Prototipo De Plataforma IoT Para El Control De Acceso Al Hogar Mediante El Uso De Una Cerradura Electrónica Con Doble Autenticación: Reconocimiento Facial y Teclado Numérico.

Sebastián Contreras Ceballos y Hernando José Rojas Castro

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería de Sistemas e Informática

Bucaramanga

2023

Prototipo De Plataforma IoT Para El Control De Acceso Al Hogar Mediante El Uso De Una Cerradura Electrónica Con Doble Autenticación: Reconocimiento Facial y Teclado Numérico.

Sebastián Contreras Ceballos y Hernando José Rojas Castro

Director: Jathinson Meneses Mendoza

Msc. En Gestión, Aplicación y Desarrollo de Software.

Codirector (es): Henry Andres Jimenez Herrera

Msc. en Ingeniería de Sistemas e Informática.

Gabriel Rodrigo Pedraza Ferreira

PhD. en Ciencias de la Computación.

Trabajo de grado para optar por el título de Ingeniero de Sistemas

Universidad Industrial de Santander

Facultad de Ingenierías Fisicomecánicas

Escuela de Ingeniería de Sistemas e Informática

Bucaramanga

2023

Dedicatoria

Dedicado a:

Mi madre Yriana Judith Ceballos Borja, que me apoyo a seguir lo que me gusta desde la infancia y fue mi soporte y estabilidad durante todo el transcurso de mi carrera brindándome todo lo que necesitaba. A mi perrita Wanda que a pesar de no estar conmigo, me acompaño en mis momentos más difíciles y me ayudo a superar todo y seguir adelante.

Y por último a mis amigos y familiares que estuvieron conmigo durante todo el transcurso de mi carrera, por el apoyo incondicional que me brindaron y sobre todo las dosis de alegría y risa que me hicieron disfrutar mi trayecto universitario.

Sebastian Contreras Ceballos.

Dedicatoria

Dedicado a:

Especialmente a mi primer amor y mi primer gran héroe, mi madre y padre, Glenys Maria Castro Mendez y Luis José Rojas Trillos, luchadores inagotables que con el ejemplo me enseñaron lo que es la constancia, perseverancia y amor en estos más de 5 años de apoyo incondicional en mi trayectoria académica, por la motivación en los momentos de subidas y bajadas, por permitirme alcanzar este logro, este sueño.

Al amor de mi vida, Camila Cabellero, que alegra mis días y con su ejemplo de perseverancia me motiva a continuar y darlo todo, quien siempre me apoyó, confió en mi y me motivó a seguir adelante, que este sea el primero de muchos logros que compartiremos juntos.

Agradecer también a mis amigos, Camilo, Brayan, Nicolas, Gino, Angel, Stiven, Oviedo, Karen, quienes de alguna manera influyeron positivamente en este largo y tedioso camino, sin ellos todo hubiera sido más difícil.

Finalmente, a Filipo, el consentido de la casa y la mejor compañía en esas noches de trasnocho, sacándome una sonrisa.

Hernando José Rojas Castro.

Agradecimientos

A nuestro director y codirector, Jathinson Menezes Mendoza y Henry Andres Jimenez Herrera por su valioso tiempo, colaboración y dedicación en el desarrollo del presente proyecto.

A nuestra Alma máter, la Universidad Industrial de Santander, la escuela de ingeniería de sistemas e informática y a cada uno de los docentes que aportaron en nuestro desarrollo profesional y personal.

A nuestros familiares anteriormente mencionados, por su comprensión en esas horas de ausencia por el cumplimiento de los deberes.

A nuestros compañeros y amigos no mencionados que aportaron un granito de arena en nuestro desarrollo personal y profesional.

Contenido

Lista de tablas

Lista de figuras

Resumen

Título: Prototipo de plataforma IoT para el control de acceso al hogar mediante el uso de una cerradura electrónica con doble autenticación: reconocimiento facial y teclado numérico.*

Autores: Hernando Jose Rojas Castro, Sebastian Contreras Ceballos.**

Palabras clave: Aplicación, Control, IoT, Plataforma Web, Reconocimiento facial.

Descripción:

Un informe de hurto a viviendas deja en evidencia de que en lo corrido del año 2022 hasta el mes de agosto se habían registrado un total de 16056 hurtos a viviendas, de los cuales 577 fueron realizados con una llave maestra y 9690 fueron realizados sin empleo de arma (Policía Nacional de Colombia, 2022). La seguridad y la privacidad son aspectos importantes y fundamentales en la vida cotidiana, sobre todo cuando se refiere al hogar, cuando se habla del ingreso a la vivienda, las personas esperan contar con la confianza de salir de sus hogares y no ser víctimas de un robo, ya que allí se encuentran sus hijos, mascotas, o cualquier tipo de objetos de valor, estos hurtos pueden ser por diferentes circunstancias, por lo que es necesario crear herramientas eficientes y eficaces en el control de acceso al hogar, que brinden confianza y aprovechen mejor la tecnología de la cual disponemos actualmente.

---

* Proyecto de Grado. Trabajo de Investigación.

** Facultad de Ingenierías Fisicomecánicas. Escuela de Ingeniería de Sistemas e Informática.

Director: Msc. Jathinson Meneses Mendoza

Codirector: Msc. Henry Andres Jimenez Herrera

PhD. Gabriel Rodrigo Pedraza Ferreira

Abstract

Title: Prototype of an IoT platform for home access control using a dual-authentication electronic lock: facial recognition and numeric keypad.

Authors: Hernando Jose Rojas Castro, Sebastian Contreras Ceballos.

Keywords: Application, Control, IoT, IoT Platform, Facial Recognition, Web Platform.

Description:

A report on home theft reveals that during the year 2022, until the month of August, a total of 16,056 home thefts had been registered, of which 577 were carried out using a master key and 9,690 were committed without the use of a weapon (Colombian National Police, 2022). Security and privacy are important and fundamental aspects in everyday life, especially when it comes to one's home. When it comes to entering the residence, people expect to have the confidence to leave their homes without becoming victims of theft, as their children, pets, or valuable possessions are located there. These thefts can occur under different circumstances, so it is necessary to create efficient and effective tools for home access control that provide confidence and make the best use of the technology currently available to us.

---

* Degree Project. Research Work.

** Faculty of Physical-Mechanical Engineering. School of Systems and Computer Engineering.

Director: Msc. Jathinson Meneses Mendoza

Co-Director: Msc. Henry Andres Jimenez Herrera

PhD. Gabriel Rodrigo Pedraza Ferreira

Introducción

La seguridad y privacidad son una de las principales preocupaciones individuales, las personas buscan sentirse seguras y confiadas de que un intruso que intente ingresar a su hogar no lo logre, en Colombia tan solo en la primera mitad del año 2022 se habían presentado más de 15.000 hurtos a viviendas (Policía Nacional de Colombia, 2022), por otra parte, la pérdida de las llaves con la que se abre la puerta de la vivienda puede indicar una vulnerabilidad de seguridad de la misma en caso de caer en manos equivocadas ya sea por pérdida o robo, cosa que se podría evitar si no se usará llaves físicas.

En el mercado se puede encontrar principalmente ‘cerrojos electrónicos’ los cuales cuentan con un teclado numérico para ingresar la clave de acceso, también se pueden encontrar numerosos cerrojos con autenticación por huella dactilar, sin embargo, son escasas las ofertas de cerraduras que brinden la opción con doble autenticación como el de reconocimiento facial y el teclado numérico y, además, que integren estas funcionalidades con una herramienta software de control de acceso desde la cual se pueda hacer una gestión oportuna de usuarios y recursos.

El proyecto propone el diseño de una arquitectura de sistema IoT e implementación de un prototipo funcional con una plataforma web haciendo uso de componentes electrónicos y tecnologías de alto nivel que faciliten la implementación de herramientas funcionales y seguras como se espera que sea una cerradura electrónica, la cual tiene dos medios de autenticación y está conectada a internet para intercambio de datos con la plataforma software que podrá almacenar información valiosa, como la cantidad de veces que un usuario (habitante del hogar) ingresa a la casa, el número total, elementos gráficos acerca de los ingresos al hogar y administrar usuarios desde esta, la finalidad es ofrecer una herramienta confiable cuyo uso brinde un valor agregado y

aporte en la modernización de los hogares ayudando a la gestión de usuarios en cuanto al control de acceso se refiere.

El desarrollo del proyecto contiene un marco referencial y conceptual con el fin de definir conceptos, tecnologías y componentes que son muy utilizados durante el desarrollo y un estado del arte con proyectos académicos y productos del mercado que hacen uso de herramientas tecnológicas para el control de acceso al hogar.

La metodología implementada en el desarrollo del trabajo se dividió en 4 fases, inicialmente se hizo una búsqueda de tecnologías y componentes con el fin de realizar una lista de potenciales tecnologías a usar para después realizar una comparación entre ellas y elegir las que mejor se adapten a lo que se espera diseñar. En la segunda fase se realizó el levantamiento de requerimientos y análisis de estos, primero se identificaron las necesidades de un sistema de estas características para posteriormente realizar el diseño del sistema basados en las tecnologías seleccionadas previamente. En la tercera fase se inicia el desarrollo del prototipo funcional basados en el diseño presentado anteriormente, en esta fase se desarrolla la plataforma web, se implementa el modelo de reconocimiento facial, se construye la arquitectura hardware y posteriormente se integran estas partes para conformar el prototipo. Sin embargo, el prototipo inicial no pudo ser implementado por completo debido a problemas en la implementación, por lo que se realizó una nueva fase de prototipado, volviendo a hacer el diseño e implementación del prototipo funcional final. Finalmente, en la cuarta fase se realiza la evaluación y validación del prototipo funcional, esto se hizo mediante un caso de uso donde se ejecutaron pruebas haciendo uso del reconocimiento facial y evaluando el comportamiento del sistema respecto al control del acceso.

Una vez finalizado el trabajo de investigación, se procedió a evaluar y validar el prototipo. Se llevaron a cabo pruebas con cuatro escenarios en el reconocimiento facial y se registraron los resultados obtenidos basados en los requerimientos definidos inicialmente. Además, se realizaron observaciones y consideraciones basadas en lo descrito en el informe de resultados.

1. Planteamiento del problema

Un reporte de hurto a viviendas realizado por la policía nacional deja en evidencia de que en lo corrido del año 2022 hasta el mes de agosto se habían registrado un total de 16056 hurtos a viviendas, de los cuales 577 fueron realizados con una llave maestra, 72 no reportaron el medio o no se conoce y 9690 fueron realizados sin empleo de arma (Policía Nacional de Colombia, 2022), lo que podría ser por varias circunstancias, entre las cuales no se puede descartar que haya sido por pérdida o robo de las llaves de la vivienda.

La seguridad y la privacidad son aspectos importantes y fundamentales en la vida cotidiana, sobre todo cuando se refiere al hogar, cuando se habla del ingreso a la vivienda, las personas esperan contar con la confianza de salir de sus hogares y no ser víctimas de un robo, ya que allí se encuentran sus hijos, mascotas, o cualquier tipo de objetos de valor, una de las causas de estos hurtos puede ser la pérdida o robo de las pertenencias, entre ellas las llaves físicas, por lo que es necesario crear herramientas efectivas en el control de acceso al hogar, que brinden confianza y sin tener la preocupación de no dejar las llaves por dentro o extraviarse ya que el medio de ingreso a su hogar no depende de estas.

En el estado del arte hay una variedad de estudios enfocados en el desarrollo de herramientas tecnológicas para el control de acceso al hogar y otros sitios, es el caso del proyecto realizado por Moreno, J (2016) que desarrolla un prototipo de cerradura electrónica inteligente con reconocimiento facial con uso de componentes electrónicos, sin embargo, es poca la cantidad de estudios o soluciones que propongan una opción de desbloqueo remoto de la cerradura y que además brinden información gráfica del número de ingresos al hogar.

Con el crecimiento exponencial en la utilización e implementación de dispositivos electrónicos en los hogares, impulsado por la evolución constante y el desarrollo de la tecnología, surgen conceptos nuevos como las 'ciudades inteligentes' o 'casas inteligentes'. Esta tendencia brinda una oportunidad significativa para la introducción de tecnologías basadas en el 'Internet de las cosas' (IoT). En este contexto, surge también la posibilidad de reemplazar las cerraduras convencionales por dispositivos electrónicos, lo cual abre nuevas posibilidades de uso dentro del hogar. Un ejemplo de ello es una cerradura electrónica que, además de controlar el acceso, incorpore un algoritmo de reconocimiento facial respaldado por inteligencia artificial haciendo uso de modelos pre entrenados y con estudios de precisión que soporten la fiabilidad de estos modelos. Esta combinación podría proporcionar una herramienta más eficaz y segura que al integrarse con un software de gestión y administración de usuarios, podría ofrecer confiabilidad y modernización en los hogares.

2. Objetivos

2.1. Objetivo General

Desarrollar un prototipo de plataforma IoT para el control de acceso al hogar mediante el uso de una cerradura electrónica con reconocimiento facial y teclado numérico.

2.2. Objetivos Específicos

Analizar las necesidades e identificar los requerimientos para un sistema de control de acceso al hogar.

Diseñar una arquitectura de sistema como solución a los requerimientos identificados

Evaluar el funcionamiento de la plataforma IoT desarrollada mediante la interacción de los módulos realizados con un prototipo funcional.

3. Marco referencial

3.1. Marco Conceptual

3.1.1. Reconocimiento facial

"Un algoritmo codifica automáticamente la imagen facial (sobre la que se quiere indagar) introducida en el sistema, para luego compararla con los perfiles almacenados en él. De este modo se obtiene una lista de "candidatos" de los aciertos más probables." (INTERPOL, 2021, p. 6).

Para efectos de una mayor comprensión se toma como referencia dos librerías de reconocimiento con modelos pre entrenados con el fin de profundizar en el funcionamiento del reconocimiento facial.

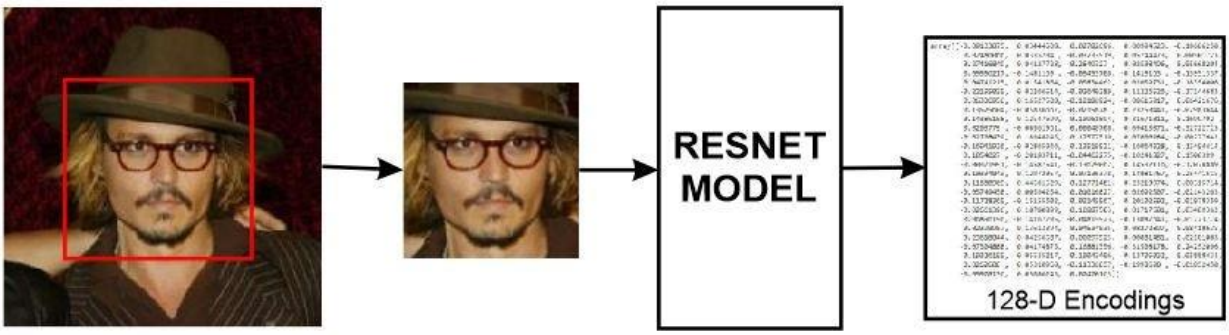
3.1.1.1. Face Recognition

Profundizando en el funcionamiento de la librería de reconocimiento facial de OpenCV 'face recognition':

1. Inicialmente se procesa una o varias imágenes obtenidas de una base de datos capturando los rostros, luego estos se codificarán en un vector de características de 128-d (128 de dimensión), cada rostro tendrá un vector diferente y será guardado en una lista de Python (Ver figura 1).

Figura 1

Codificación de imagen en vector de 128-d



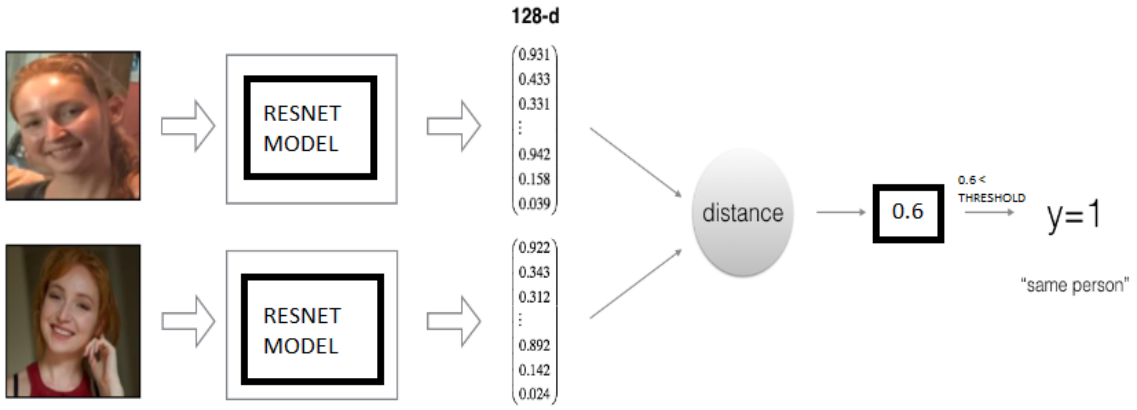
Nota. Explicación gráfica de la codificación de imagen en vector de 128-d. Tomado de Step by Step Face Recognition Code Implementation From Scratch In Python, por R Thakur, 2020. https://towardsdatascience.com/step-by-step-face-recognition-code-implementation-from-scratch-in-python-cc95fa041120

2. Se hace un video streaming con la cámara, en la cual se debe colocar como objetivo el rostro de la persona al frente de la cámara. Se aplicará el proceso anterior solo que esta vez para cada fotograma del video streaming.
3. Finalmente se comparan las primeras codificaciones que se habían hecho de los rostros existentes de la base de datos con las codificaciones del video streaming.

