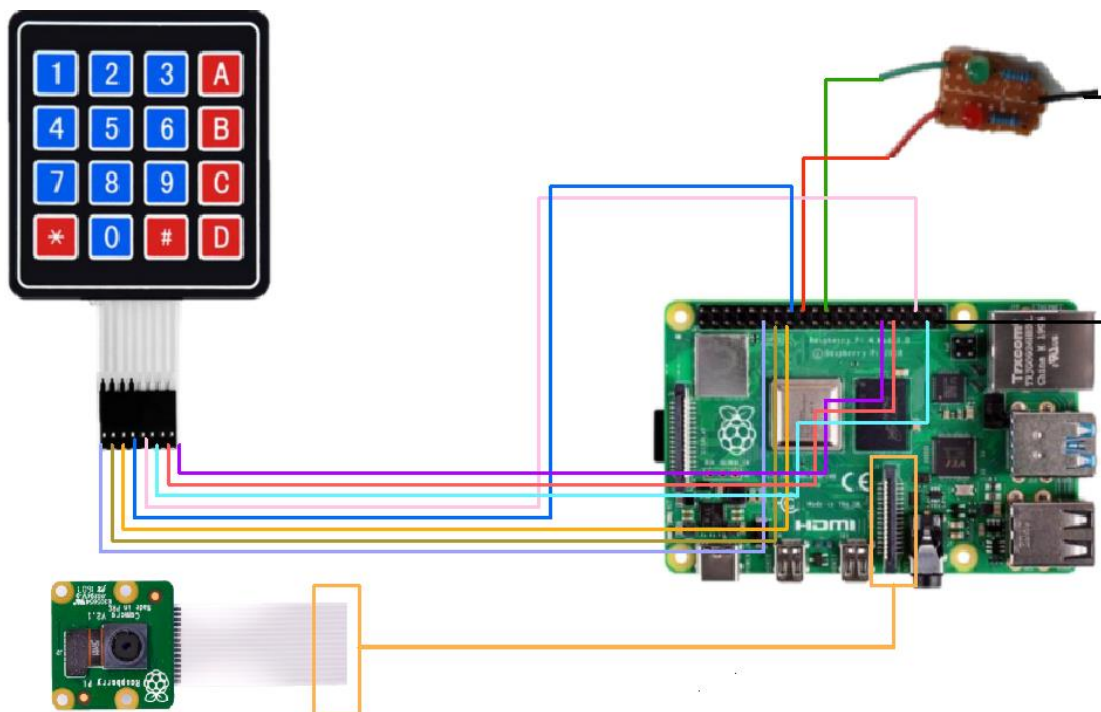


Figura 12: Esquema de conexiones en el módulo front-end. Elaboración propia empleado KiCAD.

Tabla 11: Relación de conexiones en el nodo front-end.

Componente		Raspberry		Configuración
Teclado	Columna 1	GPIO 17	PIN 11	Output
Teclado	Columna 2	GPIO 27	PIN 13	Output
Teclado	Columna 3	GPIO 22	PIN 15	Output
Teclado	Columna 4	GPIO 23	PIN 16	Output
Teclado	Fila 1	GPIO 16	PIN 36	Input
Teclado	Fila 2	GPIO 26	PIN 37	Input
Teclado	Fila 3	GPIO 6	PIN 31	Input
Teclado	Fila 4	GPIO 5	PIN 29	Input
Modulo Led	Led Rojo	GPIO 24	PIN 18	Output
Modulo Led	Led Verde	GPIO 25	PIN 22	Output
Modulo Led	Común	Tierra	PIN 39	-
Cámara	Puerto CSI	Puerto CSI		-

6.1.2 Montaje

1. Paso: Atornillar la cámara a la tapa superior

Se deberá atornillar la cámara RaspiCam Module 2 a la tapa superior por la cara posterior de forma que resulte en un conjunto como el de la Figura 13. Se deben utilizar tornillos de métrica M2.5 de 6 mm de longitud.

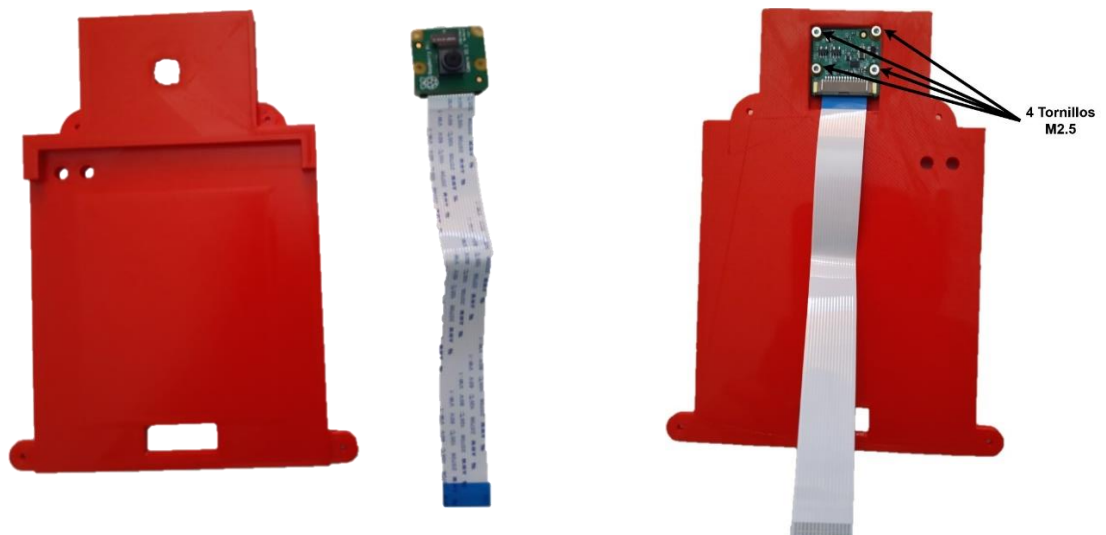


Figura 13: Resultado del paso 1: a la izquierda componentes implicados y a la derecha el resultado de la operación marcando la ubicación y métrica de la tornillería. Elaboración propia empleando Paint.

2. Paso: Inserción de los leds a la tapa superior

Se deberá adherir el módulo de leds a la tapa en los orificios ubicados en la misma. El adhesivo deberá asegurar una unión robusta y duradera. Además, el compuesto no deberá dañar ni la tapa ni la circuitería. El resultado de esta operación debe dar un resultado igual al de la Figura 14.

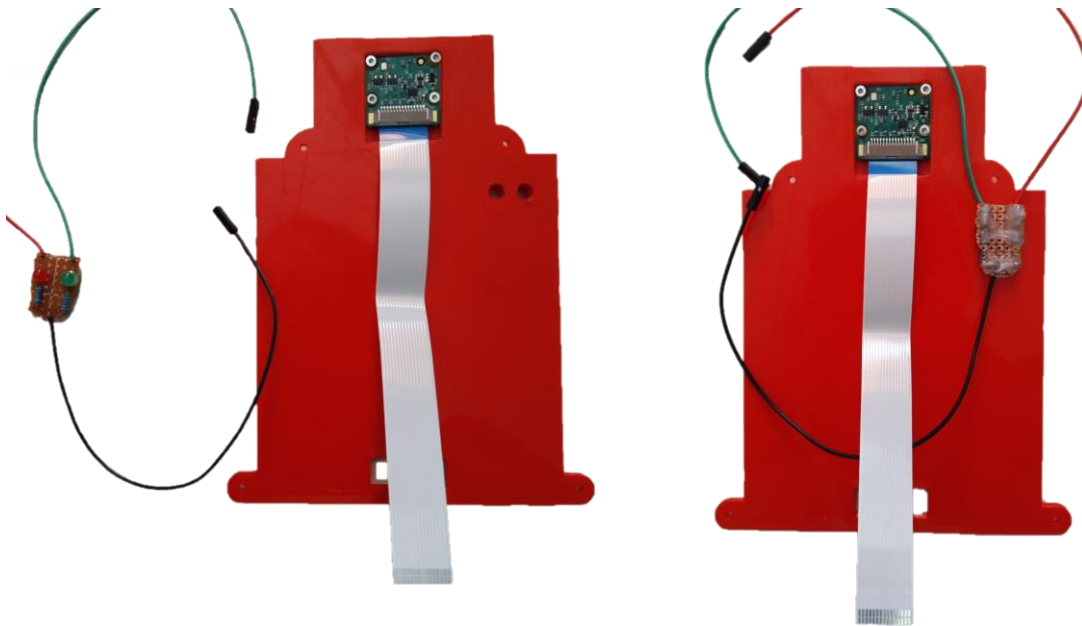


Figura 14: Resultado del paso 2; a la izquierda componentes implicado y a la derecha resultado de la operación. Elaboración propia utilizando Paint.

3. Paso: Adherir el teclado matricial 4x4 a la tapa superior

Se deberá retirar la película protectora del teclado matricial y se deberá adherir a la tapa en el espacio reservado para el teclado. Posteriormente se deberá pasar el bus del teclado por el orificio inferior de la tapa de manera que un resultado como en la Figura 15.



Figura 15: Resultado del paso 3: A la izquierda los componentes implicados en la operación y a la derecha el resultado de esta. Elaboración propia en Paint.

4. Paso: Atornillar la Raspberry Pi a la tapa inferior

Se deberá atornillar la Raspberry Pi 4 a la tapa inferior, se deberán utilizar tornillos de M2.5 y longitud 6 mm en los cuatro agujeros ubicados en el interior de la tapa, tal y como se muestra en la Figura 16.

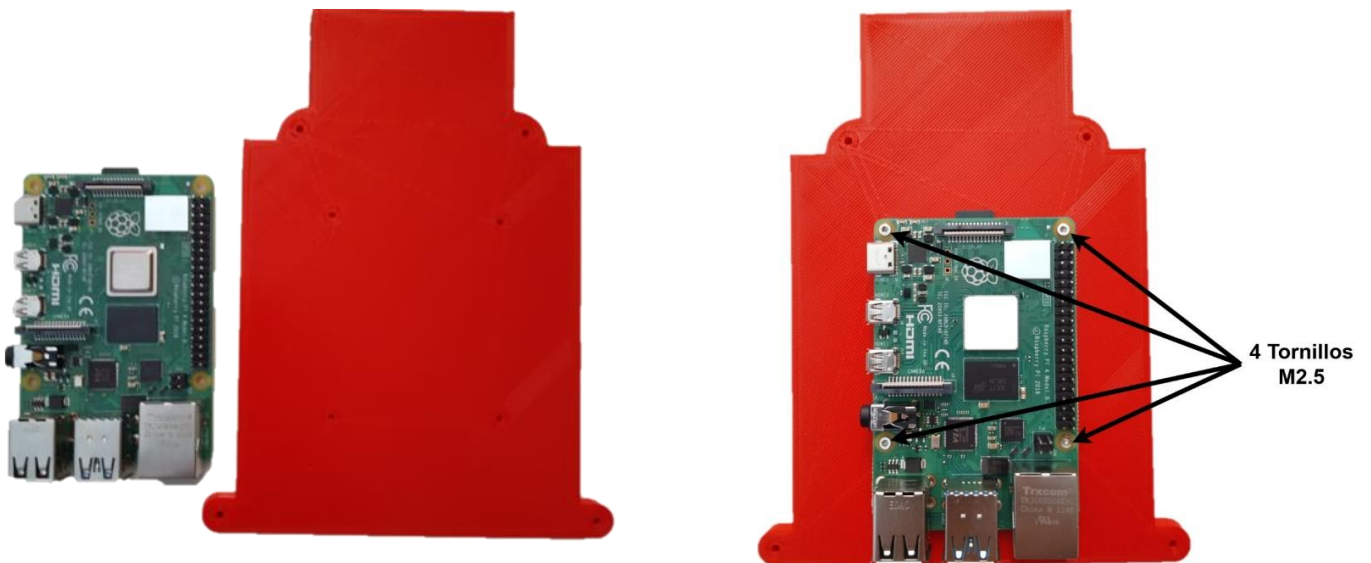


Figura 16: Resultado del paso 4: a la izquierda los componentes implicados y a la derecha el resultado de la operación junto con la ubicación de la tornillería y métrica empleada. Elaboración propia usando Paint.

5. Paso: Atornillar el lateral a la tapa inferior

A continuación, se deberá atornillar con tornillos de M3 y 8 mm de longitud a la pieza lateral a la tapa inferior con la Raspberry Pi atornillada. Primero se colocará el lateral sobre la tapa conforme muestra la Figura 18, posteriormente se deberá atornilla por los agujeros ubicados en los laterales de la tapa por la cara opuesta a donde está la Raspberry Pi.

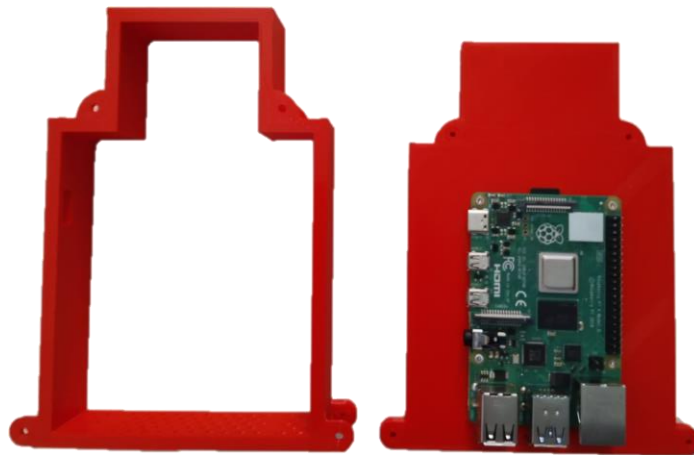


Figura 17: Elementos implicados previos al paso 5. Elaboración propia en Paint.

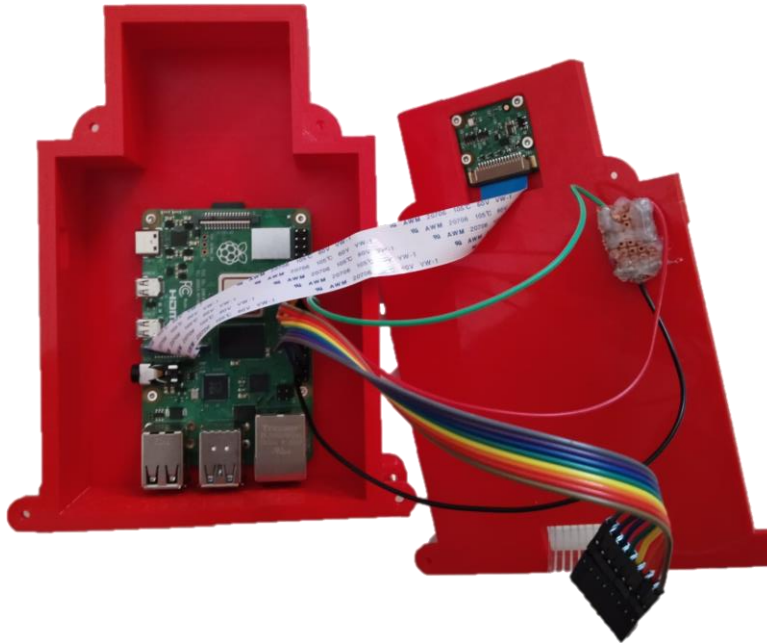


Figura 18: Resultado de la operación: a la izquierda la cara delantera y a la derecha la cara trasera con la ubicación y métrica de la tornillería empleada. Elaboración propia en Paint.

6. Paso: Conectar la electrónica a la Raspberry Pi

Se deberán realizar las conexiones entre el módulo de leds y el teclado matricial a la Raspberry Pi conforme al esquema de conexiones de la Figura 12. Además, se

deberá conectar la cámara y la Raspberry Pi mediante el bus CSI de ambos dispositivos, el cable debe quedar con el extremo azul en dirección a soporte negro del puerto. El resultado deber ser similar al de la Figura 19.



**Figura 19: Resultado del paso 6: conexionado de la electrónica de la tapa superior y la inferior.
Elaboración en Paint.**

7. Paso: Atornillar la tapa superior al conjunto de la tapa inferior y el lateral.

Por último, se deberán atornillar mediante tornillos de M3 longitud 8 mm la tapa superior y el conjunto formado por el lateral y la tapa inferior. El resultado de esta operación tiene que dar como resultado un montaje como el de la Figura 20. Con esto concluye en montaje del prototipo.

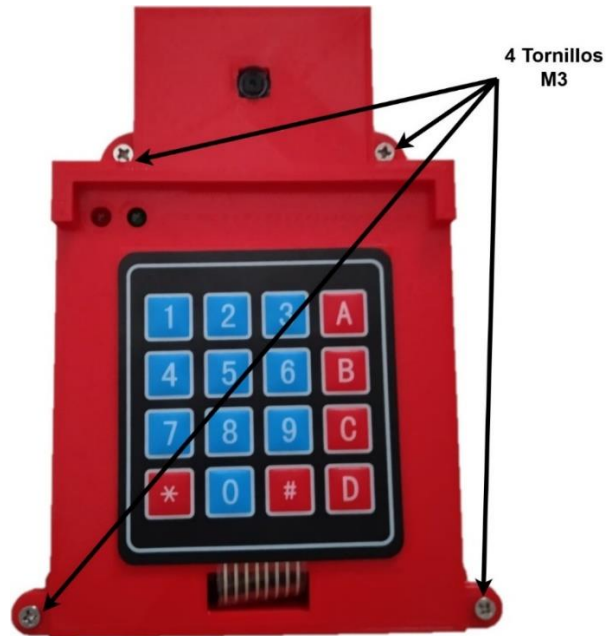


Figura 20: Prototipo montado al completo con la ubicación de la tornillería y métrica requerida en el paso 7. Elaboración propia en Paint.

6.2 DIAGRAMAS UML

En este apartado se presentan los diferentes diagramas UML que aportan una descripción más detallada del funcionamiento del sistema, con el objetivo de facilitar la comprensión de este.

6.2.1 Diagrama de casos de uso

A continuación, en la Figura 21, se muestra el diagrama de casos de uso donde participan los siguientes cinco actores: Usuario, Administrador, Raspi, Computador, Base de Datos.

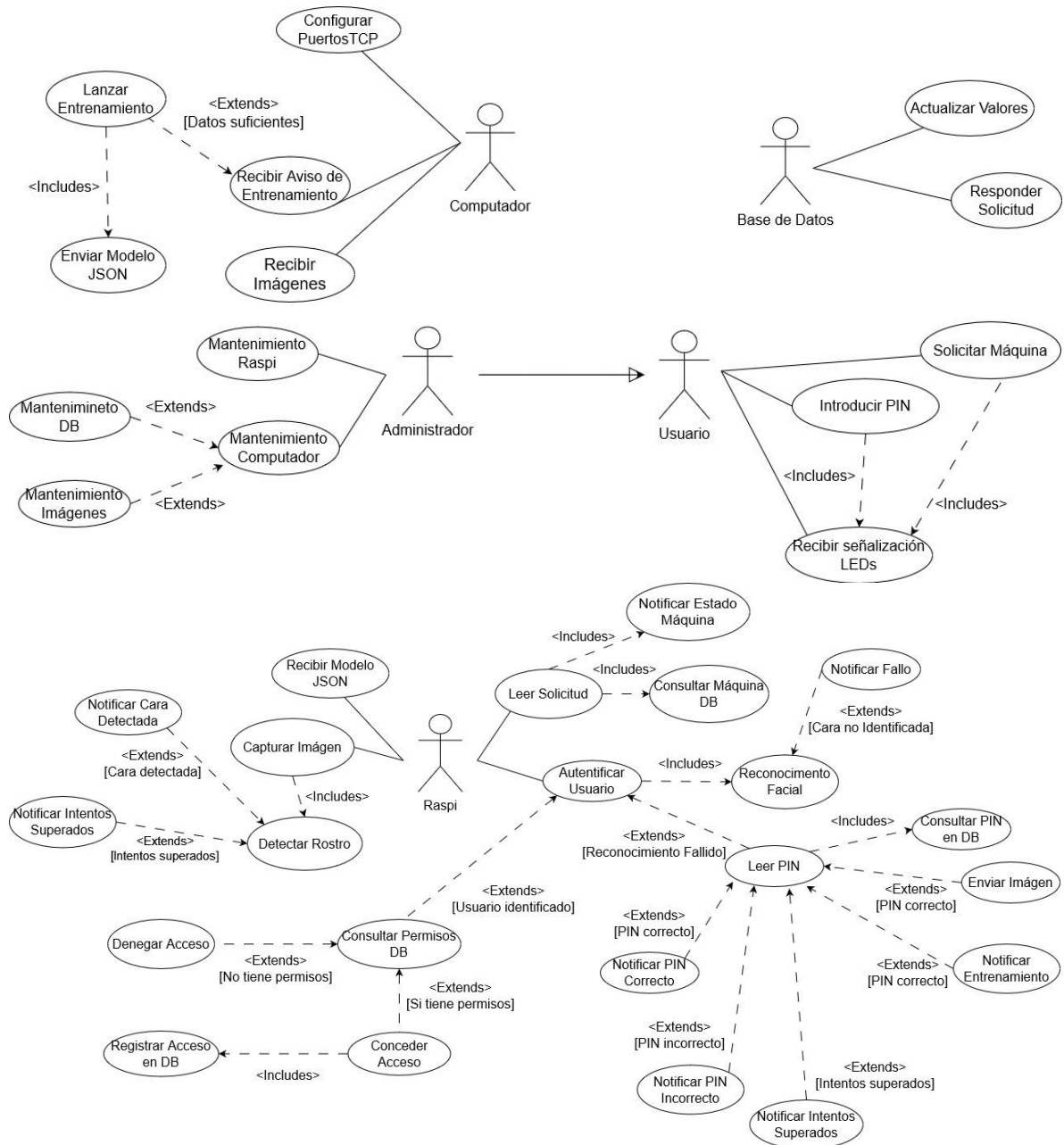


Figura 21: Diagrama de casos de usos del sistema con actores Administrador, Usuario, Computador, Base de Datos y Raspi. Elaboración propia hecha en Draw.io.

6.2.2 Casos de uso

A continuación, se recopilan y desarrollan los casos de uso más relevantes en las siguientes Tablas 12 a la 23. En algunos quedan incluida la explicación de otros casos más sencillos que se han decidió omitir. Por otro lado, también se han omitido los casos autodescriptivos que no aportan información.

Tabla 12: Caso de uso RF-01, Usuario solicita máquina.

RF-01	Solicitar maquina	
Subsistema	Front-end	
Actores	Usuario	
Objetivo	Solicitar el acceso a una máquina	
Descripción o Flujo de eventos	Paso	Acción
	1	El usuario introducirá por teclado el identificador de la máquina.
	2	Se espera a recibir una señal de los leds, se deberá encender el led verde durante 2 segundos.
Excepciones	Paso	Acción
	2	Si el identificador de la maquina introducido en el paso 1 no existe en la base de datos. El sistema encenderá el led indicador rojo. Ir al paso 1.
	2	Si el estado de la maquina es ocupado. El sistema encenderá el led indicador rojo. Ir al paso 1.
Precondiciones	Que la máquina solicitada este registrada en la base de datos, así como su estado e identificador.	
Postcondiciones	El sistema conocerá el activo libre solicitado.	

Tabla 13: Caso de uso RF-02, Usuario ingresa PIN.

RF-02	Ingresa PIN	
Subsistema	Front-end	
Actores	Usuario	
Objetivo	Introducir PIN de acceso	
Descripción o Flujo de eventos	Paso	Acción
	1	El usuario introducirá por teclado el PIN con el que está registrado en la base de datos.
	2	El usuario espera a recibir la señal lumínica, el led verde deberá parpadear tres veces.
Excepciones	Paso	Acción
	2	Si el PIN introducido en el paso 1 no es

		correcto. Se encenderá el led indicador rojo parpadeando 3 veces. Ir al paso 1.
	2	Si el PIN es incorrecto y se han superado 3 intentos. El sistema encenderá el led indicador rojo durante 2 segundos. Se reinicia el sistema.
Precondiciones	Que el sistema haya fallado el reconocimiento facial y el usuario este registrado en la base de datos.	
Postcondiciones	El sistema conocerá la identidad del usuario.	

Tabla 14: Caso de uso RF-03, Administrador realiza el mantenimiento de imágenes.

RF-03	Mantenimiento Imágenes	
Subsistema	Back-end	
Actores	Administrador	
Objetivo	Gestionar los datos de entrenamiento del modelo	
Descripción o Flujo de eventos	Paso	Acción
	1	El administrador deberá acceder al directorio "Proyecto\Caras\".
	2	El administrador deberá acceder a uno de los directorios de la carpeta y comprobar cada una de las imágenes.
	3	El administrador deberá eliminar las imágenes que no corresponden a la carpeta.
	4	El administrador deberá corregir el parámetro n_img del usuario cuya carpeta ha sido modificada.
	5	Repetir desde el paso 2 hasta haber recorrido todas las carpetas.
Precondiciones	Que sea necesario realizar una corrección en los datos porque se ha notificado o se debe realizar mantenimiento rutinario.	
Postcondiciones	Los datos almacenados en las carpetas de back-end están correctamente organizados.	

Tabla 15: Caso e uso RF-04, Administrador realiza el mantenimiento de la base de datos.

RF-04	Mantenimiento DB	
Subsistema	Back-end	
Actores	Administrador	
Objetivo	Gestionar la base de datos	
Descripción o Flujo de eventos	Paso	Acción
	1	El administrador deberá ingresar a la base de datos con su usuario y contraseña.

	2	El administrador deberá visualizar el contenido de una tabla.
	3	El administrador deberá modificar algún valor en caso de que sea necesario.
	4	El administrador deberá añadir algún elemento a la tabla en caso de que sea necesario.
	5	El administrador deberá eliminar algún elemento a la tabla en caso de que sea necesario.
	6	Repetir desde el paso 2, seleccionando la otra tabla.
Precondiciones	Que sea necesario realizar una corrección en la base de datos debido a un cambio en el sistema o debido a un mantenimiento rutinario.	
Postcondiciones	La base de datos contiene la información de todos los usuarios y maquinaria correctamente almacenada.	

Tabla 16: Caso de uso RF-05, Computador configura los puertos TCP.