   Dados estos dos datos internamente se aplica la distancia euclidiana entre ellos y si el resultado este bajo cierta tolerancia (por lo general el valor de tolerancia es de 0.6, y entre más bajo sea el valor más estricto será la comparación de rostros) quiere decir que hemos obtenido un emparejamiento, es decir que se ha determinado que ambos rostros pertenecen a la misma persona (Ver figura 2).

Figura 2

Comparación de rostros con los vectores de 128-d



Nota. Explicación gráfica del procesamiento interno del modelo. Tomado y modificado de Face Recognition for the Happy House, por A. Patel, 2019. https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/Deep%20Learning%20Notebooks%20by%20Andrew%20NG/Convolutional%20Neural%20Networks/Week4/Face%20Recognition/Face%20Recognition%20for%20the%20Happy%20House.ipynb

El funcionamiento del Deep learning con el reconocimiento facial hace uso de la técnica Deep Metric Learning.

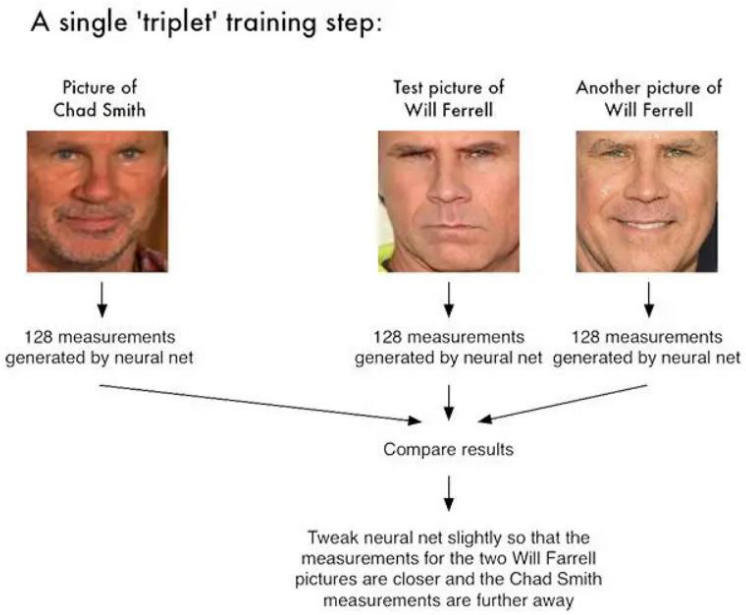
Generalmente cuando se utiliza Deep Learning se entrena una red neural para:

1. Aceptar una imagen de entrada.
2. Y generar una clasificación/etiqueta para esa imagen.
3. Sin embargo, el Deep Metric Learning es diferente.
4. En lugar de intentar generar una sola etiqueta, estamos generando un vector de características de valor real.

Para la red de reconocimiento facial dlib, el vector de características de salida es 128-d (es decir, una lista de 128 números con valores reales) que se utiliza para cuantificar el rostro. El entrenamiento de la red se realiza mediante tripletes:

Figura 3

Triple en Deep Learning



Nota. Tomado de Face recognition with OpenCV, Python, and Deep Learning, por Pyimagesearch, por A. Rosebrock, 2018. https://pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/

Aquí se proporcionan tres imágenes a la red.

Dos de estas imágenes son ejemplos faciales de la misma persona.

La tercera imagen muestra un rostro aleatorio de un conjunto de datos y no corresponde a la misma persona que las otras dos imágenes.

La red cuantifica las caras, generando una representación embebida (cuantificación) de 128-d para cada una.

A partir de ahí, la idea principal es ajustar los pesos de la red neuronal para que las medidas de 128-d de las dos imágenes del mismo rostro estén más cercanas entre sí y más distantes de las medidas del rostro que es diferente.

La arquitectura de la red que se va a utilizar para el reconocimiento facial se basa en la ResNet-3 del estudio Deep Residual Learning for Image Recognition de He et al., pero con menos capas y el número de filtros reducido a la mitad.

La red en sí fue entrenada por Davis King (el creador de dlib) en un conjunto de datos de ≈3 millones de imágenes. En el conjunto de datos Labeled Faces in the Wild (LFW), la red se compara con otros métodos de última generación y alcanza una precisión del 99,38% (Rosebrock, 2018).

3.1.1.2. Face API JS

La librería face-api.js es una potente herramienta de procesamiento de imágenes en el navegador web que utiliza modelos de aprendizaje automático para realizar tareas relacionadas con el procesamiento facial y detección de rostros. Face-api.js está basada en TensorFlow.js, una biblioteca de JavaScript para ejecutar modelos de aprendizaje automático en el navegador.

La funcionalidad principal de face-api.js se puede dividir en dos áreas principales: detección de rostros y reconocimiento facial.

Detección de rostros:

La biblioteca face-api.js utiliza un modelo preentrenado llamado "ssdMobilenetv1" para detectar rostros en imágenes y videos. Este modelo se basa en una red neuronal llamada "Single Shot MultiBox Detector" (SSD) y se entrena para reconocer la presencia y ubicación de rostros en una imagen. Al analizar diferentes regiones de la imagen, el modelo puede identificar y delimitar los rostros detectados.

El reconocimiento facial funciona de la siguiente forma:

Face-api.js permite el reconocimiento facial, que consiste en identificar y comparar características faciales para reconocer personas específicas. La biblioteca utiliza modelos preentrenados como "faceLandmark68Net" y "faceRecognitionNet" para lograr esto.

El modelo "faceLandmark68Net" se utiliza para detectar 68 puntos de referencia faciales en un rostro, como los ojos, la nariz, los labios, etc. Estos puntos de referencia se utilizan para calcular características específicas del rostro, como la forma y la posición de los ojos y la boca.

El modelo "faceRecognitionNet" se entrena para generar una representación numérica única y fija para cada rostro detectado. Estas representaciones, llamadas "incrustaciones faciales" (face embeddings), capturan las características distintivas del rostro y se pueden utilizar para comparar y reconocer rostros en diferentes imágenes. Al comparar las incrustaciones faciales de diferentes rostros, se puede determinar si pertenecen a la misma persona o no.

El proceso del reconocimiento facial se puede dividir en tres pasos, y es muy similar al funcionamiento de la librería face recognition:

1. Inicialmente se procesan los rostros obtenidos desde la base de datos, para ello se aplicará la detección facial de las imágenes para poder encontrar la ubicación de los

rostros. Luego de haber obtenido la ubicación de los rostros se codificarán en un vector de características de 128-d (128 de dimensión), cada rostro tendrá un vector diferente y será guardado en una Float32Vector.

2. Se hace un video streaming con la cámara, en la cual se debe colocar como objetivo el rostro de la persona al frente de la cámara. Se aplicará el proceso anterior en el momento en el que se tome la foto del usuario.
3. Finalmente se comparan las codificaciones que se habían hecho de los rostros existentes de la base de datos con las codificaciones del video streaming. Dados estos dos datos internamente se aplica la distancia euclidiana entre ellos y si el resultado este bajo cierta tolerancia (por lo general el valor de tolerancia es de 0.6, y entre más bajo sea el valor más estricto será la comparación de rostros) quiere decir que hemos obtenido un emparejamiento, es decir que se ha determinado que ambos rostros pertenecen a la misma persona.

3.1.2. Computadora móvil

"Una computadora con características móviles es aquella con similar poder de cómputo que una computadora de escritorio pero que por sus dimensiones se puede trasladar fácilmente de un lugar a otro." (Universidad virtual del tecnológico de Monterrey, 2011).

Para el contexto del proyecto, computadora móvil se refiere a el dispositivo con capacidad de procesamiento requerido para las operaciones de reconocimiento facial y lógicas de la cerradura.

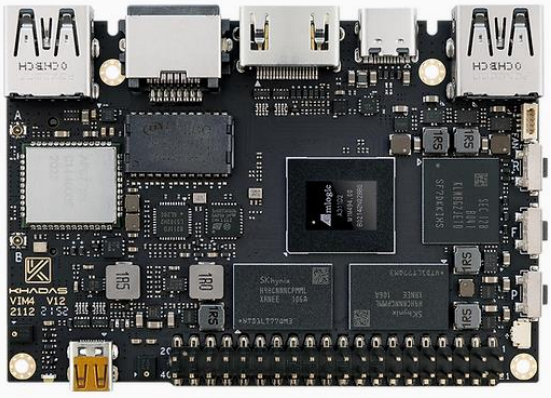
Se toman como referencia cuatro computadoras móviles con el fin de analizar las características y especificaciones técnicas de cada una.

3.1.2.1. Khadas VIM4.

Khadas VIM4, tiene un procesador Cortex-A73 de cuatro núcleos operando a 2.2GHz, memoria RAM LPDDR4X de 8GB y una memoria flash eMMC de 32GB, una GPU ARM Mali-G52 MP8(8EE) a una frecuencia de hasta 800MHz, compatible con conexión Ethernet y entrada para almacenamiento SSD M.2, compatible con bluetooth y 40 pines para conexiones de entrada y salida, entre otros.

Figura 4

Khadas VIM4



Nota. Dispositivo móvil Khadas VIM4, tomado de Khadas, 2022. https://www.khadas.com/vim4

3.1.2.2. Orange Pi 3 LTS

Orange Pi 3 LTS, tiene un procesador Cortex-A53 de cuatro núcleos operando a 1.8GHz, memoria SDRAM LPPDR3 de 2GB y una memoria flash eMMC de 8GB, una GPU Mali T720 multicore, compatible con conexión Ethernet y entrada para almacenamiento MicroSD, compatible con bluetooth y 26 pines para conexiones de entrada y salida, entre otros.

Figura 5

Orange Pi 3 LTS

Nota. Dispositivo móvil Orange Pi 3 LTS., tomado de Orange Pi. http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/details/orange-pi-3-LTS.html

3.1.2.3. Raspberry Pi 4

Raspberry Pi 4, tiene un procesador Cortex-A72 de cuatro núcleos de 64 bits operando a 1.8GHz, memoria SDRAM LPDDR4-3200 de 1, 2, 4 o 8GB dependiendo del modelo de raspberry, una GPU Video Core VI integrada con la CPU, compatible con bluetooth y 40 pines para conexiones de entradas y salidas adicionales, entre otros.

Figura 6

Raspberry Pi4

Nota. Dispositivo móvil Raspberry Pi 4, tomado de Raspberry Pi. https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

3.1.2.4. Jetson Nano 2 GB

Jetson Nano, tiene un procesador ARM A57 de cuatro núcleos operando a 1.43GHz, memoria RAM LPDDR4 de 64 bits y 2GB, GPU Maxwell de 128 núcleos, compatible con conexión Ethernet y entrada para almacenamiento MicroSD, entradas USB y HDMI, cuenta con 40 pines para conexiones de entrada y salida, entre otros.

Figura 7

Jetson Nano 2GB

Nota. Dispositivo móvil Jetson Nano 2GB, tomado de Nvidia - Kit para desarrolladores Jetson Nano, 2023. https://www.nvidia.com/es-la/autonomous-machines/embedded-systems/jetson-nano/education-projects/

3.1.3. Microcontrolador

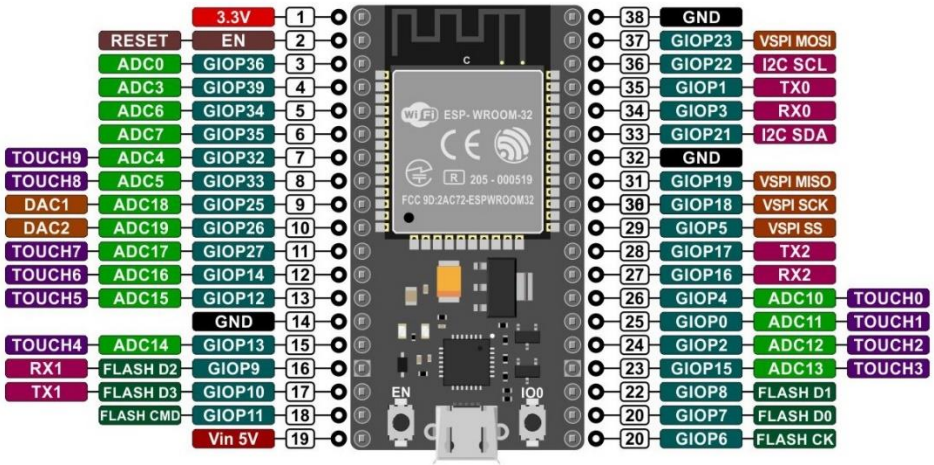
"Un microcontrolador es un dispositivo electrónico capaz de llevar a cabo procesos lógicos. Estos procesos o acciones son programados en lenguaje ensamblador por el usuario, y son introducidos en este a través de un programador" (Araguayo, 2004, p. 3)

3.1.3.1. ESP32

Un ejemplo de microcontrolador es la ESP32, la cual cuenta con un microprocesador de 32 bits Xtensa LX6 de doble núcleo operando a 160 MHz y rindiendo hasta 600 DMIPS, memoria de 520 KiB SRAM, conectividad inalámbrica por bluetooth o Wifi y 38 pines para entradas y salidas.

Figura 8

I/O ESP32



Nota. Diagrama de pines de entrada y salida de microcontrolador ESP32, tomado de Vasanza. Especificaciones del módulo ESP32, https://vasanza.blogspot.com/2021/07/especificaciones-del-modulo-esp32.html

3.1.4. Cerradura electrónica

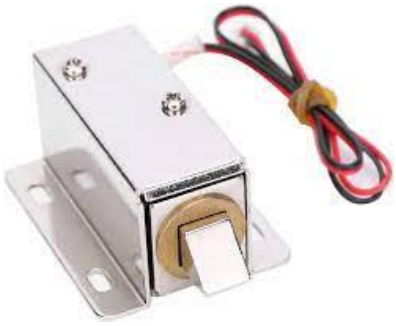
"Las cerraduras electrónicas son una evolución de las cerraduras tradicionales, que aprovechándose de la corriente eléctrica y/o de una batería propia, permite cerrar y abrir una puerta de forma mucho más segura y cómoda" (Soria Saez, 2021).

3.1.4.1. Cerradura Solenoide

Un ejemplo de cerradura electrónica es la cerradura solenoide, la cual cumple con las características descritas en la definición anterior y puede será de mucha utilidad en el desarrollo del proyecto.

Figura 9

Cerradura Solenoide



Nota. Tomado de eBay – Solenoide de tipo abierto, https://www.ebay.es/itm/185803204153

3.1.5. Aplicación Web

Una aplicación web se puede definir como una aplicación en la cual el usuario por medio del navegador realiza peticiones a una aplicación remota accesible a través de internet (o a través de intranet) y que recibe una respuesta que se muestra en el propio navegador (Luján S, 2002, p. 2).

El desarrollo web moderno consta de dos subdivisiones en su estructura o forma de desarrollo llamados ‘backend’ y el ‘frontend’.

3.1.5.1. Frontend

“Frontend se encarga de estilizar la página de tal manera que la misma pueda presentar la información de forma agradable para el usuario” (Alvarado citado por Pérez et al, 2021, p. 374-348).

Como lenguajes para la estructura y estilos de la interfaz gráfica desde el navegador están HTML y CSS, ya que HTML es el lenguaje que le da la estructura a las páginas web y por otra parte es CSS quien le da el estilo.

Además, está el componente interactivo donde aparece el lenguaje JavaScript con una serie de librerías que facilitan en mayor medida el desarrollo de las aplicaciones web, entre ellas están:

3.1.5.1.1. React

React es un framework desarrollado por Facebook y se utiliza para construir interfaces de usuario interactivas y reactivas.