RF-05	Configurar Puertos TCP	
Subsistema	Back-end	
Actores	Computador	
Objetivo	Gestiona los puertos de los Sockets TCP	
Descripción o Flujo de eventos	Paso	Acción
	1	El computador abre un puerto TCP0.
	2	El computador recibe una conexión.
	3	El computador abre los puertos TCP1 y TCP2.
	4	El computador envía el número de los puertos abiertos.
5	El computador cierra la conexión.	
Precondiciones	Que se inicie el sistema.	
Postcondiciones	La información de los puertos ha sido enviada al front-end para establecer la comunicación.	

Tabla 17: Caso de uso RF-06, Computador recibe imágenes por el socket TCP2.

RF-06	Recibir Imagen	
Subsistema	Back-end	
Actores	Computador	
Objetivo	Recibe una imagen por el puerto TCP2	
Descripción o Flujo de eventos	Paso	Acción
	1	Se establece la comunicación en el puerto.

	2	Se recibe el mensaje con la información de la imagen a recibir.
	3	Se envía un mensaje de que la información ha sido recibida.
	4	Se recibe la imagen mediante un flujo de bytes y se almacena en el directorio correspondiente al mensaje de información.
	5	Se envía una confirmación de que la imagen se ha recibido.
Precondiciones	Que se haya abierto el puerto TCP2.	
Postcondiciones	Se ha añadido una imagen al directorio del usuario que figuraba en el mensaje de información.	

Tabla 18: Caso de uso RF-07, Computador lanza el entrenamiento de la red.

RF-07	Lanzar Entrenamiento	
Subsistema	Back-end	
Actores	Computador	
Objetivo	Se lanza el entrenamiento de la red neuronal	
Descripción o Flujo de eventos	Paso	Acción
	1	Se recibe un aviso de que se puede entrenar en el socket TCP1.
	2	Se comprueban si la cantidad de datos es suficiente para entrenar
	3	Se entrena la red neuronal y se guarda una copia del modelo entrenado.
	4	Se realiza la conversión del modelo al formato JSON.
5	Se envía el JSON al front-end por el puerto TCP3.	
Excepciones	Paso	Acción
	2	Si la cantidad de datos no es suficiente se cancela la operación.
Precondiciones	Que se haya abierto el puerto TCP1 y se reciba un aviso de entrenamiento.	
Postcondiciones	El front-end cuenta con una copia del modelo JSON	

Tabla 19: Caso de uso RF-08, Raspi lee la solicitud de maquinaria.

RF-08	Leer Solicitud	
Subsistema	Front-end	
Actores	Raspi	
Objetivo	Se lee del teclado el indicador de máquina	
Descripción o Flujo de	Paso	Acción

eventos	1	Se lee el teclado hasta que se pulsa el botón “*”.
	2	Se consulta en la base de datos el identificador leído.
	3	Se señala el estado de la máquina mediante los leds.
Precondiciones	El usuario ha introducido el identificador de la maquinaria.	
Postcondiciones	El front-end almacena la información de que maquina se está solicitando.	

Tabla 20: Caso de uso RF-09, Raspi toma una captura de la cara del usuario.

RF-09	Capturar Imagen	
Subsistema	Front-end	
Actores	Raspi	
Objetivo	Se toma una foto y se detecta la cara.	
Descripción o Flujo de eventos	Paso	Acción
	1	Se espera 3 segundo.
	2	Se toma una foto del usuario.
	3	Se extrae el ROI de la cara del usuario.
	4	Se ilumina el led verde parpadeando 3 veces.
	5	Se almacena en el front-end la imagen de la cara recortada y se borra la original.
Excepciones	Paso	Acción
	3	Si no se ha detectado cara y no se han superado los intentos parpadea el led rojo 3 veces. Ir al paso 1.
	3	Si no se ha detectado cara y se han superado los intentos el led rojo se ilumina 2 segundos. Se cancela la operación
Precondiciones	Se debe haber leído la solicitud de una máquina con estado libre.	
Postcondiciones	El fron-end cuenta con una la cara recortada almacenada y conoce el activo seleccionado por el usuario.	

Tabla 21: Caso de uso RF-10, Raspi realiza la autenticación del usuario.

RF-10	Autenticación
Subsistema	Front-end

Actores	Raspi	
Objetivo	Se identifica al usuario	
Descripción o Flujo de eventos	Paso	Acción
	1	Se procede al reconocimiento facial. Si falla ir al paso 4.
	2	Si el usuario ha sido identificado, se consultan en la base de datos los permisos del usuario.
	3	Si el usuario tiene permiso se le concede el acceso, se ilumina el led verde durante 2 segundos. Fin del caso de uso.
	4	Se procede al ingreso por PIN. Ir al paso 2.
Excepciones	Paso	Acción
	1	Si no existe modelo o falla el reconocimiento. Ir al paso 4.
	3	Si el usuario no tiene permisos se ilumina el led rojo durante 2 segundos y se cancela la operación.
	4	Si el usuario introduce un PIN equivocado parpadea el led rojo 3 veces.
	4	Si el PIN es incorrecto y se ha superado el número de intentos. Se ilumina el led rojo 2 segundos, se reinicia el sistema y se cancela la operación.
Precondiciones	Se debe tener la cara recortada del usuario y conocer la máquina solicitada	
Postcondiciones	El usuario podrá acceder a la maquinaria.	

Tabla 22: Caso de uso RF-11, Raspi realiza el reconocimiento facial.

RF-11	Reconocimiento Facial	
Subsistema	Front-end	
Actores	Raspi	
Objetivo	Se identifica al usuario por reconocimiento facial	
Descripción o Flujo de eventos	Paso	Acción
	1	Se comprueba si existe un modelo de red almacenado.
	2	Se ilumina los dos leds de manera simultánea por 2 segundos.
	3	Se extrae el vector de características de la cara recortada.
	4	Se pasa el vector de características por la red neuronal y se obtiene la id del usuario.
Excepciones	Paso	Acción
	1	Si no existe modelo se cancela la operación.
	4	Si no la id predicha no supera un valor de

	confianza, se cancela la operación.
Precondiciones	Se debe haber iniciado la autenticación.
Postcondiciones	Se conocerá la identidad del usuario.

Tabla 23: Caso de uso RF-12, Raspi realiza la lectura del PIN.

RF-12	Leer PIN	
Subsistema	Front-end	
Actores	Raspi	
Objetivo	Se identifica al usuario por PIN	
Descripción o Flujo de eventos	Paso	Acción
	1	Se iluminan los leds rojo y verde alternando.
	2	Se lee el PIN introducido por teclado.
	3	Se consulta la base de datos si hay coincidencias con el PIN.
	4	Se ilumina el led verde parpadeando 3 veces.
	5	Se envía la imagen de la cara recortada por el socket TCP2.
	6	Se elimina la cara recortada almacenada en el front-end.
	7	Se envía la notificación de entrenamiento por el socket TCP1 si se dan las condiciones.
Excepciones	Paso	Acción
	4	Si el PIN es incorrecto, se ilumina el led rojo parpadeando 3 veces. Ir al paso 2.
	4	Si el PIN es incorrecto y se han superado los intentos, se ilumina el led rojo 2 segundos. Cancelar la operación.
Precondiciones	Se debe haber iniciado la autenticación y fallado el reconocimiento facial.	
Postcondiciones	Se conocerá la identidad del usuario.	

6.2.3 Diagrama de interacción

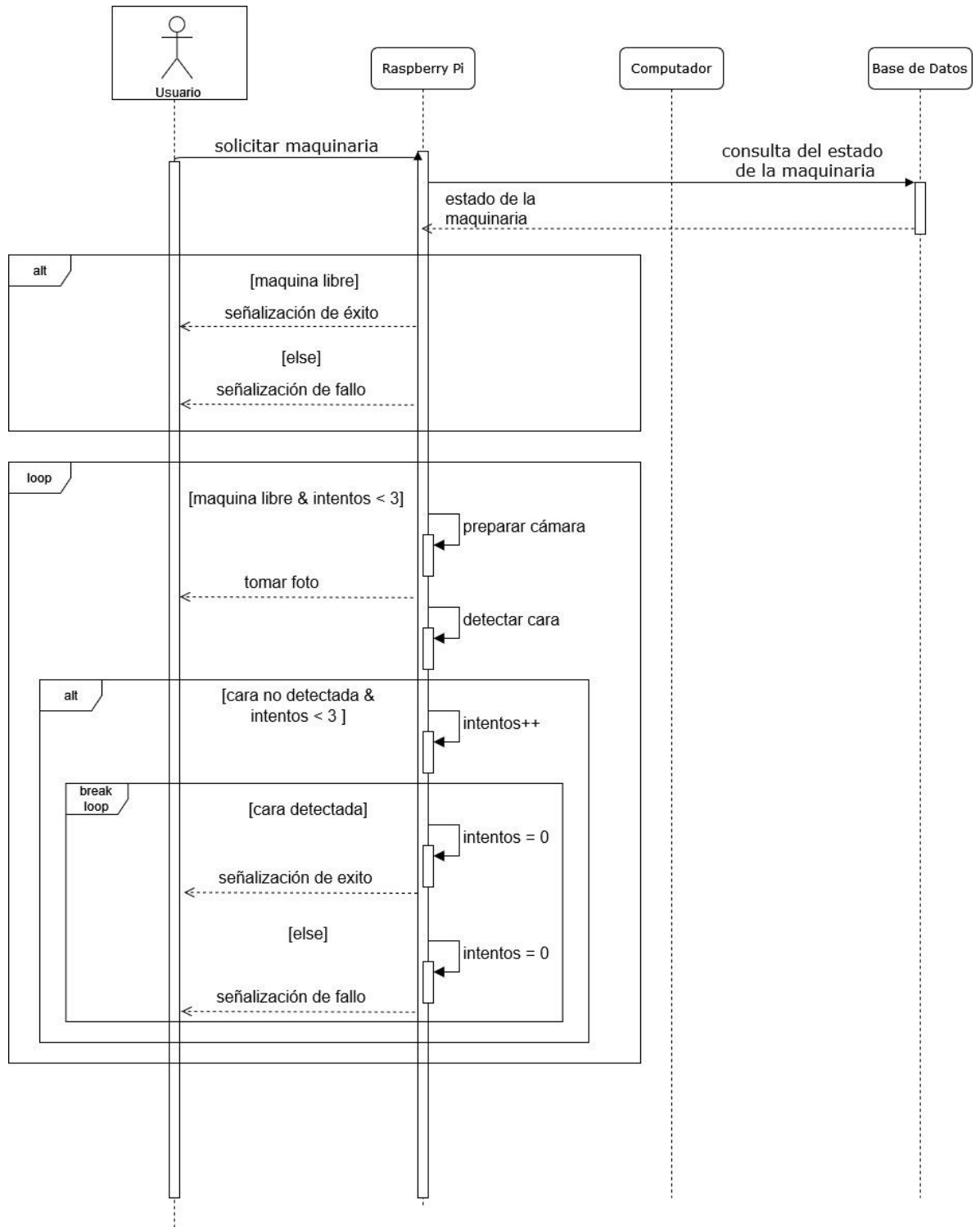


Figura 22: Diagrama de interacción del sistema, parte 1. Elaboración propia mediante Drwa.io.

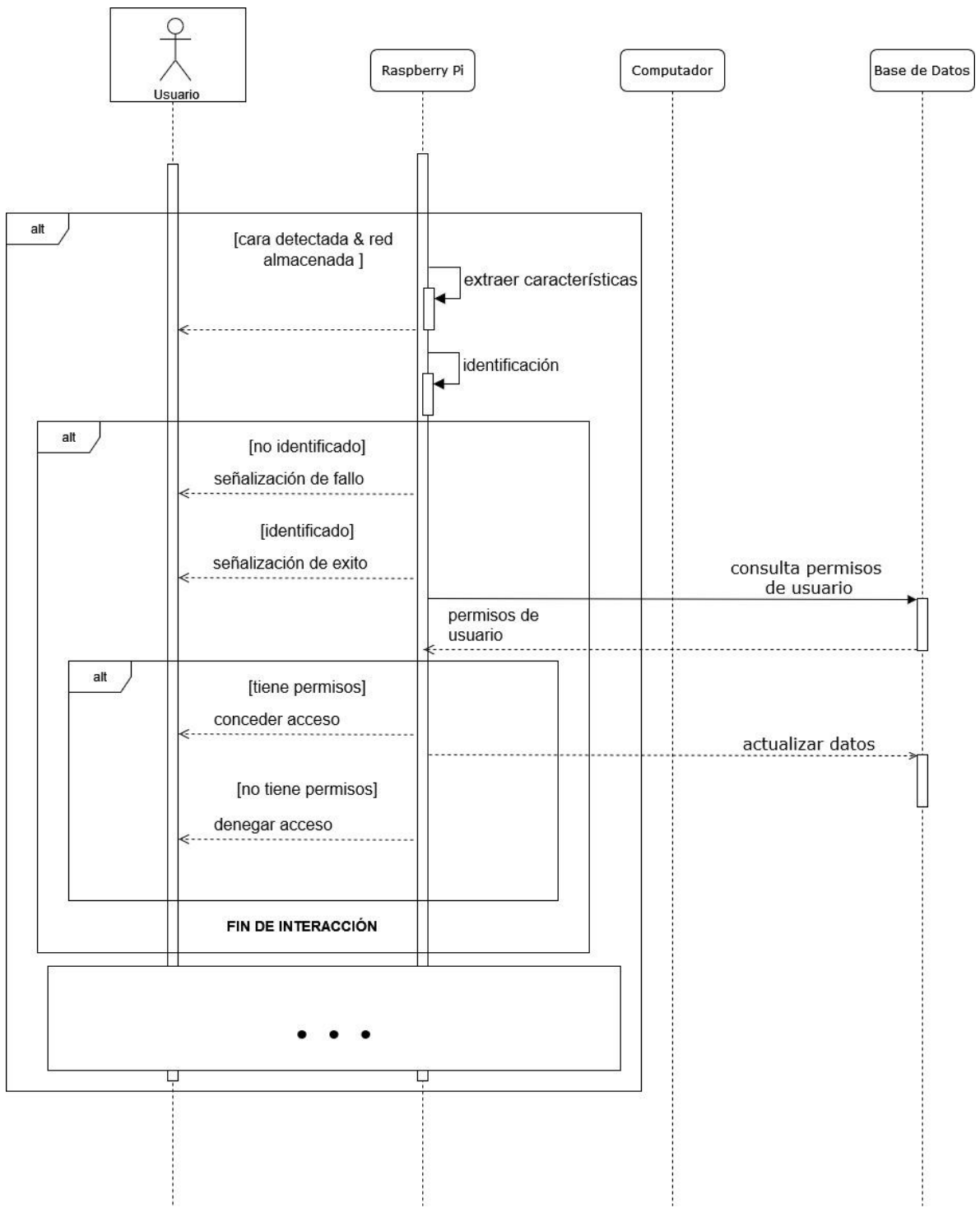


Figura 23: Diagrama de interacción del sistema, parte 2. Elaboración propia mediante Drwa.io.