3.1.5.1.2. Vue.js

Por su parte Vue.js es un framework que también se utiliza para construir interfaces interactivas que combina algunas cualidades de flexibilidad y simplicidad de React y el enfoque basado en componentes de Angular.

3.1.5.1.3. Angular

Angular es un framework basado en el modelo single page application por sus siglas SPA el cual funciona por componentes haciendo innecesaria la carga de una página por completo cada vez que se realizan peticiones, por el contrario, estos procesos se hacen de manera asíncrona desde la misma página web.

3.1.5.2. Backend

"Se denomina backend a la capa de acceso a los datos de un software que no es accesible para el usuario final. Además, esta capa contiene toda la lógica de la aplicación que maneja los datos" (Chapaval citado por Pérez et al, 2021, p. 348).

Para la capa lógica de la aplicación existe una variedad de lenguajes de programación, se toma como referencia cuatro lenguajes con el fin de conocer cada uno de ellos.

3.1.5.2.1. Javascript

Javascript se presenta como un lenguaje de desarrollo de aplicaciones cliente/servidor a través de Internet.

El programa en JavaScript tiene la particularidad de que esta insertado dentro del mismo del documento HTML que lo presenta al usuario y no es por ellos un programa aparte. Permite crear aplicaciones similares a los CGI (Common Gateway Interface). El CGI es un mecanismo que se ha utilizado en los servidores Web para implementar las páginas Web activas. El

funcionamiento de los CGI es el siguiente: Lee los datos provenientes de un formulario de una página web, procesa la información y lo escribe sobre el canal de salida estándar que es la pantalla del ordenador. (Maza, 2012, p. 9)

3.1.5.2.2. Java

Java es un lenguaje de programación orientado a objetos el cual es uno de los tres más utilizados en la actualidad. Gracias al amplio soporte con el que cuenta, así como también con la gran variedad de clases y colecciones que tiene; Java es uno de los lenguajes más robustos y utilizados en el mundo del desarrollo de software multiplataforma.

Lo anterior debido a que permite al desarrollador el crear aplicaciones o sistemas de información locales, aplicaciones o sistemas dentro de ambientes web e incluso aplicaciones para móviles. Con lo cual se consolida como uno de los mejores lenguajes de programación en la actualidad y en un futuro quizá el lenguaje de programación universal. (Moreno, s.f.)

3.1.5.2.3. Python

Python es un lenguaje de programación de alto nivel, interpretado y multipropósito. En los últimos años su utilización ha ido constantemente creciendo y en la actualidad es uno de los lenguajes de programación más empleados para el desarrollo de software.

Python puede ser utilizado en diversas plataformas y sistemas operativos, entre los que podemos destacar lo más populares, como Windows, MAC OS y Linux. Pero, además, Python también puede funcionar en smartphones. (Montoro, 2013, “¿Qué es Python?”, párrafo 1)

3.1.5.2.4. PHP

El lenguaje PHP (PHP Hypertext Pre-Processor) es uno de los más antiguos (fue creado en 1995 por la empresa PHP Group) y utilizado en el diseño de páginas web que utilizan bases de

datos.Se trata de un lenguaje interpretado en el lado del servidor que permite la creación de páginas web dinámicas que pueden estar dentro de páginas en HTML. Es uno de los lenguajes de programación web más populares por su rapidez y la facilidad de desarrollo (Perez J. L citado por Mina y Cedeño, 2018, p. 32).

3.1.6. Bases de Datos

Una base de datos es un conjunto de datos almacenados en memoria externa que están organizados mediante una estructura de datos. Cada base de datos ha sido diseñada para satisfacer los requisitos de información de una empresa u otro tipo de organización. (Marqués, 2011, p. 2).

Además, se espera que la base de datos sea confiable, consistente y fácilmente utilizable para acceder de forma rápida y precisa a la información, también se espera que sea flexible y susceptible a posibles fallos o inconvenientes que se puedan presentar en los procesos del sistema.

Se realiza una búsqueda de las distintas bases de datos que existen, donde se puede observar que hay bases de datos relacionales, no relacionales o NoSQL, distribuidas, orientadas a objetos y gráficas, sin embargo, los dos tipos de bases de datos que principalmente se usan son las bases de datos relacionales y las no relacionales, donde las bases de datos relacionales son utilizadas por su facilidad de uso gracias a las múltiples aplicaciones de gestión de bases de datos donde desde una herramienta gráfica se hace más fácil la accesibilidad a los datos, además que es más sencilla para identificar la trazabilidad de los datos y su distribución en el sistema.

Para un manejo más eficiente y amigable para el administrador de la base de datos es necesario hacer uso de los sistemas gestores de bases de datos relacionales, los cuales brindan una herramienta gráfica de gestión de los datos desde los cuales se pueden ejecutar las

operaciones básicas de una base de datos, algunos de los sistemas gestores de bases de datos relacionales son:

- MySQL
- Oracle database
- Microsoft SQL Server
- PostgreSQL

Por otra parte, están las bases de datos no relacionales cuya utilización es cada vez mayor en aplicaciones modernas dado a sus características, las bases de datos NoSQL tienen la ventaja de ser flexibles, escalables y tener un buen rendimiento cuando se manejan grandes volúmenes de información, además, estas bases de datos no utilizan una esquema rígido o estructuras definidas desde el instante inicial lo que permite que se puedan adaptar fácilmente a este tipo de proyectos

Para estas no son tan necesarios los sistemas gestores de base de datos aún así existe oferta de sistemas gestores para estas, entre las BD no relacionales más populares y utilizadas están:

- Mongo DB
- Cassandra
- Redis
- DynamoDB
- Neo4j

3.1.7. IoT

"IoT es la interconexión digital de objetos cotidianos con internet. Permite el cambio automático de información con otros dispositivos o centros de control sin intervención humana, capturando gran cantidad de información clave sobre uso y rendimiento." (López y Cardenas, 2019, p. 75)

3.1.8. HTTP (Hypertext Transfer Protocol)

HTTP es un protocolo sin estado de la capa de aplicación para sistemas de información de hipertexto distribuidos y colaborativos.

HTTP se basa en el envío de mensajes sobre el protocolo de transporte TCP:

- El cliente envía un mensaje de petición a un servidor, solicitando realizar una acción sobre un recurso determinado (habitualmente, obtener el recurso).
- El servidor envía un mensaje de respuesta a la petición del cliente (habitualmente, incluyendo el recurso solicitado). (Fisteus J, 2022, p. 2)

Si bien, hay protocolos que pueden brindar un mayor nivel de seguridad se desarrolla el prototipo funcional con este protocolo.

4. Antecedentes

En el estado del arte se puede evidenciar que la cerradura electrónica inteligente es una herramienta que tiene vastos estudios previos, sin embargo, siempre hay oportunidades de mejora y diferencias entre uno y otro. Puntualmente se encuentran un producto comercial desarrollado y tres proyectos académicos relacionados con el control de acceso al hogar.

4.1. Lockey Colombia: Cerradura electrónica Facelok

Se encuentra una cerradura electrónica disponible en el mercado ofrecida por lockey Colombia cuyas funcionalidades mencionan el reconocimiento facial, tarjetas de proximidad y código PIN, sin embargo, no se menciona la tecnología con la que se desarrolló.

4.2. Desarrollo de tecnología domótica con aplicación a la gestión de seguridad en cerraduras electrónicas

En el proyecto realizado por Sustaita, Rivera, Fernández (2013) de la facultad de ingeniería mecánica y eléctrica de la universidad autónoma de nuevo león se logró desarrollar una cerradura electrónica orientada a caja fuerte con 3 candados, interruptor de cierre con cerrojo, clave numérica (PIN) y un sistema biométrico de huellas dactilares y un ‘buzzer’ el cual sirve como un sonido de alerta cuando el acceso es denegado, el proyecto fue desarrollado con un microcontrolador Arduino y sus librerías.

4.3. Cerradura electrónica inteligente para el hogar

En el proyecto realizado por Escate, Olortegui, Rodriguez, Castillo y Panana (2019) más que una investigación enfocada en la creación de la herramienta se centró en realizar un estudio desde la perspectiva empresarial y la creación de empresa siendo la cerradura electrónica el producto que debía comercializar la empresa, se menciona que el producto va a ser controlado desde un dispositivo móvil para su apertura y cierre.

4.4. Prototipo para el control de una cerradura electrónica por medio de reconocimiento facial

En el proyecto realizado por Moreno, J (2016) se desarrolla un prototipo de cerradura electrónica inteligente con reconocimiento facial, el dispositivo cuenta con un procesador digital

de señales de Texas Instruments (TMS320DM6437s) y su sistema de reconocimiento facial fue implementado con el algoritmo de Modelo KLGaussian de color en espacio YCbCr, el procesamiento de las imágenes se realizó con Matlab.

4.5. Implementación de un sistema de control de acceso basado en reconocimiento facial

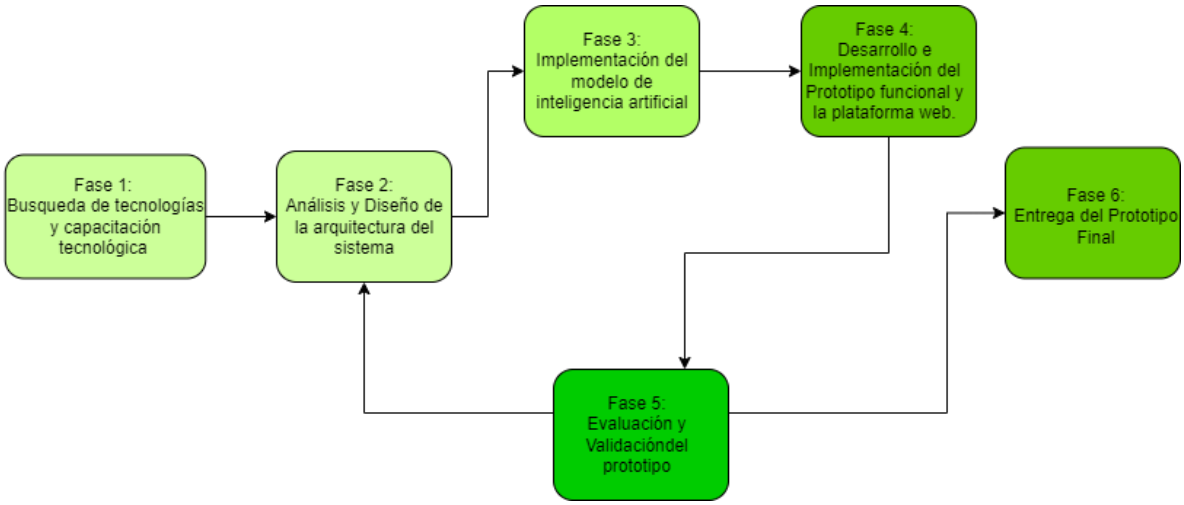
En el proyecto realizado por Poveda y Merchán se presentan aspectos de la implementación de un sistema de control de acceso basado en reconocimiento facial el cual verifica en tiempo real si las personas que entran a las instalaciones forman parte de la base de datos del personal que labora en las mismas.

5. Marco Metodológico

Para el desarrollo del proyecto se definieron una serie de fases metodológicas donde para cada una de ellas se espera un resultado tangible o intangible significativo y un avance en el desarrollo del proyecto, se propusieron 6 fases, en la siguiente figura se puede observar un diagrama con el flujo de las fases y a continuación se mencionan las actividades y resultados esperados en cada fase del proyecto.

Figura 10

Esquema de metodología de trabajo



5.1. Fase 1: Búsqueda de tecnologías y capacitación tecnológica

Para dar inicio se debe conocer los fundamentos y principios de la metodología ágil, por lo que inicialmente se debe realizar una introducción y adaptación a la metodología Kanban, además, en este momento se puede iniciar a levantar los requerimientos funcionales, no funcionales y tecnológicos, como lo son los componentes hardware y tecnologías software.

En esta etapa se analizará la viabilidad de las diferentes opciones tecnológicas disponibles como microcontroladores, computadora móvil, lenguajes de programación, librerías, modelos de

inteligencia artificial de reconocimiento facial, entre otros, y se seleccionarán los más adecuados para dar cumplimiento a los objetivos planteados, esto con el fin de definir los componentes de hardware y software necesarios para el monitoreo, adquisición y estudio de los datos requeridos por la plataforma.

Para dar inicio al proyecto se investigará acerca de las tecnologías escogidas para la implementación del prototipo, además del software y hardware asociados a este.

Actividades:

- Investigación y análisis de los conceptos fundamentales.
- Investigación de tecnologías utilizadas en proyectos similares aplicados en el campo.
- Búsqueda de herramientas de relevancia en la realización del proyecto.

Resultados:

- Dominio de conceptos fundamentales.
- Conocimiento del estado del arte en proyectos similares.
- Listado de potenciales tecnologías y componentes a utilizar en el prototipo.

5.2. Fase 2: Análisis y diseño de la arquitectura del sistema

En esta etapa se realizará el diseño del sistema en general, se iniciará por la definición de los requerimientos del sistema, diagramas de actividades, casos de uso y partiendo de esto se definirá la arquitectura hardware, software y la integración de estos.

Actividades:

- Definición de requerimientos funcionales y no funcionales.
- Análisis de requerimientos y creación de diagramas de casos de uso y diagramas de actividades.
- Comparación y selección de tecnologías y componentes.
- Diseño de la plataforma del software.
- Diseño de la arquitectura hardware.

Resultados:

- Listado de requerimientos funcionales y no funcionales.
- Diagramas de casos de uso y diagramas de actividades.
- Tecnologías para usar en el prototipo.
- Diseño de la plataforma software.
- Arquitectura del prototipo.
- Bosquejo de la red de conectividad de los dispositivos y su conexión con la plataforma web.

5.3. Fase 3: Implementación del modelo de inteligencia artificial

Como resultado de la investigación de la fase anterior sobre modelos de IA entrenados para el reconocimiento facial, se utilizará el más adecuado para el uso de nuestro prototipo.

Actividades:

- Investigación y documentación del modelo IA seleccionado.
- Implementación del modelo de reconocimiento facial.

Resultados:

- Modelo de reconocimiento facial listo para uso e implementación en el prototipo.

5.4. Fase 4: Desarrollo e Implementación del Prototipo y la Plataforma IoT.

Una vez definidas las tecnologías y los diseños de la plataforma y el prototipo, se procederá al desarrollo e implementación (hardware y software). También se comprobará el cumplimiento de los objetivos planteados, con lo cual se determinará si es necesario volver a la fase del diseño del sistema o a la implementación del modelo de IA y refinar el modelo y hacer una nueva iteración de estas fases o continuar con la evaluación y validación del prototipo funcional.

Actividades:

- Implementación del esquema de componentes de hardware.
- Implementación del modelo IA en el prototipo.
- Desarrollo de la plataforma web: servidor, backend, base de datos y frontend.
- Construcción del prototipo a escala.

Resultados:

- Base de datos y Plataforma web desarrollados.
- Arquitectura hardware implementada.
- Prototipo funcional y apto para validaciones.
- Documentación del prototipo funcional.

5.5. Fase 5: Evaluación y Validación de requerimientos del prototipo

En esta fase del proyecto se contará con el prototipo funcionando, donde se realizará una serie de pruebas que evaluarán el prototipo de acuerdo con el comportamiento, se espera que el

prototipo sea capaz de identificar correctamente al usuario que usa el dispositivo, también se harán las respectivas validaciones con el fin de determinar si el prototipo cumple con los requerimientos planteados y no se presentan errores

Posteriormente, basados en el análisis de los resultados para el prototipo se decidirá si es necesario implementar correcciones o mejoras con el fin de cumplir con un óptimo desempeño del prototipo, se determinará si el prototipo cumple con los objetivos y se puede continuar a finalizar y hacer entrega de este o es necesario volver a la fase de análisis y diseño del sistema, refinamiento del modelo de IA o a la del desarrollo de la plataforma web y hacer modificaciones en el prototipo, en este caso se repetirá el ciclo.

Actividades:

- Almacenamiento de datos en la nube e interacción con ellos.
- Verificación de la lógica e implementación.
- Validación del sistema creado mediante un caso de uso.
- Análisis de los resultados obtenidos.

Resultados:

- Prototipo validado.
- Informe de resultados.
- Toma de decisiones de acuerdo con el informe de resultados.

5.6. Fase 6: Entrega del Prototipo Final

Se realizará la documentación formal del proyecto y su posterior presentación ante la Escuela de Ingeniería de Sistemas e Informática de la Universidad Industrial de Santander culminando de esta manera el proyecto de grado.

Actividades:

- Presentación y exposición del prototipo final con los ajustes realizados.
- Entrega del libro.

Resultados:

- Prototipo final validado.
- Finalización del proyecto.

6. Capacitación tecnológica

6.1. Capacitación técnica y tecnológica

Debido a la falta de conocimiento y experiencia en tecnologías y componentes se da inicio al proyecto con una investigación de cuales podrían ser las tecnologías que más se ajusten a la idea de lo que se plantea desarrollar, es por ello que se inicia con una capacitación tecnológica en lenguajes de programación donde se hace una búsqueda y analizando los diferentes ‘stacks’ de desarrollo con mayor soporte por la comunidad y con mayor documentación disponible, también se analiza cuáles pueden ser útiles para el proyecto de acuerdo a la idea que se plantea, se visualizan cursos por medio de plataformas digitales como Udemy y Youtube, en las cuales en cada plataforma se imparte cierto conocimiento de distintos

lenguajes de programación y un abrebocas a lo que es cada stack de desarrollo así como el enfoque de estos.

Partiendo del análisis realizado se elige como potencial stack el ‘MEAN’ por sus siglas Mongo, Express, Angular y Node, los cuales son un stack de tecnologías basadas en el lenguaje de programación javascript, el cual es versátil y es escalable con el tiempo, por lo que se inicia una capacitación en esos frameworks, sin embargo, no se descartan aún los demás stacks de desarrollo y lenguajes de programación.

Para un mayor conocimiento y manejo a profundidad en estos frameworks basados en javascript se optó por estudiar cursos en la plataforma Udemy.

Los cursos vistos fueron los que ofrece el instructor Fernando Herrera de Udemy (Ver figuras 11 y 12).

Figura 11

Curso Angular: De cero a experto

Nota. Curso visto para el desarrollo frontend del proyecto. Tomado de Udemy, Curso Angular: De cero a experto, por Herrera F. https://www.udemy.com/course/angular-2-fernando-herrera/?kw=angular+de+cero+a+experto&src=sac

Figura 12

Curso Node: De cero a experto

Nota. Curso visto para el desarrollo backend del proyecto. Tomado de Udemy, Curso Node: De cero a experto, por Herrera F. https://www.udemy.com/course/node-de-cero-a-experto/

Además, para el componente electrónico del proyecto era necesario investigar acerca de las diferentes computadoras móviles que hay en el mercado con el fin de elegir la que mejor se ajuste a las necesidades del prototipo a desarrollar en el proyecto, por lo que se hizo una búsqueda acerca de las computadoras de placa única de distintas compañías entre las cuales destacaron el Khadas VIM4 de la compañía Khadas, Orange Pi 3 LTS de Orange pi, la raspberry Pi 4 de raspberry y la Jetson nano de Nvidia, siendo esta ultima una gran opción debido a su GPU y librerías especializadas para la implementación de algoritmos de inteligencia artificial.

Finalmente, debido a que el proyecto requiere de un algoritmo de inteligencia artificial, fue necesario investigar acerca de los distintos algoritmos existentes que hicieran más fácil y rápido el desarrollo del prototipo reduciendo los tiempos de desarrollo de este, entre los posibles algoritmos o métodos de reconocimiento facial pre entrenados se encuentran la librería face recognition de OpenCv en Python, la librería FaceNet de Tensorflow, , la librería face api de javascript, entre otras opciones.

7. Diseño Experimental

7.1. Definición de requerimientos funcionales y no funcionales.

Se definen roles para la accesibilidad a cada una de las características y funcionalidades que tiene la plataforma, las cuales se plasman a continuación.

Administrador: Tiene acceso a todos los módulos del sistema y es quien gestiona los usuarios del hogar, se propone que sea el usuario administrador quien contrata, es dueño o administra el hogar.

Habitante: Puede ingresar a la plataforma para mirar su historial de visitas en la casa, también puede ingresar a la casa siempre que lo requiera.

Visitante: Puede ingresar a la casa.

Tabla 1

Definición de roles y funcionalidades

Funcionalidad	**Rol**
Creación de usuarios	Administrador
Edición del estado del usuario	Administrador
Visualización de información personal de ingresos al hogar	Administrador, Habitante y Visitante
Ingreso al hogar mediante autenticación	Administrador, Habitante y Visitante
Visualización de histórico de ingresos y estadísticas	Administrador, Habitante y Visitante
Eliminación de usuarios	Administrador

7.1.1. Requerimientos funcionales

Se define un listado con los requerimientos funcionales mínimos con los que debe contar la plataforma web y el sistema en general, estos se definen por roles y describiendo la funcionalidad o característica con la que debe contar el aplicativo.

7.1.1.1. Requerimientos funcionales del aplicativo web

Se define el listado de requerimientos funcionales mínimos con los que debe contar el aplicativo web.

Tabla 2

Requerimientos funcionales del aplicativo web

N° Requerimiento	Descripción	Rol
RFW-1	El aplicativo web debe ser capaz de autenticar y autorizar el acceso del usuario.	Administrador, Habitante, Visitante
RFW-2	Se debe poder crear un usuario con sus respectivos datos.	Administrador
RFW-3	Se debe poder actualizar (cambiar) el estado de usuario en el portal web.	Administrador
RFW-4	Se debe poder eliminar usuarios.	Administrador
RFW-5	Se deben poder visualizar una vista general de los usuarios con sus datos.	Administrador
RFW-6	Se debe poder visualizar una gráfica con los ingresos de un usuario específico en un lapso determinado.	Administrador, Habitante, Visitante
RFW-7	Se debe poder visualizar una gráfica con el total de ingresos de todos los usuarios en un lapso determinado.	Administrador

7.1.1.2. Requerimientos funcionales del sistema

Se define el listado de requerimientos funcionales mínimos con los que debe contar el sistema de autenticación por reconocimiento facial en su integración con el aplicativo web.

Tabla 3

Requerimientos funcionales del sistema

N° Requerimiento	Descripción	Rol
RFS-1	El sistema debe ser capaz de procesar los datos como la imagen, nickname y el estado del usuario.	Administrador, Habitante, Visitante.
RFS-2	El sistema debe ser capaz de detectar e identificar la cara del usuario registrado.	Administrador, Habitante, Visitante.
RFS-3	El sistema debe ser capaz de autenticar y autorizar el acceso del usuario al hogar.	Administrador, Habitante, Visitante.
RFS-4	El sistema debe ser capaz de captar datos de ingreso una vez se conceda el acceso al hogar.	Administrador, Habitante, Visitante.

7.1.2. Requerimientos no funcionales

Se define un listado con los requerimientos no funcionales mínimos con los que debe contar la plataforma web y el sistema en general, estos se definen por roles y describiendo la funcionalidad o característica con la que debe contar el aplicativo.

7.1.2.1. Requerimientos no funcionales del aplicativo web

Para garantizar un funcionamiento óptimo, se detallaron los requisitos no funcionales mínimos que el aplicativo web deberá cumplir.

Tabla 4

Requerimientos no funcionales del aplicativo web

N° Requerimiento	Nombre	Descripción
RNFW-1	Usabilidad	La interfaz de usuario debe ser intuitiva y comprensible. La aplicación debe tener diseño responsive.
RNFW-2	Escalabilidad	Los módulos del aplicativo tanto backend como frontend debe estar separados. Permitir la fácil integración con otras librerías, componentes o servicios.
RNFW-3	Soporte	El aplicativo debe ser soportado por los navegadores Chrome, Mozilla Firefox y Edge.
RNFW-4	Seguridad	Las contraseñas de usuario deben estar encriptadas. Para manejo de permisos por roles para la modificación de datos se debe contar con tokenización para la autenticación y autorización de usuarios.

7.1.2.2. Requerimientos no funcionales del sistema

Se define el listado de requerimientos no funcionales mínimos con los que debe contar el sistema de autenticación por reconocimiento facial en su integración con el aplicativo web.

Tabla 5

Requerimientos no funcionales del sistema

N° Requerimiento	Nombre	Descripción
RNFS-1	Usabilidad	El sistema debe ser intuitivo al momento de intentar la autenticación por reconocimiento facial. El sistema debe ser claro al momento de conceder o denegar el acceso.
RNFS-2	Escalabilidad	Diseñar de forma modular los componentes. Permitir la integración con otros servicios y librerías.
RNFS-3	Seguridad	Se debe garantizar que el sistema no permita el acceso a usuarios que no se encuentran registrados en el portal.

RNFS-4	Conectividad	Se requiere conexión a internet de forma inalámbrica Se requiere que el sistema funcione con batería recargable en caso de no contar con electricidad domestica.

7.2. Análisis de requerimientos

Una vez se han definido los requerimientos funcionales y no funcionales se continua con el análisis de requerimientos con el fin de proporcionar un mejor entendimiento a cada una de las funcionalidades del prototipo.

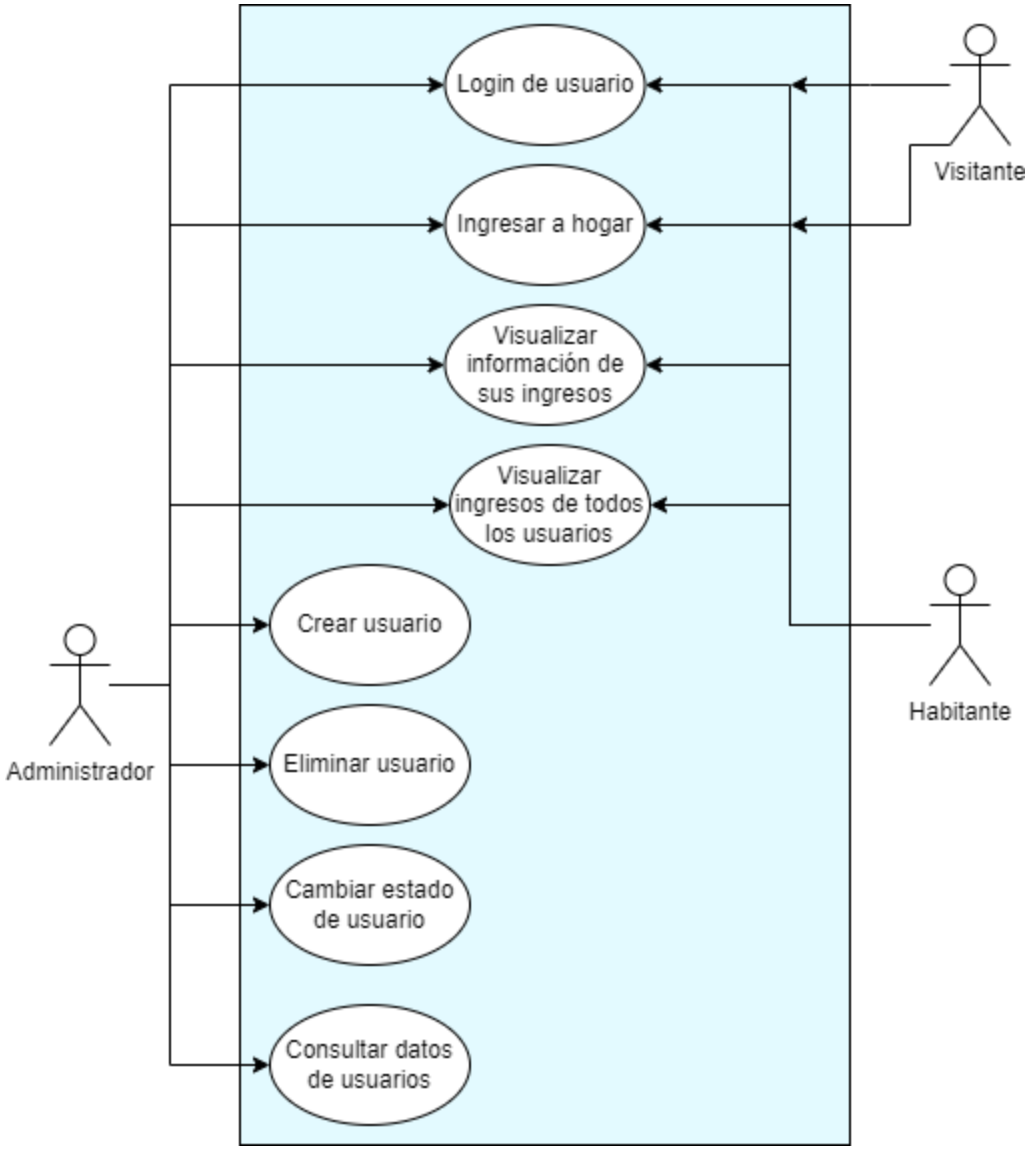
7.2.1. Casos de uso

Se elabora un diagrama de casos de uso para revisar las funcionalidades que tendría la plataforma en los diferentes roles, con el objetivo de identificar de manera clara las acciones realizadas por cada uno.

Se recuerda que el rol administrador puede ingresar al hogar porque será él quien contrata, es dueño o administra al sistema y hogar

Figura 13

Diagrama de casos de uso



7.2.2. Diagrama de actividades

Para una mayor claridad de las principales actividades que debe permitir realizar el sistema se crean los siguientes diagramas de actividades.

7.2.2.1. Diagrama de actividades para el ingreso al hogar

Se define el flujo lógico de actividades que debe realizar cada uno de los actores físicos o lógicos del sistema para dar el ingreso al hogar a un usuario con rol administrador, habitante o visitante.

Esta actividad inicia con un usuario administrador, habitante o visitante activando el sistema de reconocimiento facial, posteriormente el sistema realiza las validaciones para la autenticación del usuario de acuerdo con las coincidencias con las imágenes en la base de datos para así conceder el acceso al hogar, luego envía datos del ingreso al servidor donde se procesarán estos datos y serán almacenados en la base de datos (Ver figura 14).

7.2.2.2. Diagrama de actividades para la creación de usuarios nuevos desde el portal web

Se define el flujo lógico de actividades que debe realizar el usuario con rol administrador para la creación de un nuevo usuario.

Esta actividad inicia con el usuario administrador haciendo login en el sistema y posteriormente creando un usuario nuevo completando el formulario, luego el sistema realiza las respectivas validaciones y posteriormente crea el nuevo usuario y lo guarda en la base de datos (Ver figura 15).