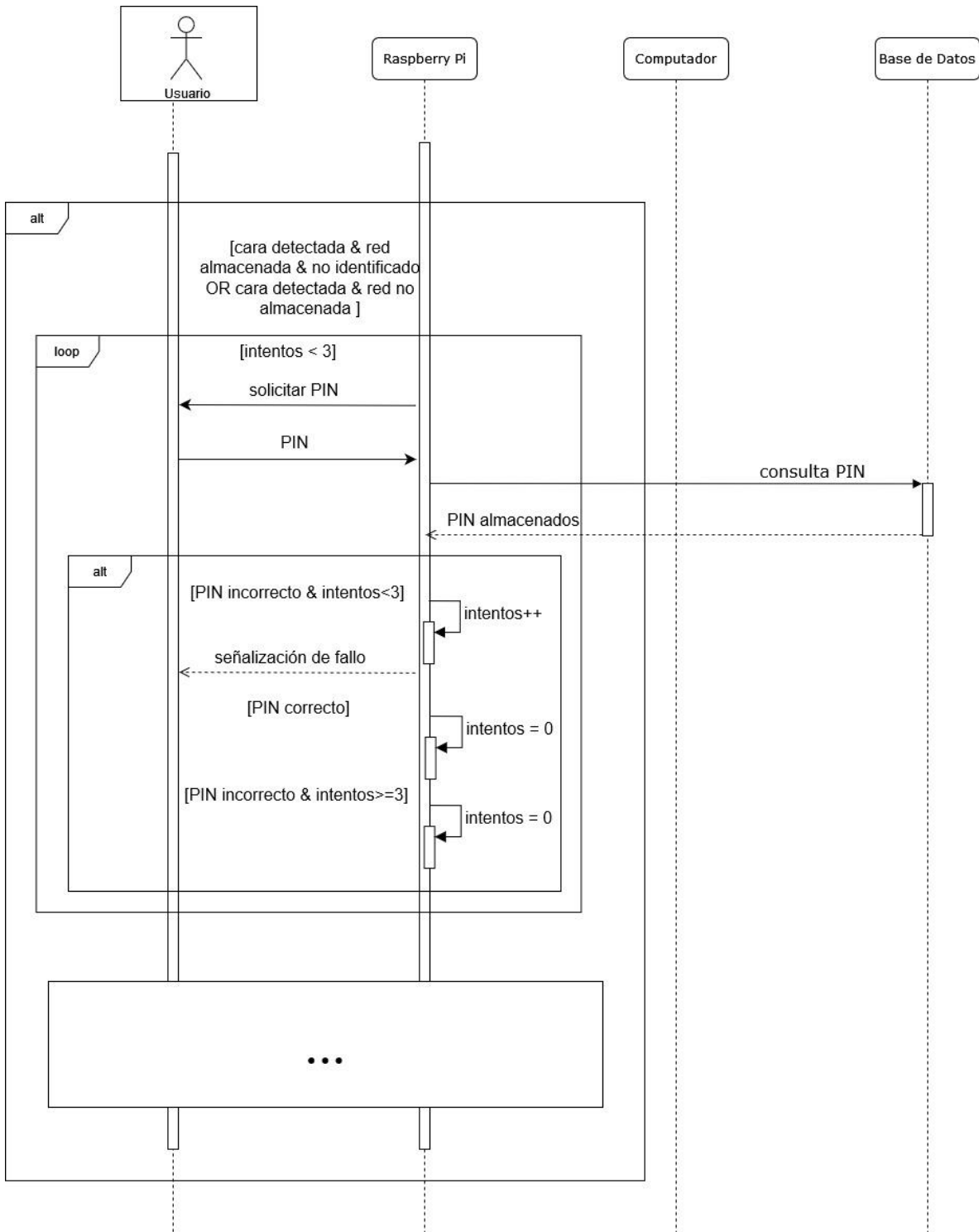


Figura 24: Diagrama de interacción del sistema, parte 3. Elaboración propia mediante Drwa.io.

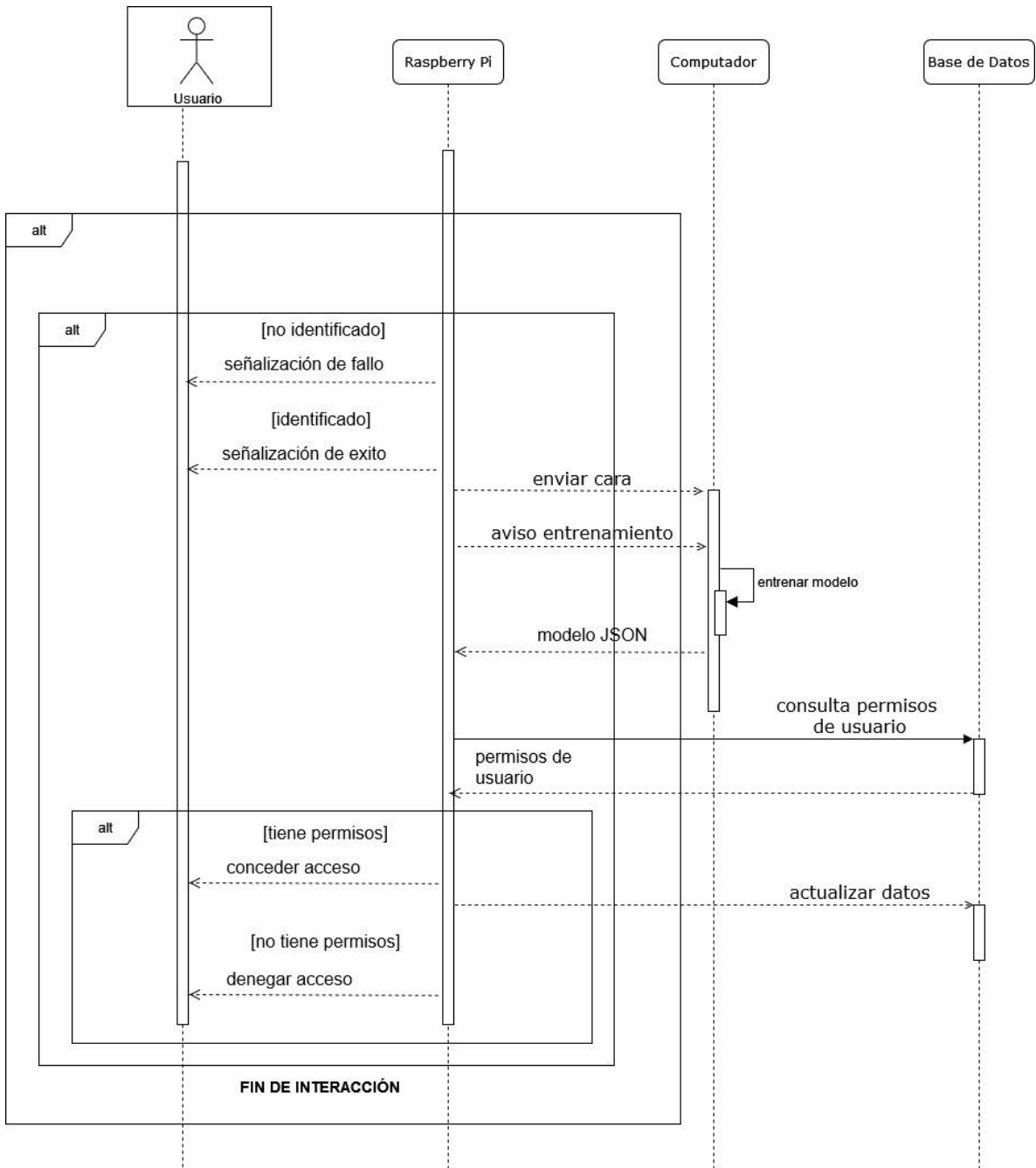


Figura 25: Diagrama de interacción del sistema, parte 4. Elaboración propia mediante Drwa.io.

6.2.4 Diagrama de clases

En la siguiente Figura 26 se muestran el diagrama de clases que se han elaborado para el proyecto.

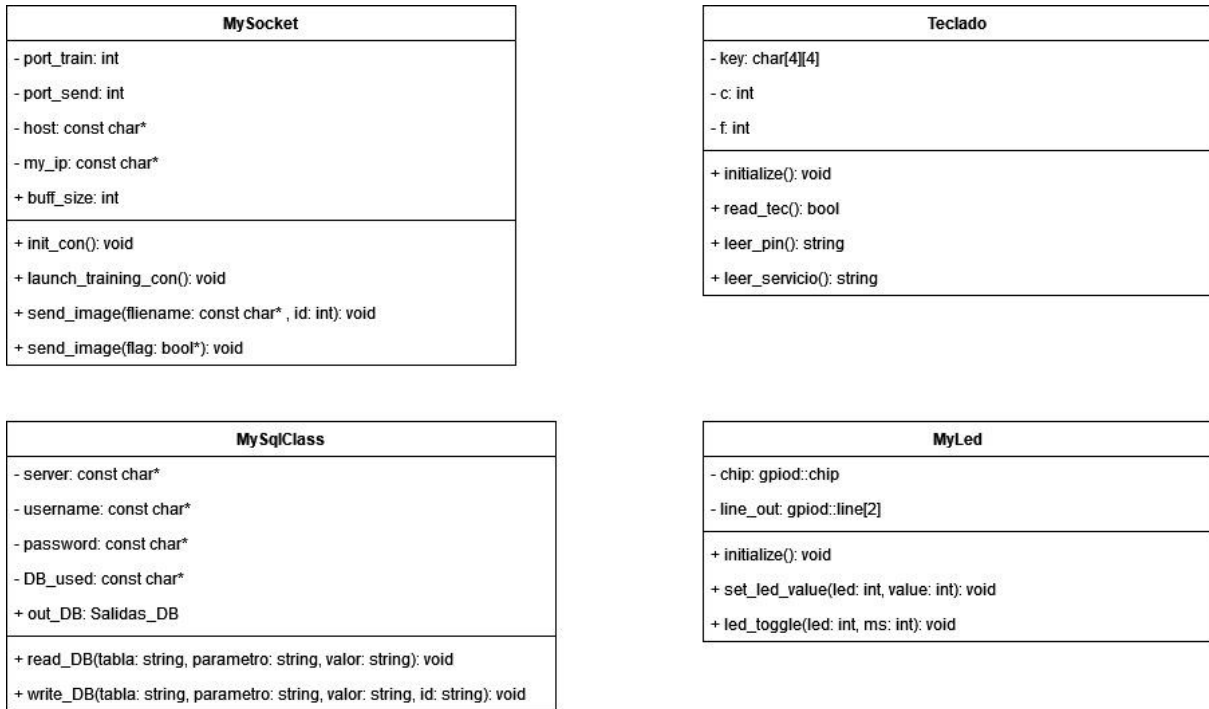


Figura 26: Diagrama de clases del sistema. Elaboración propia en Draw.io.

6.3 INSTALACION Y DESPLIEGUE

En este apartado se atenderá a las cuestiones necesarias para la correcta puesta en marcha del sistema. Considerando la compilación del código para el nodo front-end, la instalación del nodo, así como la ejecución del código.

6.3.1 Compilación

El código propio del front-end se deberá compilar mediante compilación cruzada, para ello será necesario instalar las herramientas de compilación de la arquitectura de la Raspberry Pi y las librerías para esta versión. Para esto se llevará a cabo la instalación de las librerías en la Raspberry y posteriormente se extraerá una copia de estas en el computador compilador. Una vez realizado esta operación se deberá proceder a la compilación cruzado con entorno de desarrollo adecuado para esta tarea, por ejemplo, Eclipse CDT.

6.3.2 Instalación

El nodo frot-end se deberá instalar conectando la alimentación y el ethernet a los puertos abiertos del encapsulado. Adicionalmente se deberá conectar la salida digital para el control del acceso a la maquina a uno de los pines libres de la Raspberry Pi. Además, se deberá configurar la base de datos y los archivos respectivos a las comunicaciones TCP, este proceso queda detallado en el apartado 6.4.2.3.

6.3.3 Despliegue

Una vez configurado e instalado el nodo front-end, se deberá iniciar el back-end asegurando que se ha iniciado el servicio de la base de datos. Posteriormente se ejecutarán los archivos `sock_mult.py` y `sock_train.py`. Finalmente, se encenderá el front-end que iniciará como servicio el control de accesos.

6.4 INSTRUCCIONES DE USO

Este apartado contine los manuales de usuario y administrador del sistema con el objetivo de dar una guía para la interacción con el prototipo y el mantenimiento del sistema. El manual de usuario, se centra en cómo utilizar el prototipo y no tanto en cómo funciona el sistema. En el caso del administrador, se enfoca más en cómo poner en funcionamiento el sistema y como realizar el mantenimiento de este.

6.4.1 Manual de USUARIO

6.4.1.1 Introducción

El dispositivo con el que se interactúa sirve para permitir el acceso a una determinada maquinaria. Previamente a la interacción con el mismo debe asegurarse de que consta como que está registrado en la base de datos del sistema y de que tiene permisos para utilizar la máquina.

6.4.1.2 Descripción del sistema

El dispositivo cuenta con tres elementos fundamentales que lo componen el teclado, dos indicadores led, y la cámara.

- **Teclado:** Mediante este componente, deberá introducir tanto su pin como el identificador de la maquina cuando se le solicite.
- **Indicadores led:** Estos dos diodos led se encenderán de una manera determinada para señalar el estado del sistema. Más adelante se explican que significa cada señal y como se debe proceder.
- **Cámara:** Este elemento tomará una foto cuando el sistema lo indique mediante los leds. Es importante que a la hora de tomar la foto se coloque a una distancia no menor a X cm y deberá asegurarse de que mira de frente a la lente durante la foto asegurándose de que su cara aparezca en la imagen capturada.

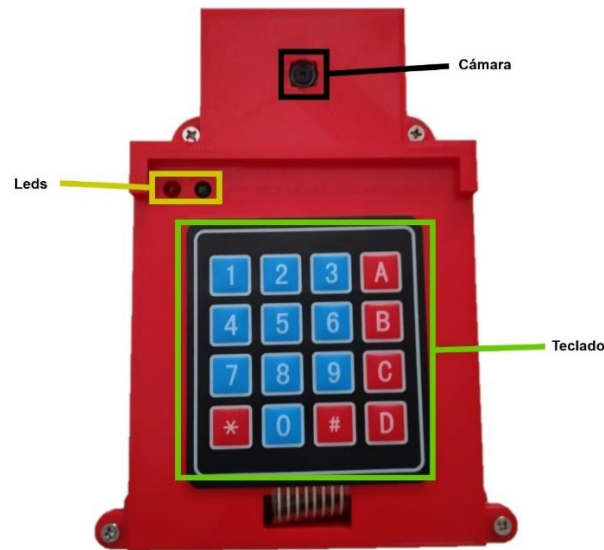


Figura 27: Elementos fundamentales del prototipo. Elaboración propia empleado Paint.

6.4.1.3 Operación diaria

A continuación, se describe la secuencia de operaciones para interactuar con el dispositivo. Previamente a utilizar el dispositivo asegúrese de que se encuentra registrado en la base de datos y sus credenciales.

1. Se deberá ingresar mediante el teclado el identificador de la máquina a la que se quiere acceder. En el caso de que la máquina no se encuentre disponible se encenderá el indicador rojo durante 2 segundos, si por el contrario la máquina está libre se iluminará el indicador verde.
2. Posteriormente se deberá colocar frente al dispositivo a una distancia tal que su cara sea visible por la cámara. Tras 3 segundos de espera, se tomará una foto de su cara, si se ha detectado la cara correctamente se parpadeará el indicador verde 3 veces. Si no su rostro no ha sido detectado se indicará mediante el parpadeo del indicador rojo y se volverá a tomar otra foto. En el caso de que se haya repetido este proceso 3 veces y no se detecte la cara, el indicador rojo no parpadeará y se mantendrá iluminado durante 2 segundos, entonces se volverá al paso 1.
3. Se procede a la autenticación del usuario, si el usuario es reconocido de forma automática se iluminarán ambos indicadores durante 2 segundos, ver paso 4. Si no es reconocido automáticamente se deberá proceder al ingreso mediante PIN, se iluminarán los indicadores alternativamente. Para el ingreso por PIN, ver paso 5.
4. Se procede a evaluar los permisos del usuario, si cuenta con los permisos necesarios se iluminará el led verde 2 segundos indicando que se concede

el acceso, aquí termina su interacción con el dispositivo. En cambio, si se enciende el led rojo, no se cuenta con los permisos necesarios, se niega el acceso, volver al paso 1.

5. Se procede a la autenticación mediante el ingreso del PIN. Se deberá ingresar la clave de usuario por el teclado. Si el PIN se reconoce, parpadeará el led verde 3 veces, ir al paso 4. En cambio, si se introduce mal el PIN, parpadeará el indicador rojo y deberá volver a ingresar el PIN. Si se han superado 3 intentos el sistema se reiniciará y el led rojo brillará durante 2 segundos, ir al paso 1.

6.4.1.4 Uso del teclado

El teclado este compuesto por tres partes bien diferenciadas, el botón de confirmación “ * ”, que se pulsa para introducir el final de la cadena de caracteres introducidos, el botón de borrar el ultimo carácter introducido “ # ” y el resto de botones que corresponden a identificadores o caracteres.

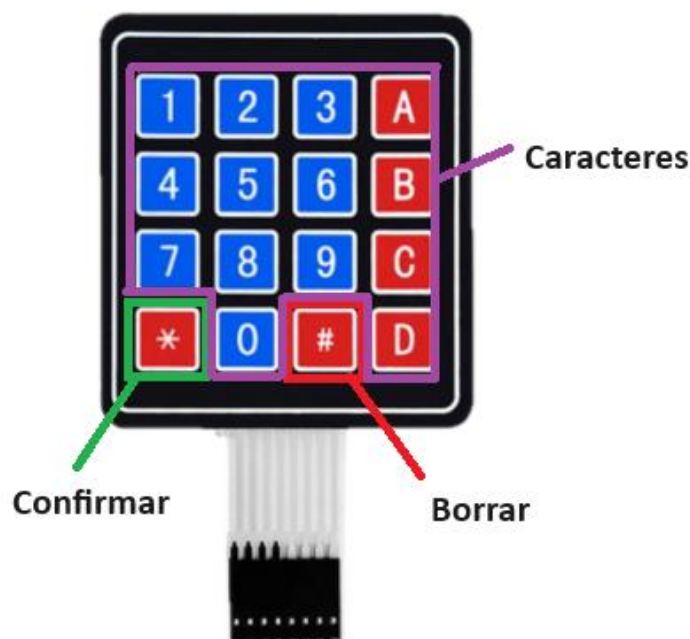


Figura 28: Teclado matricial señalizando el rol de los botones. Editado mediante Paint.

6.4.1.5 Señal de leds

Explicación por pasos como utilizar el sistema, primero elegir la maquinaria

Tabla 24: Señales luminosas del prototipo junto con su duración, evento previo y significado.

Led	Duración	Evento	Significado
-----	----------	--------	-------------

Rojo	2 s	Tras introducir el identificador de maquina	El activo no esta disponible o no se introdujo correctamente el identificador
Verde	2 s	Tras introducir el identificador de maquina	El activo solicitado está disponible
Rojo	Parpadea 3 veces	Tras sacar la foto	No se ha detectado la cara se procederá a sacar otra foto.
Rojo	2 s	Tras sacar la foto	No se ha detectado la cara y se han superado los intentos máximos, se reinicia el sistema
Verde	Parpadea 3 veces	Tras sacar la foto	Se ha detectado la cara
Ambos	2s	Tras detectar la cara	Se ha reconocido automáticamente
Ambos	Se iluminan alternado	Tras detectar la cara	No se ha reconocido automáticamente, deberá ingresar el PIN
Rojo	2 s	Tras completar la autenticación	Acceso denegado debido a que no tiene permisos para acceder al activo
Verde	2 s	Tras completar la autenticación	Acceso concedido
Verde	Parpadea 3 veces	Tras ingresar el PIN	PIN correcto
Rojo	Parpadea 3 veces	Tras ingresar el PIN	PIN incorrecto, vuelva a introducir el PIN
Rojo	2 s	Tras ingresar el PIN	Numero de intentos superado, reinicio del sistema
Rojo	Permanece encendido	-	Ha habido una falla en las comunicaciones del sistema. Avisar administrador.

6.4.1.6 Resolución de problemas comunes

A continuación, se muestra un conjunto de posibles problemas que se puedan producir y sus soluciones.

1. No se reconoce el pin

En el caso de que cuando ingrese su pin durante la fase correspondiente. Deberá asegurarse de que el pin se ha introducido correctamente. Si sigue fallando, deberá consultar a un administrado si usted figura en la base de datos del sistema y si los datos que figuran son correctos.

2. El sistema no reconoce la cara

Es normal que el sistema no reconozca la cara de los usuarios las primeras veces debido a que necesita varios ingresos previos al reconocimiento facial. Si aun así el sistema no reconoce su cara póngase en contacto con un administrador.

3. El sistema no reconoce el identificador de maquina

Este error puede deberse a que no se ha introducido correctamente el identificador. En el caso de que el identificador sea el correcto puede deberse a una pérdida o fallo en la conexión, se deberá informar a un administrador de este problema.

6.4.2 Manual de ADMINISTRADOR

En este apartado se desarrolla la interacción con el sistema desde el punto de vista del rol de administrador cuyas funciones principales son las de gestionar el mantenimiento del sistema software y hardware.

6.4.2.1 Introducción

El sistema de control de accesos se compone de dos nodos, uno de front-end y otro de back-end. El primero tendrá la función de la interacción con los usuarios, tomando fotos, leyendo datos por teclado y dando indicaciones mediante leds. El segundo nodo corresponde con un computador que tendrá como función el almacenamiento de los datos y mantener activo el servicio de la base de datos. Se deberá gestionar el mantenimiento del sistema para evitar problemas durante el funcionamiento de mismo.

6.4.2.2 Rutas del sistema

Es posible que durante el mantenimiento del sistema se deban consultar algunos archivos almacenados como podrían ser las imágenes de los usuarios, así que en este apartado se describe como está organizado el sistema de archivos de ambos nodos.

Nodo front-end:

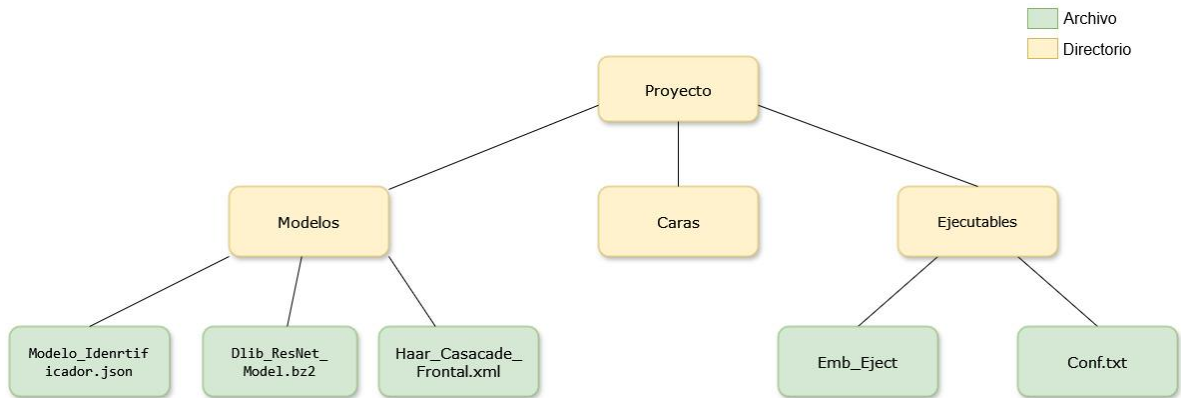


Figura 29: Árbol de archivos de la aplicación correspondiente al front-end. Elaboración propia empleando Draw.io.

En el directorio caras se almacena la última foto tomada por el nodo antes del envío de la imagen. En el directorio Ejecutables, encontramos el programa del nodo y un archivo Config.txt, este último es el que se debe configurar para asignar la ip del nodo back-end y los datos para ingresar acceder a la base de datos. Por último, está el directorio modelo que almacenan herramientas que emplea el programa, no se deberá modificar ningún archivo del directorio exceptuando el JSON que es el que genera el back-end.