Figura 14

Diagrama de actividades para ingreso al hogar

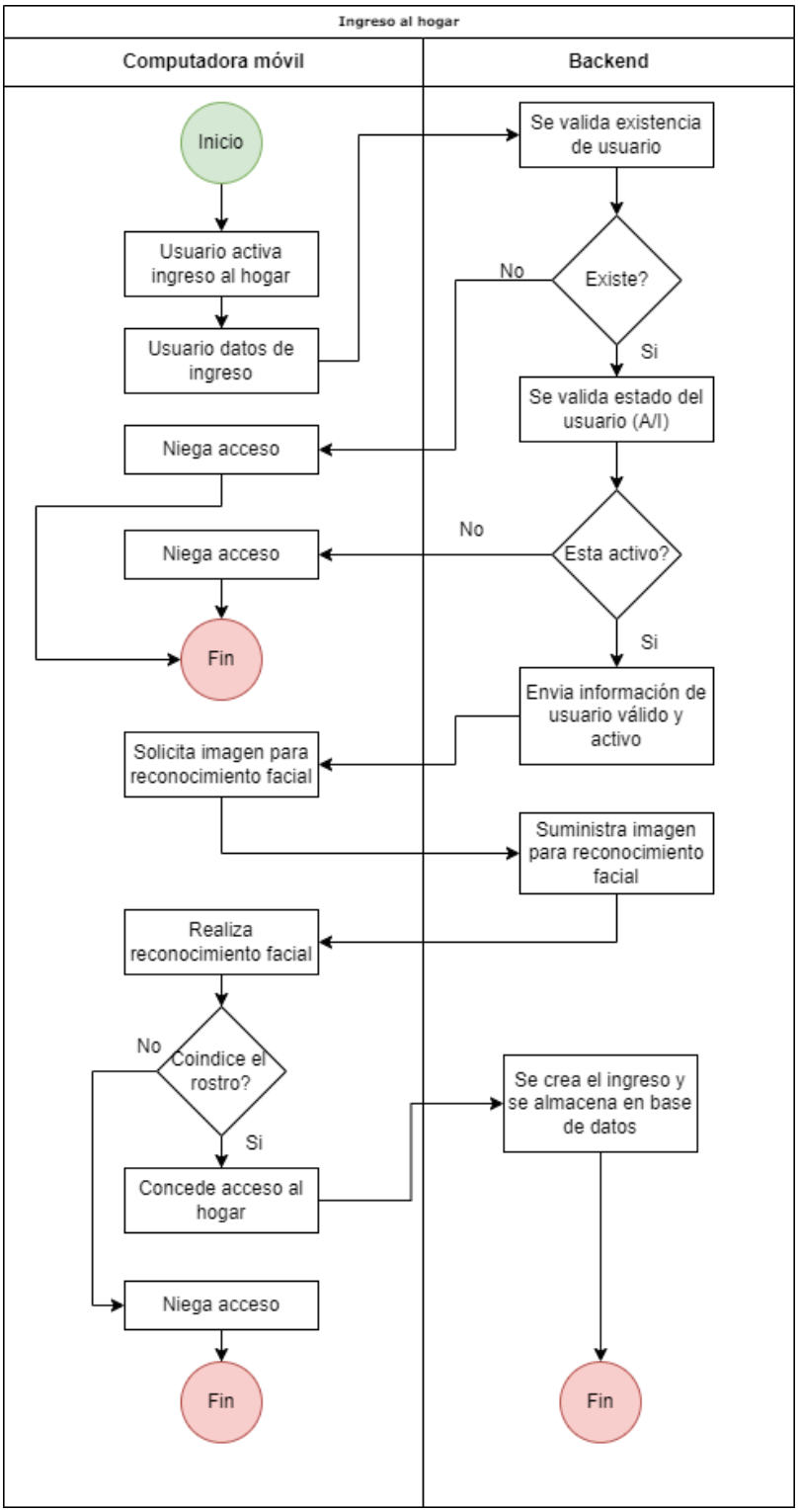
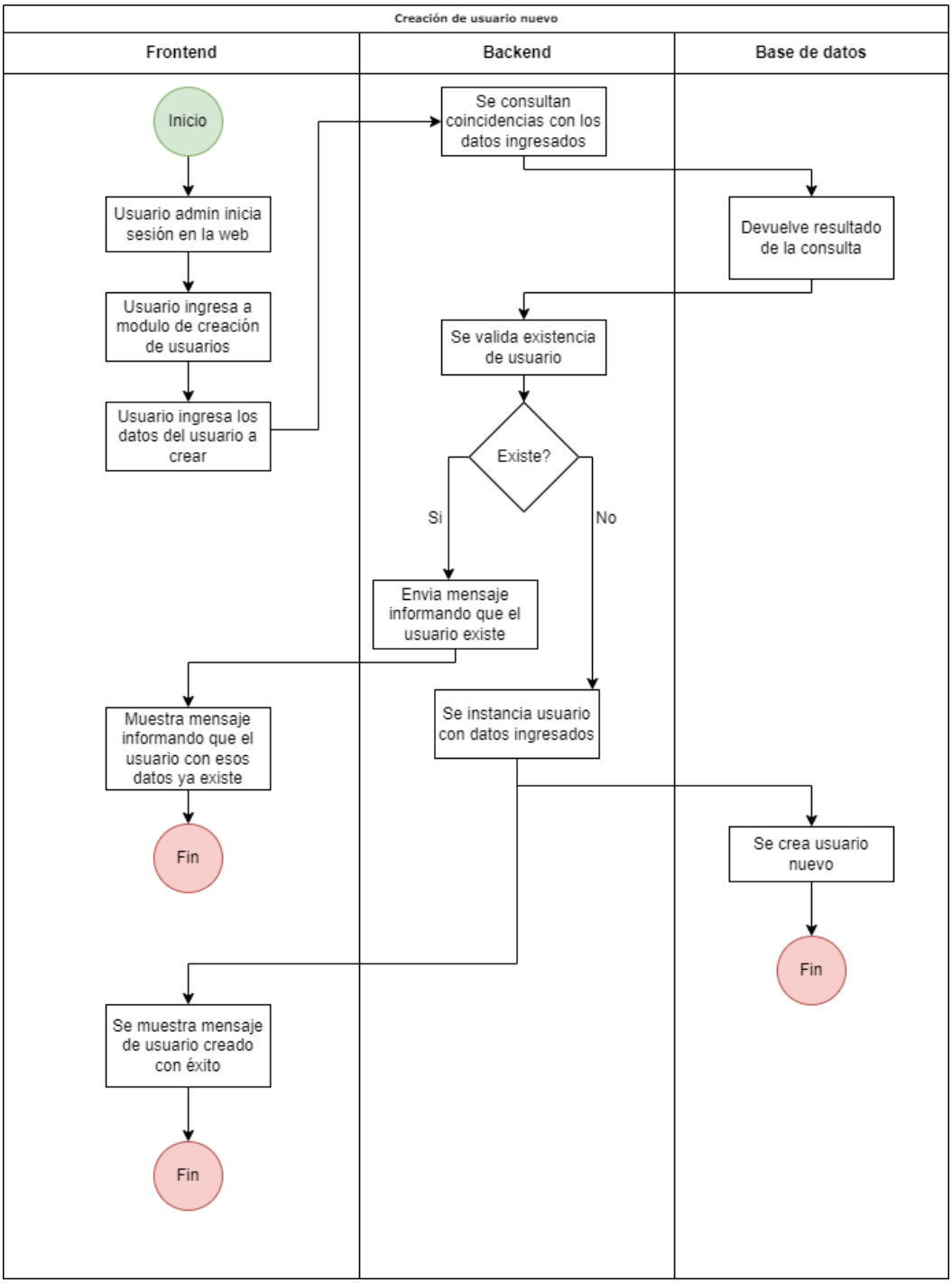


Figura 15

Diagrama de actividades para creación de usuarios nuevos



7.3. Comparación y selección de tecnologías y componentes.

Existe una gran cantidad de tecnologías de las que hacer uso en el proyecto, por eso se optó por realizar una comparación de estas para cada uno de los componentes del sistema.

7.3.1. Comparación de bases de datos

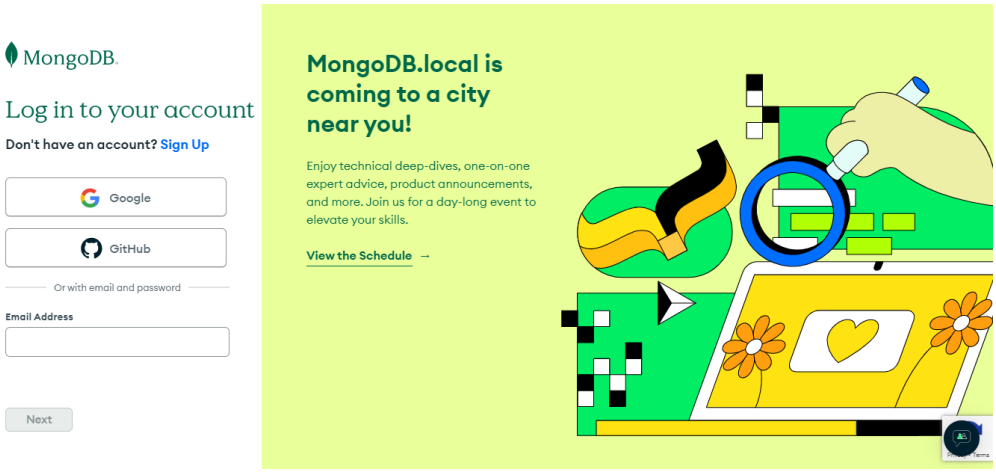
Para efectos del desarrollo del proyecto se optó por una base de datos no relacional ya que al no tener tantas entidades no es necesario realizar muchas relaciones, además de ser un proyecto que se basará en registros, donde se guardaran registros para los usuarios que ingresen al hogar e información de los usuarios registrados.

La elección en este caso es MongoDB debido a su fácil integración e implementación con tecnología basadas en javascript como Node.js la cual se propone como tecnología a usar en el backend.

Además, se opta por almacenar la información en la nube con MongoDB Cloud (Ver figura 16), donde al no contar con un alto flujo de información ni almacenamiento no tiene ningún costo, además como sistema gestor local mientras se hace el desarrollo de la aplicación se usará MongoDB Compass para visualizar y gestionar de una forma sencilla y amigable los datos almacenados en la base de datos. (Ver figura 17)

Figura 16

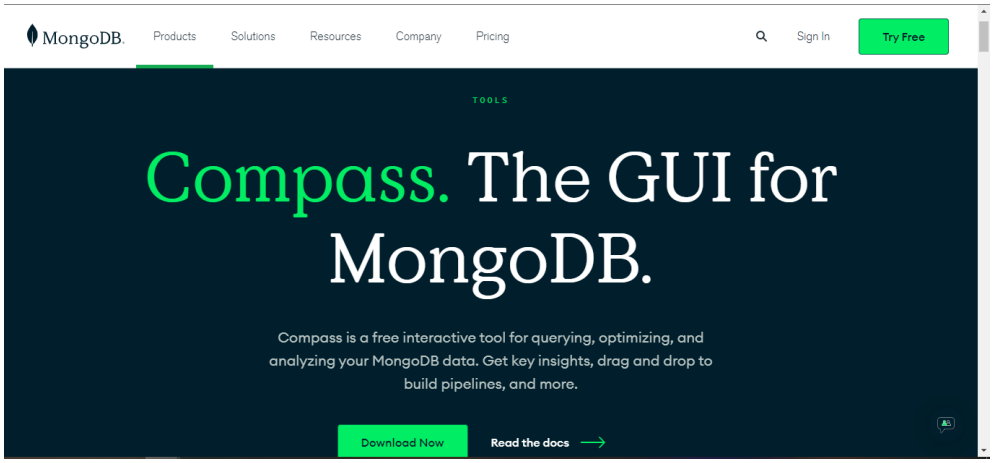
MongoDB Cloud



Nota. MongoDB – página principal de registro. Tomado de MongoDB. https://account.mongodb.com/account/login

Figura 17

MongoDB Compass



Nota. Página principal de MongoDB Compass. Tomado de MongoDB Compass. https://www.mongodb.com/products/compass

7.3.2. Comparación de lenguajes y frameworks para el backend

Partiendo de la definición de lo que es banckend se requiere de un lenguaje de programación que sea robusto y riguroso que resguarde los datos y que además sea escalable. Se

realiza una comparativa entre los lenguajes de programación más populares y utilizados en la actualidad donde se destacan los lenguajes Javascript, Java, PHP, Python, .NET, entre otros.

Finalmente, para el desarrollo se decide usar como lenguaje Javascript con su entorno de tiempo de ejecución Node.js, ya que se hace llamativo por su comunidad participativa, su curva de aprendizaje poco pronunciada y su fácil integración con la base de datos seleccionada, además, haciendo uso de la librería Express.js se puede levantar el servidor cuya implementación es sencilla en node.js.

7.3.3. Comparación de lenguajes y frameworks para el frontend

Para agregar interactividad a las páginas web esta nuevamente el mencionado javascript, para el que existe una serie de tecnologías y frameworks basados en este lenguaje los cuales fueron mencionados en el marco referencial del proyecto, entre estos se hace una comparación para tomar la decisión final.

El framework elegido para agregar interactividad al frontend es Angular por el modelo SPA y su modelo basado en componentes lo que hace que no sea necesario recargar la página para actualizar la información en ella, además, por sus propiedades de escalabilidad e integración con otras librerías para el frontend.

7.3.4. Comparación de computadoras móviles

Para el desarrollo del proyecto es de vital importancia contar con una computadora móvil que tenga buenas especificaciones técnicas de rendimiento y velocidad ya que inicialmente se propone que el reconocimiento facial se haga desde esta, en la investigación realizada se hace una comparación de las especificaciones técnicas más relevantes para el correcto funcionamiento y

buen rendimiento del proyecto, la comparativa se realiza entre la Khadas VIM4, Orange Pi 3 LTS, la raspberry Pi 4 y la Jetson nano.

Como información adicional a la presentada en el marco referencial se tiene que cualquiera de estas placas es compatible con cámaras, solenoides y otros componentes hardware que son requeridos para el proyecto, si bien algunas no son compatibles de fabrica con wifi se les puede conectar, además, todas las placas son compatibles con el sistema operativo Linux y específicamente a la distribución Ubuntu.

Se tiene que la VIM4 tiene un costo de 219,90 siendo esta la más completa y con mayor capacidad pero con un costo de 220 dólares lo que la hace costosa, la Orange Pi 3 LTS también cuentas con unas buenas especificaciones de hardware que serían suficientes para el proyecto y tiene un costo bastante accesible el cual es de 48 dólares, la raspberry Pi 4 tiene versiones desde 1 a 8 GB y de acuerdo con eso varía su precio donde la más costosa tiene un precio de 150 dólares en su versión de 8GB de RAM, y finalmente se encuentra la jetson nano de 4 GB con 150 dólares.

Si bien la Orange Pi 3 LTS sería la opción más económica y la que cumple con especificaciones técnicas requeridas para el desarrollo del proyecto, se tiene que la jetson nano contiene un kit de desarrollo de Nvidia en el cual está integrado OpenCV la cual es una librería que potencialmente se podría estar utilizando para la implementación del modelo de reconocimiento facial.

La elección final es la Jetson nano por sus especificaciones de rendimiento, además, se elige esta porque es a la que más fácil se tiene acceso, ya que la escuela de ingeniería de sistemas e informática tiene en su inventario una jetson nano, aunque la que suministrará la escuela en

calidad de préstamo para efectos del desarrollo del prototipo es la versión de 2 GB de memoria RAM, sigue siendo suficiente para la implementación de un modelo de reconocimiento facial de las características del proyecto.

7.3.5. Comparación de librerías de reconocimiento facial

Para efectos prácticos de un sistema de reconocimiento facial es necesario la implementación de un modelo de inteligencia artificial con estas características, sin embargo, al ser un proyecto IoT que realiza reconocimiento facial y no se basa en el desarrollo de un modelo, se optó por hacer uso de un modelo de inteligencia artificial pre entrenado con el fin de hacer más fácil y veloz el desarrollo del prototipo, por lo tanto, se realizó una búsqueda de los modelos más utilizados y con mayor comunidad con el propósito de analizar las bondades de cada uno y teniendo también como base las tecnologías que se seleccionaron previamente.

Se tiene la librería face recognition de OpenCV y la librería FaceNet de Tensorflow las cuales son en lenguaje Python, lo que no sería ningún impedimento ya que el sistema operativo que corre en la jetson nano es basado en Linux y la instalación y ejecución de Python sobre Linux es fácil de configurar.

Por otra parte, está la librería basada en lenguaje javascript llamada face api que continúa siendo una opción interesante ya que funciona bien en el lado del servidor o en el frontend del lado del cliente.

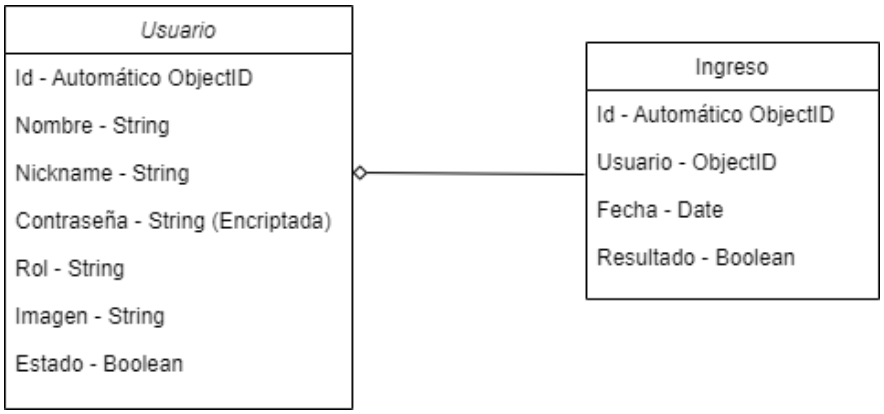
Se toma la decisión de usar la librería face recognition de OpenCV, ya que como se mencionó en la comparación de computadoras móviles, el kit de desarrollo de la jetson nano contiene CUDA la cual es una herramienta de desarrollo que tiene integrada OpenCV, lo que haría más fácil y rápido la implementación del modelo.

7.4. Modelo de datos

Para una mejor comprensión de la estructura de la base de datos no relacional y de cómo se almacenará la información se construyó un diagrama de modelo de datos en la cual se puede observar la interacción entre colecciones en base a referencias donde se utiliza la librería Mongoose para crear esquemas y patrones de información según el tipo de dato y las características requeridas o valores por defecto.

Figura 18

Modelo de datos



Se tiene que para el usuario se almacenará un identificador único para cada registro, además, se almacenará información personal como el nombre y una imagen, además deberá crearse con un nombre de usuario o nickname, una contraseña que servirá para acceder al sistema y al hogar, le será asignado un rol y finalmente se almacenará el estado del usuario.

Para la colección de ingresos se tiene que se almacenará un identificador único para cada registro con la fecha en que fue el ingreso, el resultado del ingreso, si fue aprobado o rechazado, una referencia del usuario que intentó ingresar.

Por otra parte, se tiene que para el reconocimiento facial se debe calcular un vector de características, estas inicialmente no se almacenarán en la base de datos sino será en la memoria de la jetson nano durante la ejecución del algoritmo.

7.5. Diseño del Mockup

Para iniciar con el desarrollo de la interfaz de usuario se define inicialmente un diseño como guía para la posterior implementación con código, para ello se sugieren las siguientes vistas:

Se propone una página de login en la cual se tiene la palabra de bienvenido y los campos para ingresar el nombre de usuario y la contraseña. (Ver figura 19)

Figura 19

Mockup del Login

Se propone una segunda página que se verá una vez el usuario inicia sesión, esta deberá tener a la izquierda una sección con la imagen del usuario logeado y las opciones a las que tiene acceso, como lo son la gestión de usuario en el caso del usuario administrador y, en la sección de la derecha se deberá ver gráficamente la cantidad de ingresos por día de cada usuario de los últimos siete días. (Ver figura 20)

Figura 20

Mockup vista inicial

Se propone una tercera página la cual será visible únicamente por el usuario administrador y está será para la gestión de usuarios, donde se deben ver los usuarios registrados con sus datos personales y las fotos cargadas de cada uno (Ver figura 21).

Se propone una cuarta vista la cual será visible únicamente por el usuario administrador y corresponde a la creación del usuario, donde se debe ingresar la información personal del usuario a registrar y una foto para el reconocimiento facial (Ver figura 22).

Figura 21

Mockup gestión de usuarios

Figura 22

Mockup creación de usuarios

Se propone una quinta vista la cual será visible para todos los usuarios, en esta el usuario que ingresó podrá ver si información personal, así como el rol que tiene en el hogar y las veces que ha ingresado (Ver figura 23).

Figura 23

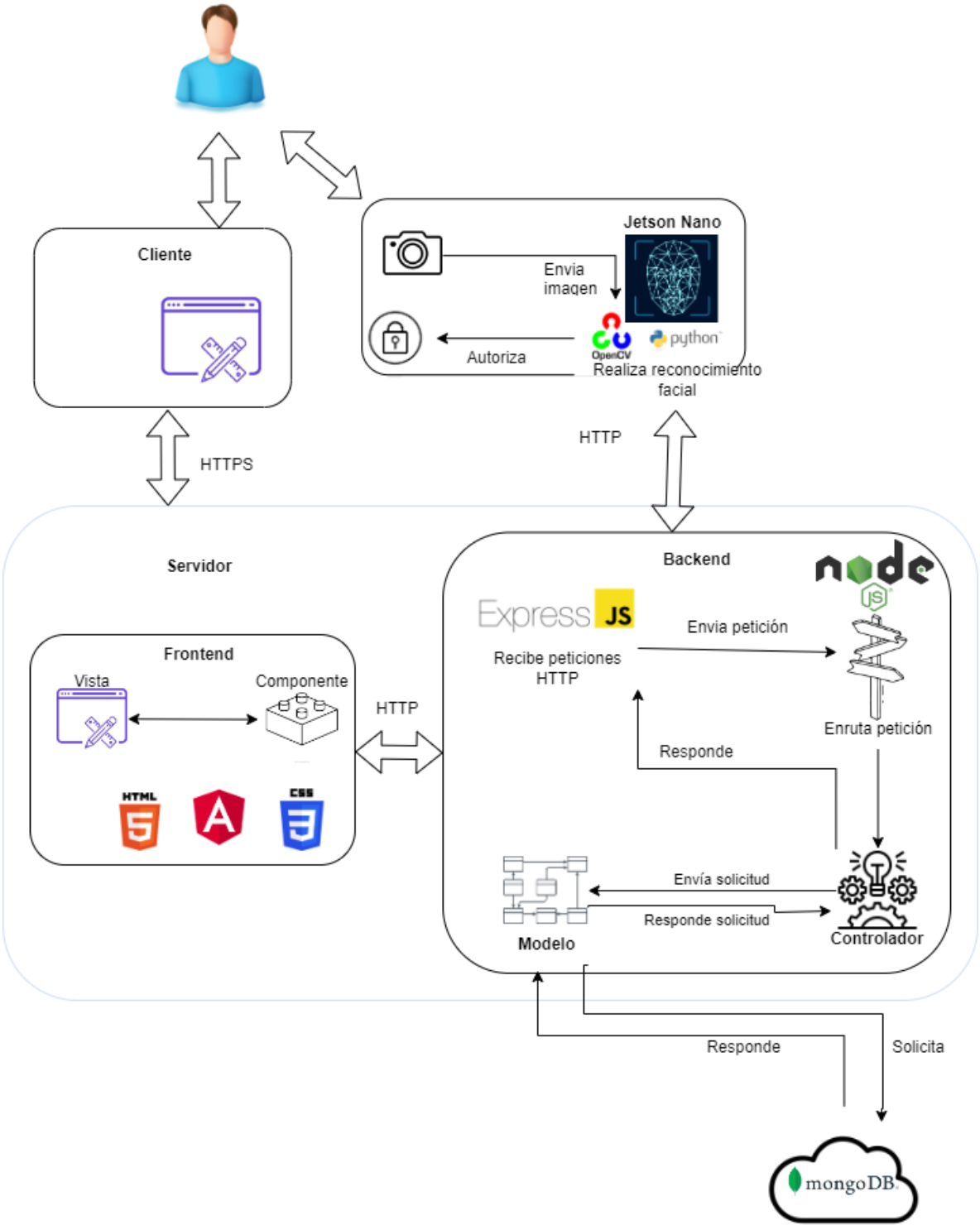
Mockup Dashboard de usuario

7.6. Diseño de la arquitectura.

Después de estructurar los mockups y definir los requerimientos solicitados, es crucial establecer las herramientas de trabajo. Para facilitar este proceso, se diseña una arquitectura que permite visualizar a gran escala la forma en que se integra cada una de los componentes y las tecnologías que se usan para estas, creando así un sistema estructurado y organizado.

Figura 24

Visión de arquitectura Prototipo 1



La figura muestra que el diseño arquitectónico está dividido en 3 secciones o componentes, que son:

1. Hardware: Esta sección hace referencia a la computadora móvil y los dispositivos periféricos que hacen parte de la captura de datos y en el cual esta el modelo de reconocimiento facial.
2. Servidor: Este componente hace referencia al lugar donde se almacenan e integran el frontend y el backend creando así un solo sistema donde los datos son recibidos y almacenados.
3. Cliente: Esta sección hace referencia al lugar desde donde se conecta el usuario del aplicativo web y consume los servicios del servidor.

Se puede observar que el sistema se divide en 3 secciones, donde el sistema podría ser iniciado desde el componente del hardware o en el cliente, cuando es el caso que se inicia desde el cliente, se tiene que el usuario al realizar ciertas acciones desde la página del aplicativo web envía una serie de solicitudes HTTP desde el frontend hacia el backend, donde este las maneja a través del controlador, el cual tiene configuradas las rutas en la API (Interfaz de programación de aplicaciones) y es quien se encarga de transformar y obtener la información mediante los modelos y así dar la respuesta que espera el usuario.

Cuando es el caso que se inicia desde la cerradura este funciona prácticamente de la misma forma, solo que en este escenario entran en acción otros periféricos como el sensor, la cámara y la cerradura, los cuales acumulan información importante para después enviar una petición HTTP al servidor.

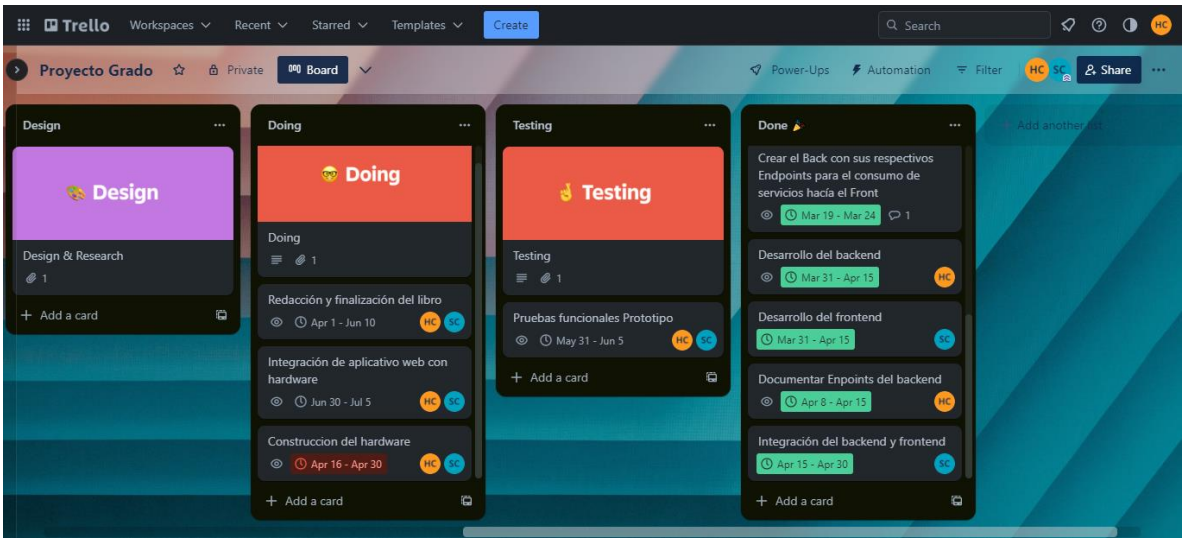
8. Prototipado

8.1. Software de gestión de actividades

Para una mejor planificación y gestión del trabajo se decidió usar Trello como herramienta colaborativa para la gestión de las actividades del proyecto, Trello permite llevar a cabo una mejor organización de las tareas por medio de los tableros donde se puede organizar estas actividades por estados como la metodología Kanban lo indica (Ver figura 25).

Figura 25

Trello - Tablero de gestión de actividades



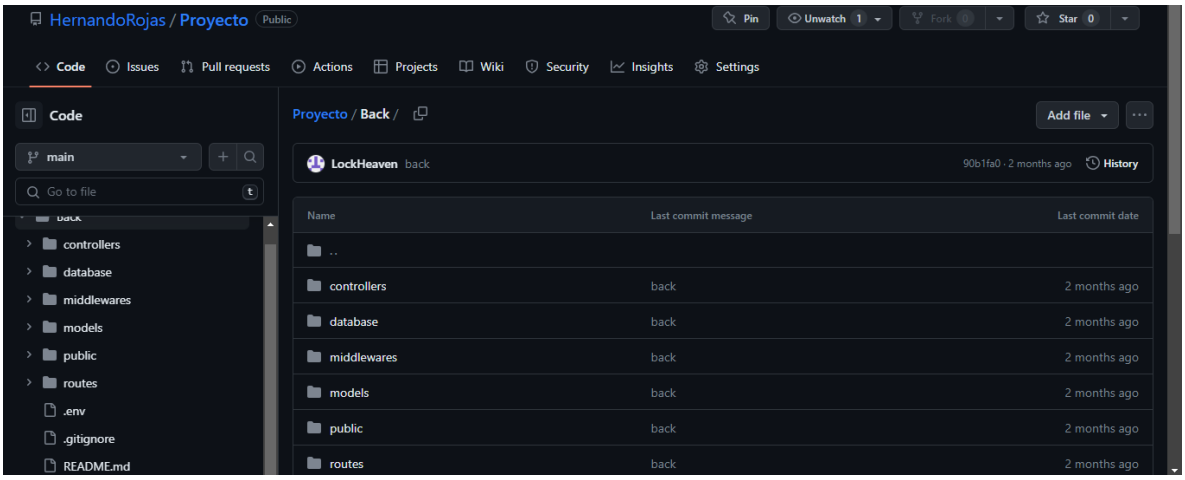
Nota. Tomado de Trello. https://trello.com/b/1t8FmfZ5/proyecto-grado.

8.2. Control de versiones y repositorio

Con el fin de llevar un orden, cronología, historial de cambios y control de versiones durante el desarrollo del aplicativo web se utilizó git y como repositorio en la nube se usó GitHub, además que usando GitHub se hacía más practico el trabajo colaborativo remoto para los dos desarrolladores del proyecto.

Figura 26

Repositorio GitHub



8.3. Prototipo 1

Se da inicio al desarrollo e implementación del prototipo funcional creando los componentes que deben integrar el sistema para funcionar, se inició con el desarrollo del aplicativo web, luego se implementó el modelo de reconocimiento facial para posteriormente integrar todas las partes y realizar el prototipo funcional.

8.3.1. Desarrollo del software

El desarrollo del software se realizó por partes independientes, inicialmente se creó el servidor sobre el que funcionaría el backend, luego se creó el modelo de la base de datos haciendo uso de Mongoose y la nube de Mongo para posteriormente hacer el desarrollo del frontend el cual debía consumir los servicios de la API.

8.3.1.1. Creación y levantamiento del servidor

Inicialmente se configuran las variables de entorno que servirán para configurar el servidor y definir las variables que serán accedidas por el entorno de ejecución.

También se levantó el servidor de manera local haciendo uso de express, para ello se importó la librería y se hizo la configuración del servidor, se establecieron las rutas para las peticiones http, las rutas para las funciones que sirven como intermedio para realizar validaciones en la API, la función para la conexión con la base de datos y la función para poner el servidor a la escucha por el puerto definido en la variable de entorno.

Figura 27

Función para levantar servidor web

```
listen(){
    //se define el puerto y se pone a la escucha el servidor
    this.app.listen(this.port, () => {
        console.log("Servidor corriendo en puerto", this.port)
    });
}
```

8.3.1.2. Desarrollo del backend

Se realizó la lógica del sistema usando una arquitectura REST API para exponer y transportar la información obtenida y suministrada a la base de datos, esta API podría ser accesible por la cerradura para el intercambio de datos como también para el aplicativo web.

8.3.1.2.1. Controlador

Se definen los controladores para cada tipo de solicitud HTTP para los casos que aplique, es desde estos que se hará el manejo de los datos por cada modelo y se hará la obtención, manipulación y envío de la información al cliente.

8.3.1.2.2. Modelo

Si bien la base de datos utilizada es MongoDB, es necesario crear los modelos estructurados en esquemas, esto se hace con el fin de establecer las restricciones y el tipado de datos para poder realizar validaciones y crear la conexión entre el backend desarrollado en node y la base de datos de Mongo, para esta representación de esquemas se usó la librería Mongoose la cual es útil para optimizar estas operaciones.

8.3.1.2.3. Helpers

Se crearon una serie de ‘helpers’ o funciones que están encargadas de brindar un nivel de seguridad, estos helpers sirven de apoyo con la validación de los datos antes de ser enviados a la base de datos. También se creó un helper desde el cual se creará el JSON Web Token que servirá para la autenticación del usuario.

8.3.1.2.4. Middlewares

Se definen unos middlewares, los cuales también son funciones intermedias que pueden ser reutilizadas por diferentes controladores, los middlewares creados sirven para validar que el JSON web token que se esta recibiendo como parámetro sea válido y corresponda a un usuario registrado, también se define middleware para validar roles cuando se esta accediendo a ciertas funcionalidades lo que sirve para agregar ciertas restricciones y brindar una mayor seguridad al acceso a la información.

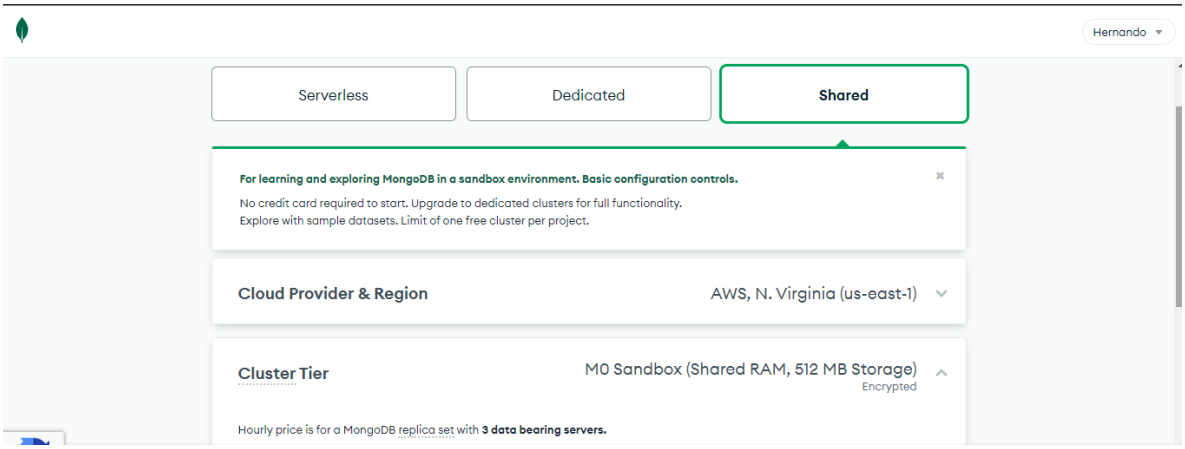
8.3.1.3. Creación de la base de datos

Para el alojamiento de la base de datos se utilizó MongoDB Atlass el cual es un servicio de alojamiento de base de datos en la nube. Para efectos del desarrollo y creación del prototipo funcional se hace uso del servicio de manera gratuita, la creación de una base de datos de esta

escala es gratuito y fácil, inicialmente es necesario registrarse en la plataforma, luego es necesario crear un lugar donde se alojará la base de datos llamada cluster donde se deben configurar los parámetros bajo los que funcionará la base de datos como lugar de del servidor y tipo de servidor, capacidad de almacenamiento, entre otros (Ver figura 28).