Nodo back-end:

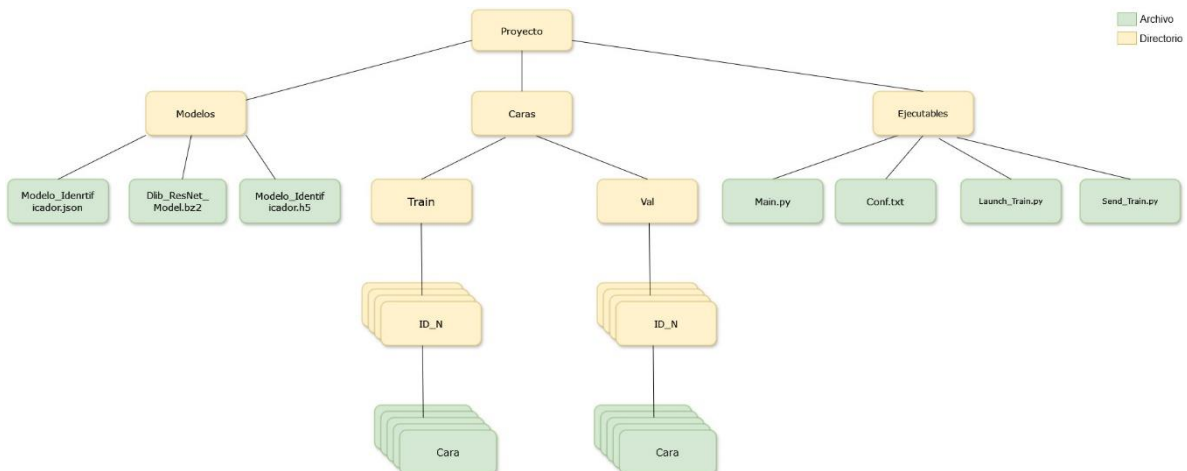


Figura 30: Árbol de archivos de la aplicación correspondiente al back-end. Elaboración propia empleando Draw.io.

En este nodo encontramos el directorio caras con un conjunto de carpetas train y val que dentro tienen un conjunto de carpetas que almacenan las caras de todos los usuarios definidos por su ID. Será necesaria la revisión de estas carpetas para comprobar que las caras están en las carpetas de su ID correspondientes. Por otro lado, está el directorio modelos, que almacena los modelos generados y el modelo .bz2 que no se deberá tocar. Finalmente, el directorio ejecutable,

almacena los scripts .py que necesita el sistema, junto con el archivo config.txt que se deberá configurar para introducir la ip de los nodos front-end.

6.4.2.3 Configuración inicial

Para que el sistema funcione correctamente se debe realizar la configuración inicial del mismo, esta queda definida en los siguientes pasos:

1 En primer lugar, se deberá conectar el nodo de front-end a la red, posteriormente, se deberá obtener la ip, y anotarla en el archivo config.txt tanto del front-end y el back-end.

2 En segundo lugar, se deberá poner en marcha el servidor MySQL, primero se deberá descargar e instalar el software. Posteriormente, se deberá configurar crear la base de datos con dos tablas asociadas, usuarios y maquinaria. El formato de las tablas se puede observar en la Tabla 9 y 10 del apartado de metodologías. Además, se deberá registrar el usuario y garantizar permisos de lectura y escritura a los nodos front-end y back-end, para que puedan acceder a la base de datos. Por último, se deberá añadir la información respectiva al usuario de MySQL de cada nodo en su respectivo archivo config.txt.

3 Por último, configurar los puertos de cada archivo config.txt y comprobar que las ip y las credenciales de inicio en la base de datos son correctos, el resultado debería ser similar al de las figuras 27 y 28. Una vez, se ha comprobado que los .txt están bien configurados únicamente falta ejecutar el archivo main.py del back-end, activar el servicio de la base de datos y encender el front-end.

```
MyIP: :192.168.1.170
MyMult_Port:6000:
MyLaunch_Port:5002:
ClientIP:192.168.1.182:
Client_Port:5003:
usuario
```

Figura 31 : Ejemplo del archivo Config.txt del back-end. Captura de pantalla archivo usando nano.

Esta figura muestra un ejemplo del contenido del archivo Config.txt del back-end donde la primera línea corresponde a la ip del servidor, la segunda a la dirección del puerto que se abre para la conexión de sockets TCP0, la tercera para el puerto del socket TCP1. Las siguientes dos líneas hacen referencia a la dirección IP del cliente y el puerto al que se deberá enviar el modelo JSON. Finalmente, la última línea es la contraseña para acceder a la base de datos.

```
192.168.1.179
192.168.1.182
tcp://192.168.1.179:3306
emb_client
3mb_cli3nt_T3st
```

Figura 32:: Ejemplo del archivo Config.txt del front-end. Captura de pantalla archivo usando nano.

Esta figura muestra un ejemplo del contenido del archivo Config.txt del front-end, donde la primera línea corresponde a la ip del servidor, la segunda a la dirección del propio front-end, la tercera hace referencia a la conexión TCP0 indicando el tipo, dirección y puerto del servidor. Por último, las dos líneas restantes corresponden al nombre y contraseña del front-end en la base de datos.

6.4.2.4 Mantenimiento y solución de problemas detallados

Es posible que durante el funcionamiento del sistema se produzca algún error por lo que en este apartado se abordarán las soluciones a los fallos más comunes que se dan en el sistema. Además, también se explica cómo realizar el de mantenimiento del sistema.

1. Consulta en la base de datos

En el caso de que se dese consultar la base de datos para verificar si son correctos los datos almacenados se ha de proceder de la siguiente forma. En primer lugar, se debe abrir un terminal e iniciar sesión como administrador de la base de datos MySQL, mediante la instrucción “mysql -u root -p” e ingresar la contraseña. Posteriormente, seleccionar la base de datos mediante la instrucción “USE nombre_base_datos” por defecto el nombre será “users”. A continuación, seleccionar la tabla que se desea consultar mediante el comando “SELECT * FROM nombre_tabla” para toda la tabla. En el caso de que se quiera consultar algún dato en concreto se deberá introducir el comando correspondiente en lenguaje SQL, por ejemplo “SELECT id, nombre FROM usuarios_registrados” para obtener todos los ids y nombres que se encuentran en la tabla usuarios_registrados.

2. Modificar datos en la base de datos

Para modificar algún parámetro en la base de datos se deberá proceder de la siguiente forma. En primer lugar, se debe abrir un terminal e iniciar sesión como administrador de la base de datos MySQL, mediante la instrucción “mysql -u root -p” e ingresar la contraseña. Posteriormente, seleccionar la base de datos mediante la instrucción “USE nombre_base_datos” por defecto el nombre será “users”. A continuación, empleara la instrucción UPDATE de la sintaxis SQL para modificar el parámetro deseado, por ejemplo “UPDATE usuarios_registrado SET n_img=12 WHERE id=1”, en este ejemplo se ha cambiado la tabla usuarios_registrados, ahora el valor de n_img, número de imágenes almacenadas,

es 12 para el usuario con id=1. Para cambiar cualquier otro valor utilizar una sintaxis similar para modificar la tabla y el parámetro que se desea.

3. Añadir usuarios registrados en el sistema

En el caso de que se tengan que registrar un usuario nuevo en el sistema se deberá cumplir con las siguientes instrucciones. En primer lugar, se debe abrir un terminal e iniciar sesión como administrador de la base de datos MySQL, mediante la instrucción “mysql -u root -p” e ingresar la contraseña. Posteriormente, seleccionar la base de datos mediante la instrucción “USE nombre_base_datos” por defecto el nombre será “users”. A continuación, emplear la siguiente instrucción “INSERT INTO usuarios_registrados VALUES (valor1, valor2, valor3 ...)” donde usuarios_registrados es la tabla por modificar y los valores entre paréntesis son los datos del usuario siguiendo el orden descrito en las tablas del apartado configuración inicial. Junto con esto se deberá crear una carpeta en el directorio de “caras” con nombre igual a la id del mismo.

4. Eliminar usuarios del sistema

En el caso de que se tengan que registrar un usuario nuevo en el sistema se deberá cumplir con las siguientes instrucciones. En primer lugar, se debe abrir un terminal e iniciar sesión como administrador de la base de datos MySQL, mediante la instrucción “mysql -u root -p” e ingresar la contraseña. Posteriormente, seleccionar la base de datos mediante la instrucción “USE nombre_base_datos” por defecto el nombre será “users”. A continuación, utilizar la instrucción “DELETE FROM usuarios_registrados WHERE id= id_del_usuario”, de esta forma se eliminará de la base de datos el usuario con la id introducida. Junto con esto se deberá eliminar la carpeta en el directorio de “caras” con nombre igual a la id del mismo.

5. Consultar y modificar imágenes almacenadas

Para consultar las imágenes que se han ido almacenado durante el funcionamiento del sistema se deberá acceder a las carpetas dentro del directorio caras del back-end, la ubicación de este directorio esta descrito en el apartado de rutas del sistema. Si recorriendo los directorios de imágenes se encuentra alguna que no corresponde con la cara del usuario se deberá eliminar o reubicar la imagen en el caso de que el usuario de la imagen sea diferente al de la carpeta. Para ambos casos se deberá modificar en la base de datos los parámetros n_img de los usuarios afectados, el valor del parámetro debe coincidir con el número de imágenes de la carpeta del usuario.

6. Fallo en la conexión.

Si se encuentra que el sistema no puede establecer la comunicación entre ambos nodos puede deberse a una mala configuración de los archivos con la información de la comunicación. Se deberá revisar que las ips y puertos de los archivos

config.txt de los nodos tengan la información correcta de las direcciones del sistema.

7. ANALISIS Y DISCUSION DE RESULTADOS

A continuación, se presentará un análisis de los resultados obtenidos discutiendo tanto el rendimiento del sistema como el cumplimiento de los requisitos especificados en el apartado 5.1.

7.1 RENDIMIENTO

Tras la puesta a prueba del prototipo podemos determinar que el consumo de recursos que representa el sistema en el nodo front-end está entorno a los 150 MB siendo el segundo donde más memoria necesita el sistema debido a que está efectuando las tareas más pesadas y tiene el modelo del clasificador cargado en memoria. Es importante tener en consideración este aspecto puesto que un criterio importante en los sistemas empujados es la limitación de los recursos.

Respecto a la velocidad de reconocimiento frente al ingreso por PIN, se esperaba que el último implicase un mayor tiempo en el proceso de identificación. Esta hipótesis, es debida a que, en la autenticación por PIN, existen dos tareas secuenciales, la primera el ingreso por teclado y la segunda la consulta de la base de datos. En el primer caso, se requiere la intervención humana y esto afecta en el tiempo para completar el proceso total. Por otro lado, la consulta en la base de datos dependerá de la saturación de la red. En cambio, la autenticación mediante reconocimiento facial se presupone más rápida debido a que no requiere de intervención humana, únicamente depende de la velocidad de extracción de características y el paso del vector por el modelo de red neuronal. Finalmente se obtuvieron tiempos de 11.51 segundos para la extracción de características, 0.000573 segundos para el clasificador siendo un total de 11.510573 segundos. En la autenticación por PIN, se obtuvieron un tiempo de 3.393 segundos para introducir el PIN. Cabe destacar que la medición del tiempo en ingresar la contraseña puede presentar grandes variaciones ya que depende del factor humano y a su vez se ha considerado que el PIN se introduce correctamente en el primer intento, pero esto no siempre es así. Además, el número de personas sobre las que se ha medido este tiempo es reducido por lo que tampoco representa de manera precisa el tiempo medio total de este método.

Sin embargo, y visto los resultados obtenidos se puede afirmar que el ingreso por reconocimiento facial es más lento que el ingreso por PIN, debido al elevado tiempo de extracción de características que alcanza el orden de los 11.5146 segundos en completar la extracción. Esto se debe a que el modelo ResNet para la extracción de características que facilita la librería dlib es muy pesado influyendo el resultado de tiempos que hemos obtenido. Una solución a este problema, es emplear un modelo más sencillo para esta tarea pudiendo reducir así los tiempos en gran medida. Otra alternativa es, realizar la extracción de

características en el back-end y enviar los resultados al front-end pero este acercamiento presenta ciertos inconvenientes principalmente dificulta la escalabilidad del sistema añadiendo más carga computacional al servidor. Pese a estos inconvenientes, la velocidad que se ha obtenido con este último método llegando a un tiempo de 0.0519s de extracción de características, lo que resulta en un tiempo total de autenticación de 0.05195 segundos.

7.2 RELACION CON LOS REQUISITOS DEL SISTEMA

Atendiendo con los requisitos funcionales descritos al inicio, el sistema cumple las especificaciones.

- **Detección facial:** el sistema realiza la adquisición de imagen mediante la cámara RaspiCam module 2, y procesa la imagen mediante la funcionalidad que ofrece OpenCV usando el algoritmo haar.
- **Autenticación facial:** el sistema reconoce a los usuarios registrados y devuelve como resultado la id del mismo. Esto se consigue mediante el modelo entrenado en Keras y la extracción de características de la librería Dlib.
- **Control de acceso:** el sistema consulta los permisos de cada usuario antes de denegar o permitir el acceso a una máquina. Teniendo en cuenta previamente el estado de ocupación de esta.
- **Autenticación por PIN:** el sistema autentifica al usuario mediante el ingreso del PIN por el teclado matricial cuando el reconocimiento facial falla al identificar al usuario o todavía no se ha generado ningún modelo.
- **Entrenamiento automático:** el sistema lanza el entrenamiento de la red neuronal cuando se cumple la condición de que el número de imágenes para al menos dos usuarios es suficiente para entrenar. Esto se consigue mediante la comunicación del front-end y el back-end avisando de cuando se puede realizar el entrenamiento.
- **Registro de accesos:** el sistema cuenta con una base de datos MySQL con una tabla destinada a almacenar los datos de usuarios y maquinaria. Entre estos datos, se encuentra el valor de hora de ingreso que se va actualizando cada vez que se accede a un servicio.
- **Acceso a la base de datos:** tanto el front-end como el back-end pueden consultar la base de datos para leer la información

almacenada y cambiar los valores necesarios. Esto se consigue mediante las librerías de mysqlconnector y un registro previo de los usuarios de ambos nodos y con privilegios para acceder a las tablas.