Figura 28

Configuraciones en MongoDB Cloud



Nota. Configuraciones iniciales de MongoDB Cloud, Tomado de MongoDB Cloud. https://cloud.mongodb.com/v2/64140881c857de5457a59dff#/clusters/edit/Cluster0

El paso a seguir una vez se tiene creado el cluster es conectar el backend a la base de datos en la nube para posteriormente alojar los datos en esta, para este proceso se hizo uso de la librería de Mongoose la cual nos permite conectar la base de datos para el envío de peticiones.

8.3.1.4. Desarrollo del frontend.

Se realiza el componente visual del aplicativo desde el cual el usuario consumirá los recursos del sistema, como se mencionó anteriormente para el desarrollo del frontend se utilizaron los lenguajes HTML, CSS y Angular como framework web de Javascript.

Para un desarrollo más ágil y flexible se usó Angular Material, la cual es una librería de componentes desarrollada por Google que permite agregar a la vista diferentes componentes como botones, campos de entrada, entre otros, además de utilizar diseños basados en cuadrículas, animaciones y transiciones receptivas.

Para las gráficas para la visualización de los datos se utilizó HighCharts la cual es una biblioteca de gráficos interactivos y proporciona las gráficas que se implementaron en las vistas del frontend de la aplicación.

Figura 29

Frontend - Vista del Login

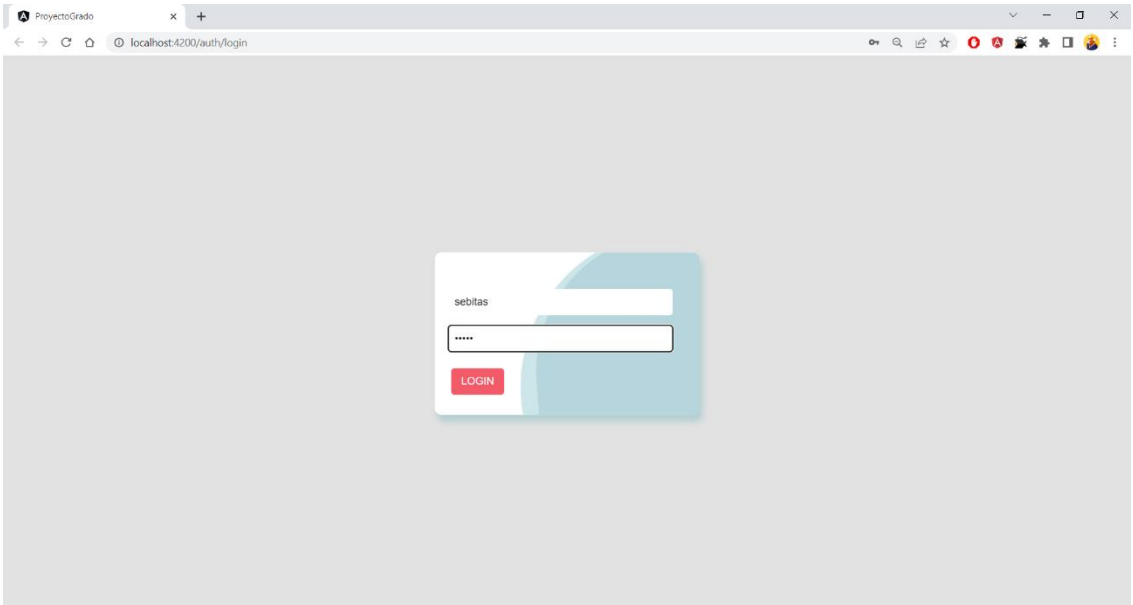


Figura 30

Frontend - Vista del dashboard de todos los usuarios

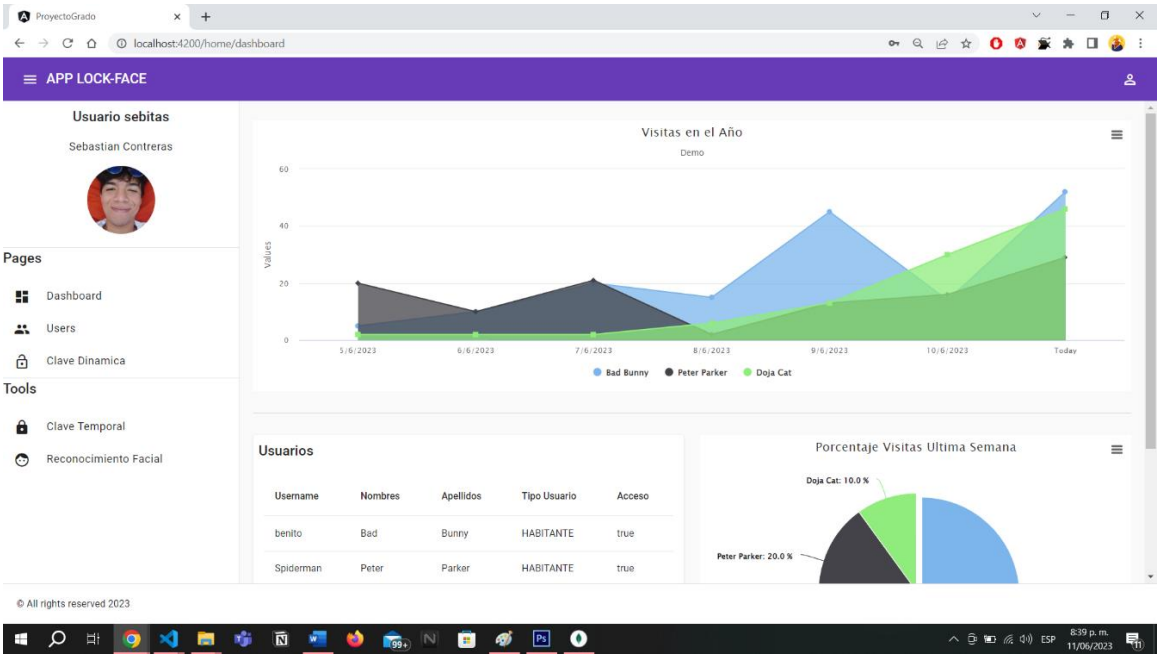


Figura 31

Frontend - Vista del dashboard de todos los usuarios

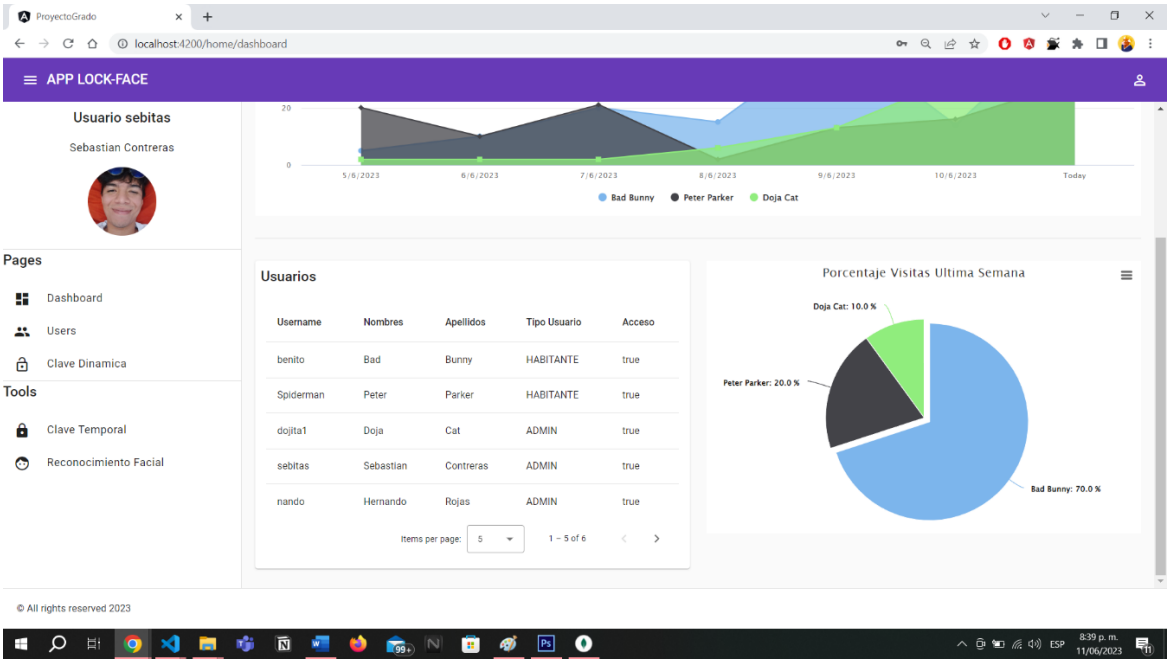


Figura 32

Frontend - Vista de usuarios registrados

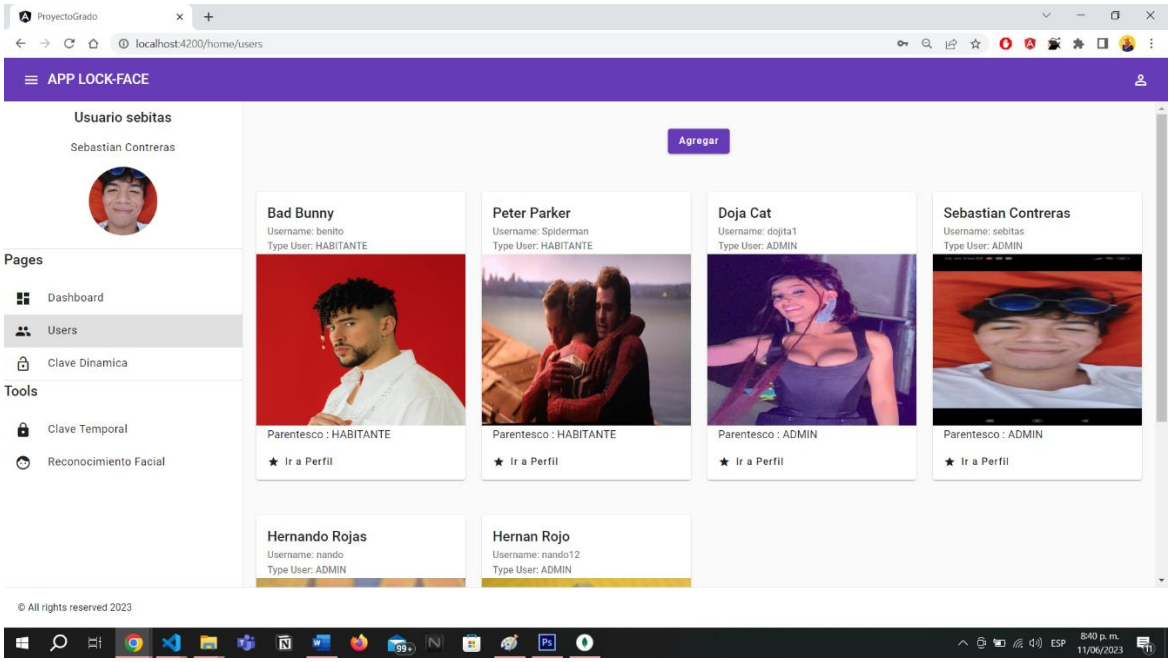


Figura 33

Frontend - Formulario de registro de usuarios

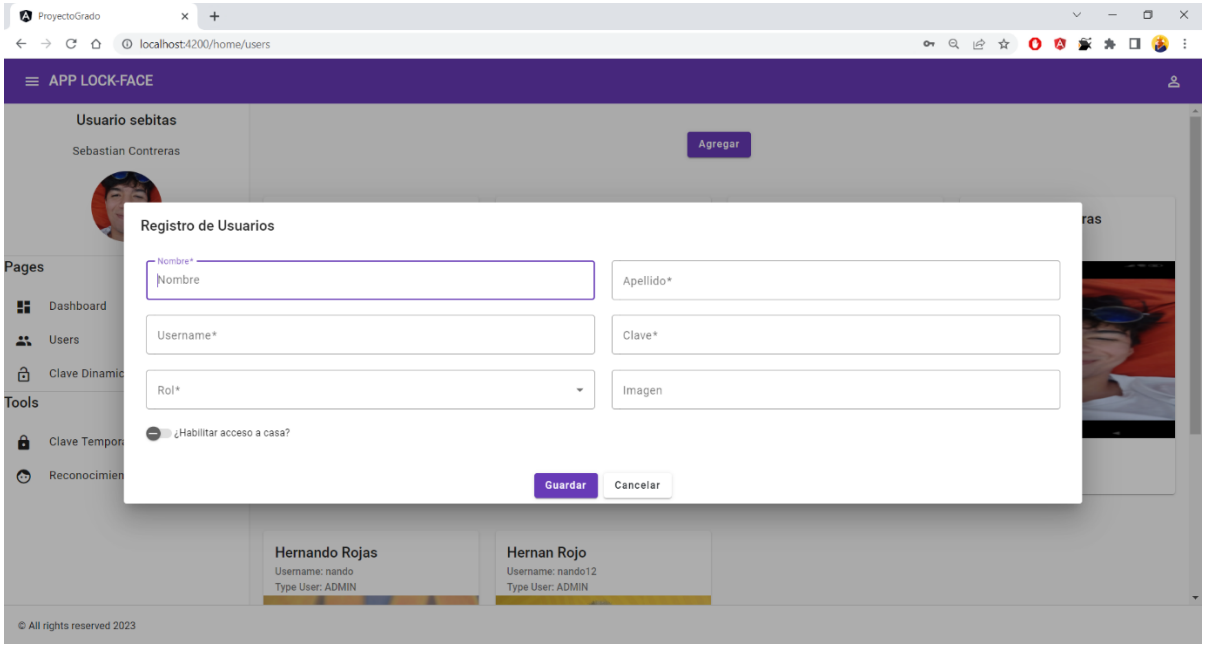
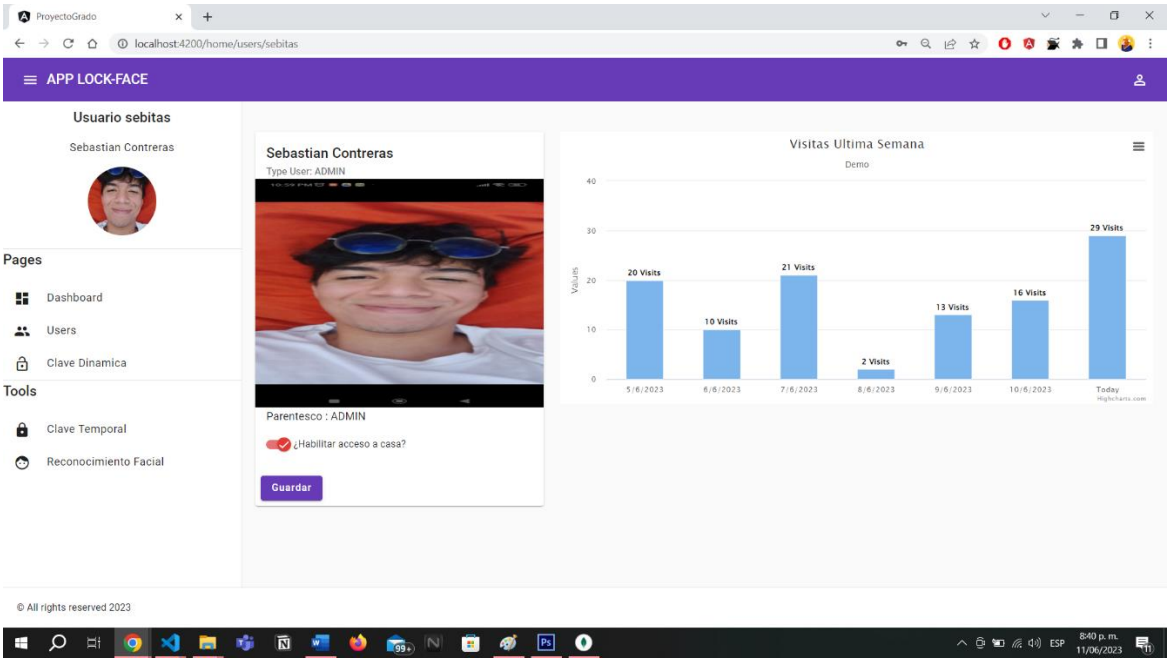