- Mantenimiento de la base de datos: el sistema da lugar a poder acceder a la base de datos por un administrador para corregir posibles erratas en las tablas. Así como la posibilidad de registrar nuevos usuarios o maquinaria en el sistema.
- Mantenimiento de imágenes: el sistema da lugar a poder alterar el contenido de las carpetas con las imágenes almacenadas para corregir errores en el almacenamiento. Estas correcciones deben ir acompañadas de cambios en la base datos.

Respecto a los requisitos no funcionales hay que destacar que el prototipo se encuentra dentro de las dimensiones especificadas. Siendo de un tamaño 155x120x50 mm menor a los 170x130x60 mm máximos. Un peso de 249 gramos inferior a los 400 gramos máximos. Por último, la cámara captura imágenes en la resolución y formato especificados.

7.3 LIMITACIONES

Dentro del desarrollo del sistema se identificaron las siguientes limitaciones:

- Potencial escalabilidad: el sistema actual no presenta la posibilidad de funcionar con varios nodos front-end de manera simultánea, ya que se diseñó y puso a prueba con un único nodo. De forma que, no es posible escalar el sistema en las condiciones actuales. Sin embargo, se ha observado durante el desarrollo que es viable la escalabilidad modificando el código, especialmente en la comunicación entre el front-end y el back-end. Cabe destacar que parte de estas modificaciones ya se han implementado en la comunicación respectiva al envío y recepción de imágenes.
- Falsos positivos: el sistema, aunque presenta un funcionamiento adecuado en el reconocimiento facial no está exento de falsos positivos que implican un riesgo para la seguridad y robustez de este. Por ejemplo, en los casos de utilizar una imagen impresa de la cara de un usuario se podría acceder a una maquinaria. Este comportamiento no es deseable y supone un riesgo en la seguridad del sistema. En la literatura existente sobre el reconocimiento facial se encuentran técnicas avanzadas para resolver el problema de la suplantación de identidad.
- Iluminación: el sistema es sensible a una mala iluminación, al igual que en gran parte de las aplicaciones de visión por computador, la luz del entorno

afecta al correcto funcionamiento del sistema. En el prototipo todas las etapas del reconocimiento facial se ven afectadas por la luz, especialmente la etapa inicial de detección facial. Ya que se ha observado que no se detectan rostros en la imagen para ambientes de iluminación desfavorables imposibilitando el progreso en el resto de las etapas.

8. CONCLUSIONES

En conclusión, este trabajo final de máster ha desarrollado un sistema embebido de reconocimiento facial con entrenamiento automático para el control y registro de accesos a un conjunto de maquinaria. Queda demostrado en el desarrollo del proyecto y en el cumplimiento de los requisitos del sistema, la viabilidad de esta tecnología en el campo de los sistemas empotrados aplicados a la gestión de activos y reconocimiento biométrico.

El sistema diseñado ha probado su eficacia en condiciones controladas, proporcionando una solución al problema de la autenticación y registro de los usuarios autorizados en el sistema. Sin embargo, cabe destacar que los tiempos que se han obtenido de la extracción de características son del orden de 8 segundos más lentos que el ingreso por PIN. Esto es debido a la complejidad de la red empleada para la obtención del vector. Sin embargo, se plantean acercamientos que resuelven el problema del elevado tiempo de extracción de características, pudiendo reducir este en algunos casos unas diez veces este tiempo.

Además, se identifican diversas limitaciones relacionadas con la escalabilidad y robustez del sistema. Que abren paso a posibles mejoras que den lugar a aplicaciones más seguras y arquitecturas más extensas. Para afrontar la escalabilidad, se deben añadir más nodos front-end, que en el estado actual del sistema no se contempla. Pero que implica modificar las comunicaciones existentes para incrementar la envergadura del sistema. Por otro lado, para mayor robustez y seguridad se deben aplicar técnicas para resolver el problema de la suplantación de identidad, que añaden mayor complejidad al sistema y más procesamiento de la imagen.

Otro factor limitante es la iluminación. Ya que el sistema depende de adquirir imágenes donde la cara del usuario sea diferenciable y así poder detectar su rostro. Proceso que consolida la base del reconocimiento facial. Por lo tanto, la iluminación del entorno es un elemento clave en el funcionamiento del sistema.

En resumen, el prototipo diseñado demuestra la viabilidad de esta tecnología pese a las numerosas limitaciones que se presentan. Estas limitaciones, dan lugar mejoras que conllevan al desarrollo de un sistema más completo.

9. RECOMENDACIONES Y TRABAJOS FUTUROS

Una de las líneas futuras, es la de mejorar el tiempo de obtención del vector de características, esto se consigue empleando un modelo más sencillo. Otra alternativa relacionada, es afrontar el problema de la autenticación con otro método, como una red para comparar la similitud entre caras en lugar de un clasificador. En las primeras etapas del desarrollo del proyecto, se realizaron algunas pruebas con este acercamiento y es una opción viable como solución alternativa, aunque presenta ciertos inconvenientes. Principalmente que requiere compara la cara del usuario con el resto de las caras registradas cada vez que se realiza la autenticación, lo que conlleva a tiempos elevados en casos donde existan muchos usuarios registrados.

Por otro lado, se podría ampliar el proyecto adaptando el código para escalar el sistema utilizando varios módulos front-end trabajando de manera concurrente. En el proyecto esta opción ya se consideró por lo que parte del código de la comunicación TCP fue ideada para funcionar con varios módulos.

Asimismo, otro posible avance sería la implementación de un programa complementario que facilite el mantenimiento de la base de datos y almacenamiento de imágenes en el computador. Así se simplificaría este proceso para el administrador. Una opción viable, es realizar en Python una interfaz con Tkinter que permitiese visualizar y navegar por las carpetas de las caras de usuarios y permitiendo acciones como eliminar o mover imágenes que no estén en el directorio correcto.

Por otro lado, y siguiendo con la idea de facilitar el mantenimiento de las imágenes se podría realizar un algoritmo que fuera recorriendo carpeta por carpeta detectando que imágenes difieren de la mayoría almacenada en el directorio. De esta forma se podrían eliminar falsos positivos en la detección de caras. También se desarrolló en el proyecto un script que realizaba esta tarea, pero no se terminó de aplicar debido a que el funcionamiento no era tan preciso.

Por último, es posible mejorar la robustez del sistema frente a la suplantación de identidad mediante técnicas basadas en el análisis de la textura, iluminación, visión en 3D. Dando como resultado un sistema más complejo, pero de mayor robustez.

10. REFERENCIAS BIBLIOGRÁFICAS

Bakshi, U., & Singhal, R. (2014). A survey on face detection methods and feature extraction techniques of face recognition. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET)*, 3(6), 2277-2285.

Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Computing Surveys (CSUR)*, 35(4), 399-458.

Wang, X., Peng, J., & Zhang, S. (2019). A survey of face recognition. *Journal of Image and Graphics*, 6(4), 234-244.

Shepley, A. J. (2018). Deep learning for face recognition: A critical analysis. *International Journal of Computer Vision*, 126(7), 700-720.

Diatronic (2024). Producto del catálogo. Recuperado de <https://diotronic.com/820-componentes-electr%C3%B3nicos>

Microlog (2024). Producto del catálogo. Recuperado de <https://microlog.es/2-inicio>

Yosinski, J. (2020). *Understanding and visualizing ResNets*. Towards Data Science. Recuperado de <https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8>

PyImageSearch. (2021, 19 de abril). *Face detection with dlib: HOG and CNN*. Recuperado el 3 de julio de 2024, de <https://pyimagesearch.com/2021/04/19/face-detection-with-dlib-hog-and-cnn/>

dlib. (s.f.). *Dlib C++ Library*. Recuperado el 3 de julio de 2024, de <http://dlib.net/>

Dobiasd. (s.f.). *Frugally Deep: Fast and simple deployment of Keras models*. GitHub. Recuperado el 3 de julio de 2024, de <https://github.com/Dobiasd/frugally-deep>

MySQL. (s.f.). *MySQL Connector/C++ Developer Guide*. Recuperado el 3 de julio de 2024, de <https://dev.mysql.com/doc/dev/connector-cpp/latest/>

Keras. (s.f.). *Keras documentation*. Recuperado el 5 de julio de 2024, de <https://keras.io/>

DigitalOcean. (2020, 19 de mayo). *How To Install MySQL on Ubuntu 20.04*. Recuperado el 3 de julio de 2024, de <https://www.digitalocean.com/community/tutorials/how-to-install-mysql-on-ubuntu-20-04-es>

W3Schools. (s.f.). *SQL Syntax*. Recuperado el 3 de julio de 2024, de https://www.w3schools.com/sql/sql_syntax.asp

11. ANEXOS

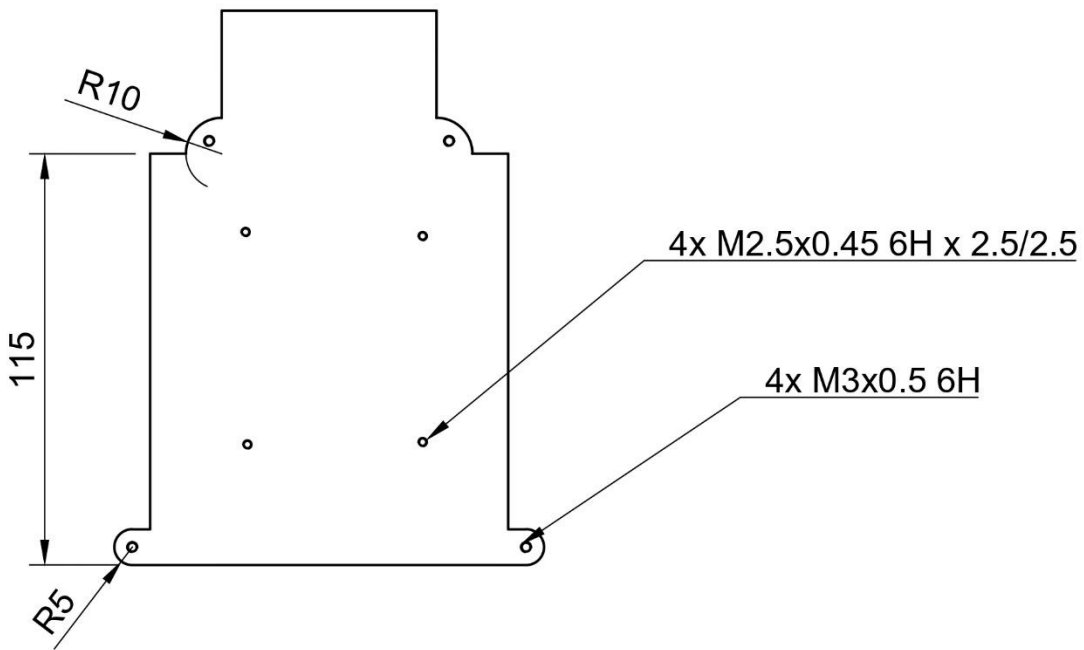
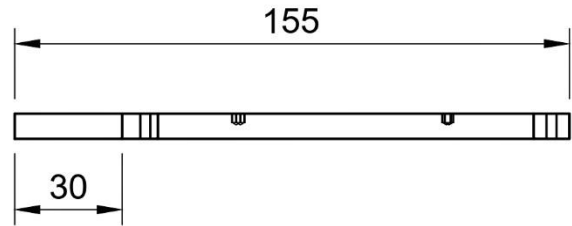
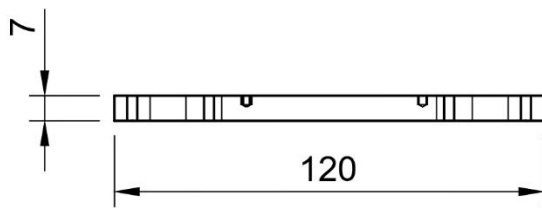
11.1 RELACIÓN DEL TRABAJO CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE DE LA AGENDA 2030.

Objetivos de Desarrollo Sostenible	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.	-	-	-	
ODS 2. Hambre cero.	-	-	-	
ODS 3. Salud y bienestar.	-	-	-	
ODS 4. Educación de calidad.	-	-	-	
ODS 5. Igualdad de género.	-	-	-	
ODS 6. Agua limpia y saneamiento.	-	-	-	
ODS 7. Energía asequible y no contaminante.	-	-	-	
ODS 8. Trabajo decente y crecimiento económico.	-	-	-	
ODS 9. Industria, innovación e infraestructuras.	-		-	-
ODS 10. Reducción de las desigualdades.	-	-	-	
ODS 11. Ciudades y comunidades sostenibles.	-	-	-	
ODS 12. Producción y consumo responsables.	-	-	-	-
ODS 13. Acción por el clima.	-	-	-	
ODS 14. Vida submarina.	-	-	-	
ODS 15. Vida de ecosistemas terrestres.	-	-	-	
ODS 16. Paz, justicia e instituciones sólidas.	-	-	-	
ODS 17. Alianzas para lograr objetivos.	-	-	-	

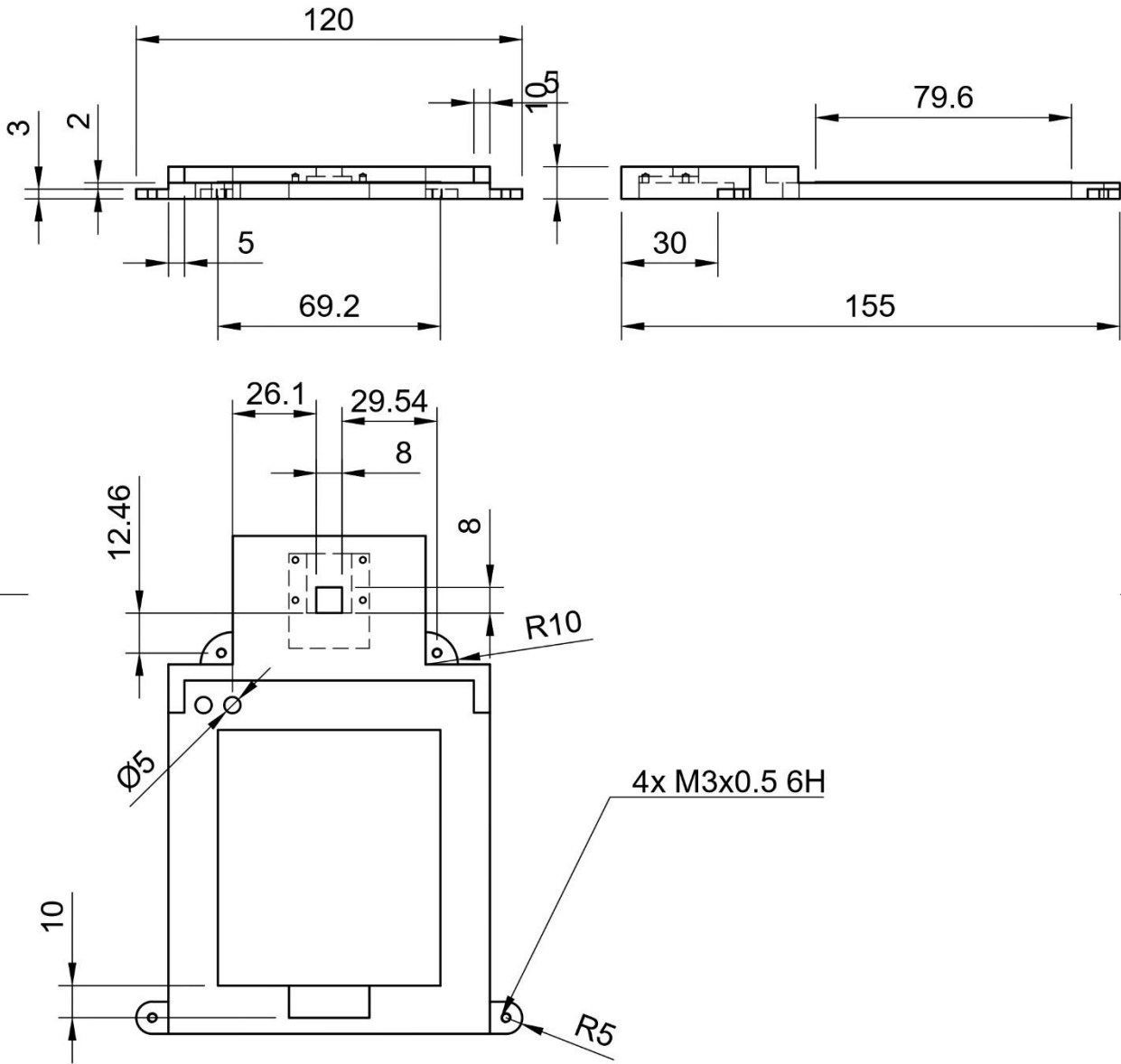
Se ha considerado que el proyecto cumple con el objetivo de desarrollo sostenible sobre la industria, innovación e infraestructuras, puesto que el dispositivo diseñado realiza la gestión de accesos a la maquinaria permitiendo una mejor organización.

11.2 PLANOS

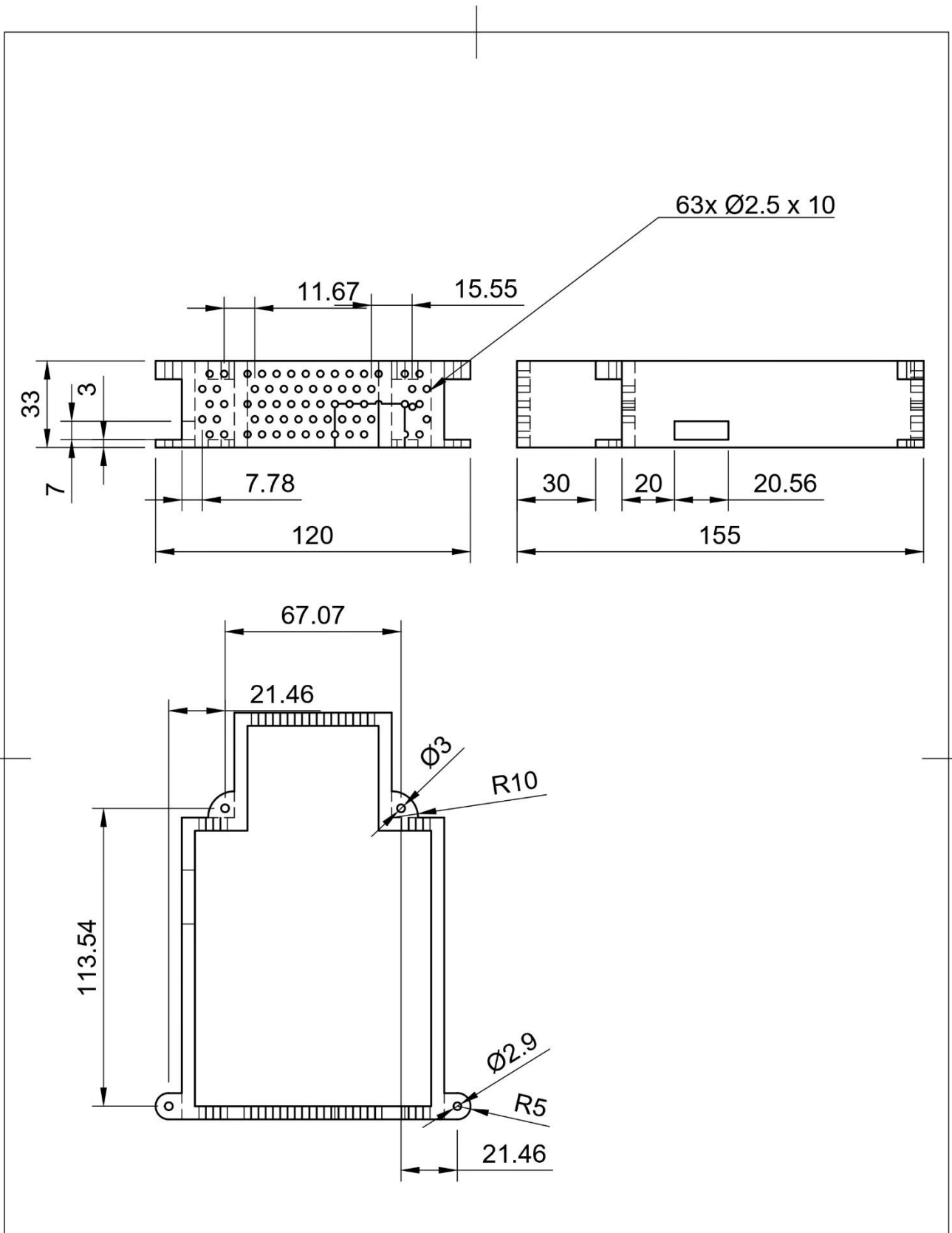
El encapsulado del dispositivo se ha realizado en Fusion360 para generar los archivos stl para imprimir la pieza. Estos se dividen en 3 piezas la base, la tapa y el lateral. La primera corresponde con el plano 1 y el así sucesivamente.



Dept.	Technical reference	Created by Marc Fontalba Roca 16/07/2024	Approved by	
		Document type Plano	Document status	
		Title Plano 01	DWG No. 01	
		Rev.	Date of issue	Sheet 1/1



Dept.	Technical reference	Created by Marc Fontalba Roca 16/07/2024	Approved by	
		Document type Plano	Document status	
		Title Plano 02	DWG No. 02	
		Rev.	Date of issue	Sheet 1/1



Dept.	Technical reference	Created by Marc Fontalba Roca 16/07/2024	Approved by	
		Document type Plano	Document status	
		Title Plano 03	DWG No. 03	
		Rev.	Date of issue	Sheet 1/1

11.3 PRESUPUESTOS

Se ha realizado el análisis del coste de producción mediante cuadros descompuestos.

COSTE DE PRDOCUCCIÓN					
1. Precios Unitarios					
Ref	Ud	Descripció	Precio/Ud.	Cantidad	Total
<u>Materiales</u>					
m1	ud.	Raspberry Pi 4 8GB	69.94 €	1	69.94 €
m2	ud.	RaspPi Camera Module 2 v2	24.14 €	1	24.14 €
m3	ud.	Teclado Matricial 4x4	5.99 €	1	5.99 €
m4	ud.	Diodo Led Rojo 5 mm diametro	0.07 €	1	0.07 €
m5	ud.	Diodo Led Verde 5 mm diametro	0.07 €	1	0.07 €
m6	ud.	Resistencia 5kΩ	0.05 €	1	0.05 €
m7	ud.	Resistencia 10kΩ	0.05 €	1	0.05 €
m8	ud.	PerfBoard	0.80 €	1	0.80 €
m10	ud.	Piezas CAD en PLA	15.33 €	1	15.33 €
m11	ud.	Cablecillos	0.07 €	11	0.78 €
m12	ud.	Tornillo M3	0.02 €	8	0.20 €
m13	ud.	Tornillos M2.5	0.02 €	8	0.20 €
m14	ud.	Micro SD 32 GB	13.99 €	1	13.99 €
<u>Subtotal del Material</u>					
131.61 €					

<u>M.O.D.</u>						
h1	h	Oficial 1ª mecánica	20.00 € (€/h)			
h2	h	Ayudante mecánica	15.90 € (€/h)			
h1	h	Oficial 1ª electronica	20.00 € (€/h)			
h2	h	Ayudante electronica	15.90 € (€/h)			
<u>Secciones</u>						
s1	ch	Montaje	25.20 € (€/h)			
s2	ch	Soldadura	20.40 € (€/h)			
s3	ch	Adhesión	22.20 € (€/h)			
s4	ch	Control de calidad	12.00 € (€/h)			
2. Costes de producción						
Ref	Ud	Descripción	Precio	Cantidad	Parcial	
d1	ud.	Leds, resistencias , cablecillos y placa perforada. Correctamente soldados.				
<u>Materiales</u>						
m4	ud.	Diodo Led Rojo 5 mm diamtero	0.07 €	1	0.07 €	
m5	ud.	Diodo Led Verde 5 mm diamtero	0.07 €	1	0.07 €	
m6	ud.	Resistencia 5kΩ	0.05 €	1	0.05 €	

m7	ud.	Resistencia 10kΩ	0.05€	1	0.05€	
m8	ud.	PerfBoard	0.80€	1	0.80€	
m11	ud.	Cablecillos	0.07€	3	0.21€	
Secciones						
s2	ch	Soldadura	20.40€	0.08	1.70€	
M.O.D.						
h1	h	Oficial 1ª electronica	20.00€	0.08	1.67€	
h2	h	Ayudante electronica	15.90€	0.08	1.33€	
				TOTAL	5.94€	
3. Cuadro de precios Descomp						
D1	ud.	Dispositivo correctame nte ensamblad o , con la raspberry pi, el modulo de leds, la camara, el teclado y las piezas CAD				
 Materiales						
d1	ud.	Leds, resistencias , cablecillos y placa perforada. Correctame nte soldados.	5.94€	1	5.94€	

m1	ud.	Raspberry Pi 4 8GB	69.94 €	1	69.94 €
m2	ud.	RaspPi Camera Module 2 v2	24.14 €	1	24.14 €
m3	ud.	Teclado Matricial 4x4	5.99 €	1	5.99 €
m10	ud.	Piezas CAD en PLA	15.33 €	1	15.33 €
m11	ud.	Cablecillos	0.07 €	8	0.57 €
m12	ud.	Tornillo M3	0.02 €	8	0.20 €
m13	ud.	Tornillos M2.5	0.02 €	8	0.20 €
m14	ud.	Micro SD 32 GB	13.99 €	1	13.99 €
Secciones					
s1	ch	Montaje	25.20 €	0.25	6.30 €
s3	ch	Adhesión	22.20 €	0.08	1.85 €
s4	ch	Control de calidad	12.00 €	0.25	3.00 €
M.O.D.					
h1	h	Oficial 1ª mecánica	20.00 €	0.58	11.67 €
h2	h	Ayudante mecánica	15.90 €	0.58	9.28 €
CD02					
%	Porcentaje	Costes Directos	0.02		3.37 €
TOTAL POR UNIDAD					171.76 €

11.4 CODIGOS

Los códigos se pueden encontrar en el siguiente repositorio de Github:

<https://github.com/Mswp0/Embedded-face-recognition-system>