Figura 34

Frontend - Vista de información personal



8.3.2. Implementación del modelo de reconocimiento facial

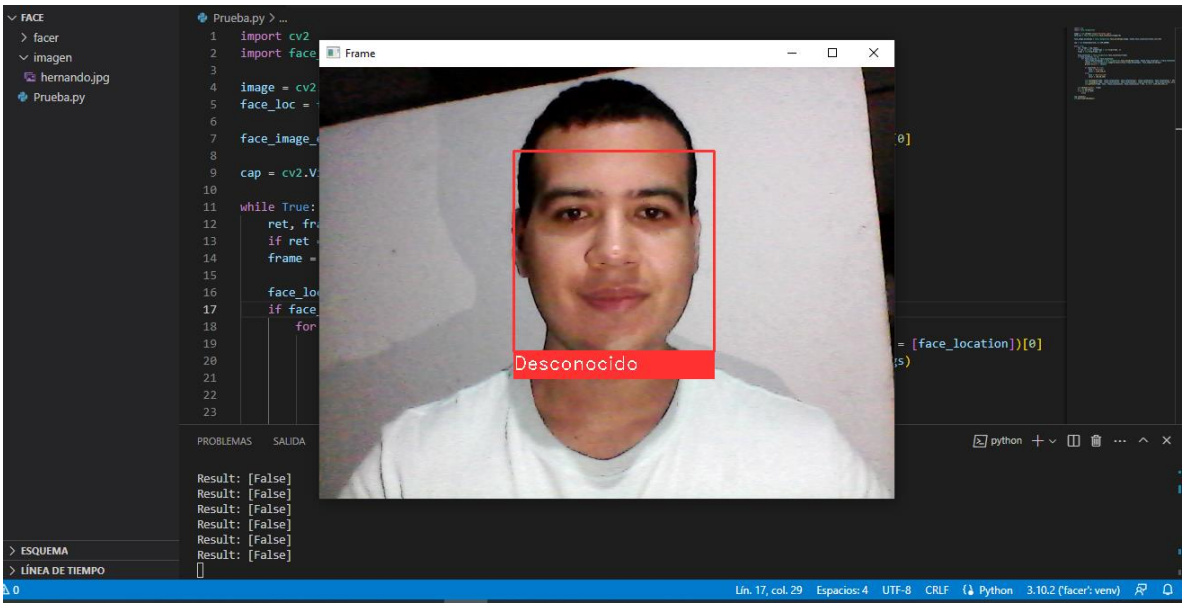
Antes de implementar el modelo de conocimiento facial en el hardware se hizo la implementación de manera local en el computador, se realizó la instalación de la última versión de Python en el computador, posteriormente se creó un proyecto de Python en el cual se hizo la importación de la librería face recognition que forma parte a su vez de la librería OpenCV implementando el modelo de reconocimiento facial con las herramientas que brinda esta, al ser un modelo que se corre de manera local y es como se espera que funcione en la Jetson Nano, el vector de características que se extrae de cada rostro se almacena en la memoria local de la Jetson.

Una vez implementado el modelo se hicieron pruebas con uno de los autores del proyecto (Hernando), inicialmente se cargó una foto del segundo autor (Sebastian) en una carpeta donde se ubicaba el proyecto, foto con la cual se realizó la comparación para verificar si es la misma

persona que está frente a la cámara, en este caso el mensaje que debía salir en la pantalla es ‘desconocido’ con el cuadro en rojo ya que la persona frente a la cámara no era la misma de la foto que se había almacenado, además, en la consola se vería que el valor de la variable ‘Result’ es ‘False’, lo que significa que se validó y no se reconoció como un usuario registrado en el sistema (Ver figura 35).

Figura 35

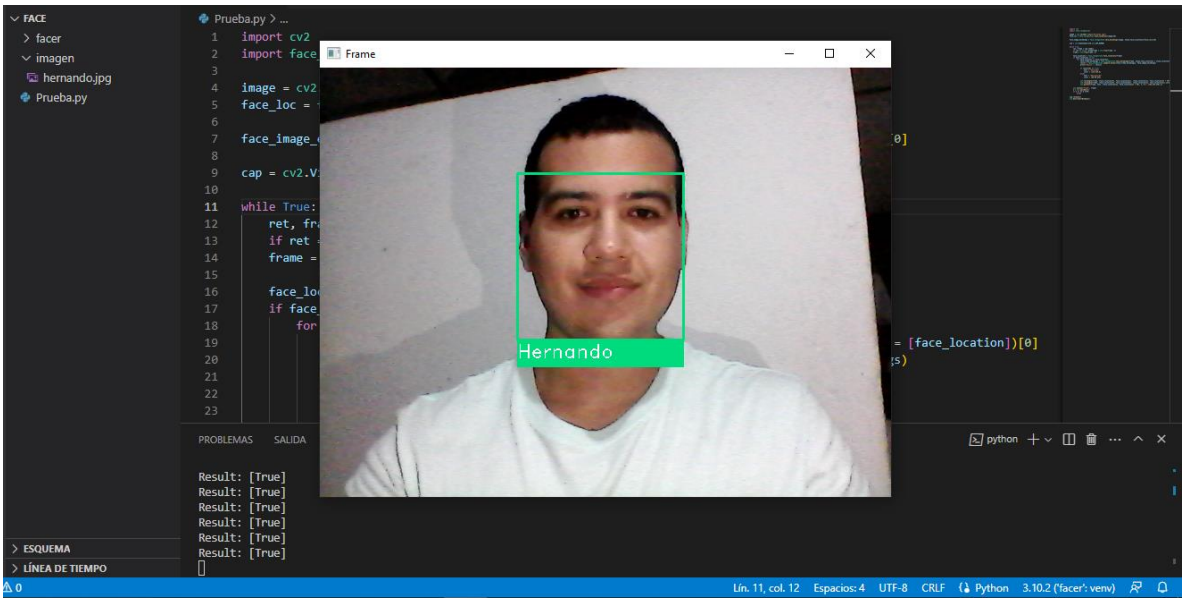
Reconocimiento facial con usuario desconocido en el sistema



Posteriormente se realizó otra prueba, pero esta vez se hizo cargando una foto que si correspondía a la misma persona que estaba frente a la cámara, en este caso el mensaje que debía aparecer en pantalla es el nombre ‘Hernando’ con el cuadro de la cámara en verde, además, en la consola se vería que el valor de la variable ‘Result’ es ‘True’, haciendo referencia a que si lo reconoció como una persona registrada en el sistema (Ver figura 36).

Figura 36

Reconomiciento facial con usuario registrado



8.3.2.1. Mejora al modelo de reconocimiento facial

Si bien el modelo de reconocimiento facial funciona correctamente, se identificó una falencia en concreto, se observaba que al poner una foto de la persona que se estaba intentando validar la identidad el algoritmo lo reconocía exitosamente (Ver figura 37), siendo esto una vulnerabilidad de seguridad ya que cualquier persona que tenga una foto de un habitante del hogar podría ingresar.

Como solución a este problema, se implementó un algoritmo que solicita al usuario realizar una cantidad de parpadeos determinada, donde el algoritmo hace el conteo y si es el caso que se reconoce al usuario como uno registrado en el sistema, autoriza el acceso (Ver figura 38).

Figura 37

Vulnerabilidad del modelo de reconocimiento facial

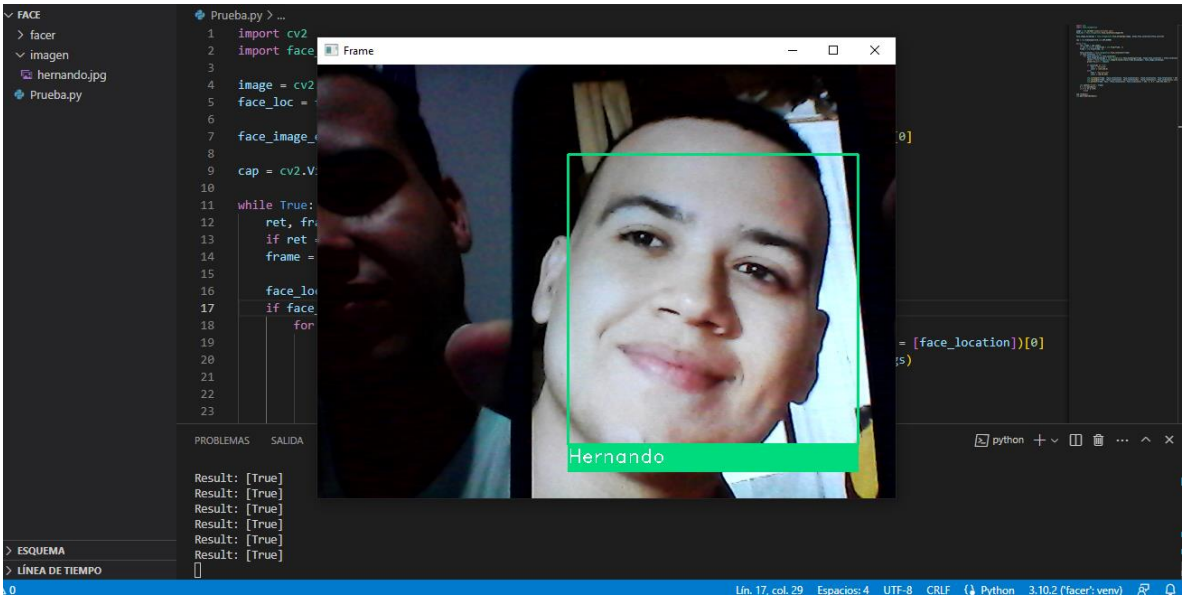
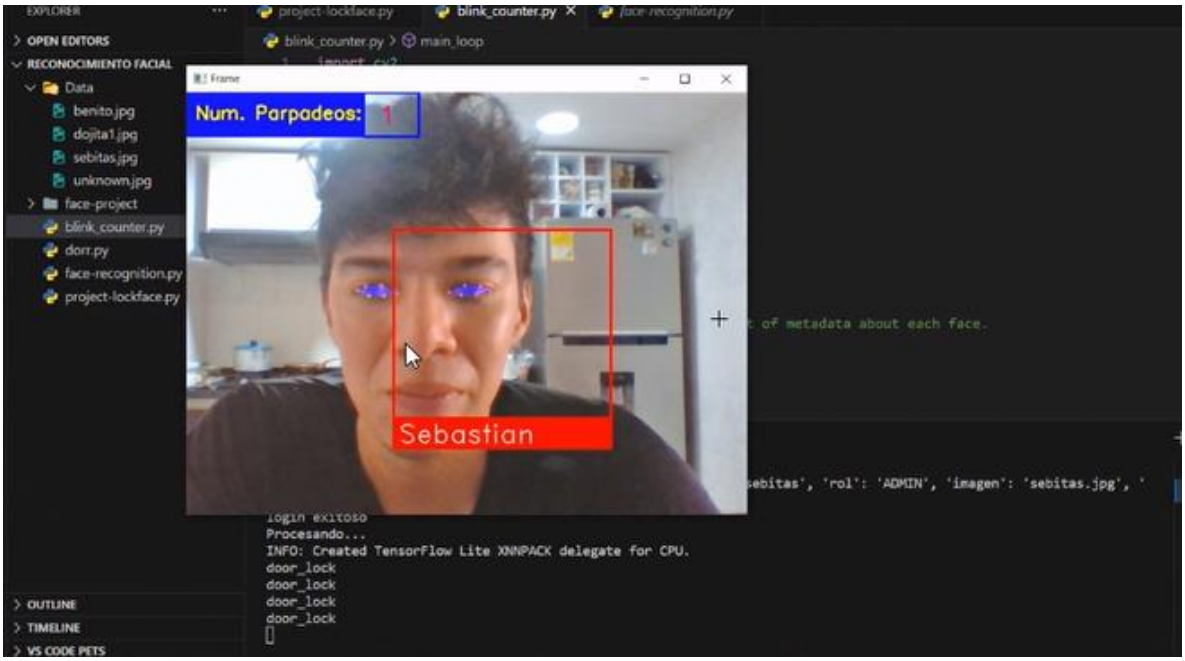


Figura 38

Mejora de parpadeos al modelo de reconocimiento facial



8.3.3. Configuración del hardware

Como primer paso para la configuración del hardware se utilizó una MicroSD que fue configurada con la imagen del sistema operativo el cual fue descargado de la página oficial de NVIDIA que ofrece el kit de desarrollo de la Jetson Nano, este kit llamado JetPack contiene CUDA que es una plataforma de computación en parelelo y será de vital importancia para la activación y correcto funcionamiento de la GPU de la Jetson.

Una vez se ha configurado la MicroSD se conecta a la Jetson, luego esta se conecta a la electricidad y luego se conecta a una pantalla para realizar las configuraciones iniciales, y posteriormente implementar el modelo de reconocimiento facial (Ver figura 39).

Figura 39

Jetson Nano conectada a electricidad y video

Figura 40

Imagen de salida de video de la Jetson Nano

Los pasos siguientes son correr el algoritmo con el modelo de reconocimiento facial y configurar la jetson para que realice solicitudes HTTP al servidor, sin embargo, al momento de ejecutar el modelo de reconocimiento facial se observa que no se activa el frame sobre el cual se visualiza la persona frente a la cámara, se logra identificar que esto se debe a que el JetPack instalado en la jetson no contiene la versión de OpenCV sobre la cual se ejecuta CUDA para la jetson nano, motivo por el cual es necesario realizar una re instalación de la versión de OpenCV que se requiere para correr el modelo haciendo uso de CUDA.

8.4. Análisis de prototipo 1

Si bien se ha logrado desarrollar la aplicación web y el modelo de reconocimiento facial exitosamente, se tiene el mayor inconveniente el cual impide la implementación del prototipo

funcional, al momento de instalar la versión de OpenCV sobre la que se ejecuta CUDA se visualiza un error en consola que indica errores en componentes de C++ por lo que se hace una búsqueda exhaustiva del origen del error que culmina sin éxito.

Además, como se mencionó anteriormente el sistema operativo de la Jetson Nano corre sobre una microSD donde la lectura y escritura de datos no es la más veloz haciendo que una instalación de una librería del estilo de OpenCV tarde horas en realizarse, esto sumado a que una vez se ha solucionado un error puede surgir otro que requiere de una nueva instalación lo cual podría convertirse en una serie de errores en cadena que podría tardar una cantidad elevada de horas en solo instalación de componentes, librería u otros, por lo tanto, se opta por buscar una alternativa para la implementación del modelo de reconocimiento facial cuya configuración sea más rápida y simple, donde incluso la infraestructura hardware sea de menor costo.

Se realiza un análisis del estado del sistema con el fin de identificar qué se cumplió con el prototipo respecto a los requerimientos funcionales propuestos obteniendo lo siguiente:

Tabla 6

Prototipo 1 - Cumplimiento de requerimientos funcionales del aplicativo web

N° Requerimiento	Descripción	Estado
RFW-1	El aplicativo web debe ser capaz de autenticar y autorizar el acceso del usuario.	Cumplido
RFW-2	Se debe poder crear un usuario con sus respectivos datos.	Cumplido
RFW-3	Se debe poder actualizar (cambiar) el estado de usuario en el portal web.	Cumplido
RFW-4	Se debe poder eliminar usuarios.	Cumplido
RFW-5	Se deben poder visualizar una vista general de los usuarios con sus datos.	Cumplido

RFW-6	Se debe poder visualizar una gráfica con los ingresos de un usuario específico en un lapso determinado.	Cumplido
RFW-7	Se debe poder visualizar una gráfica con el total de ingresos de todos los usuarios en un lapso determinado.	Cumplido

Tabla 7

Prototipo 1 - Cumplimiento de requerimientos funcionales del sistema

N° Requerimiento	Descripción	Estado
RFS-1	El sistema debe ser capaz de procesar los datos como la imagen, nickname y el estado del usuario.	No cumplido
RFS-2	El sistema debe ser capaz de detectar e identificar la cara del usuario registrado.	No cumplido
RFS-3	El sistema debe ser capaz de autenticar y autorizar el acceso del usuario al hogar.	No cumplido
RFS-4	El sistema debe ser capaz de captar datos de ingreso una vez se conceda el acceso al hogar.	No cumplido

Se observa que no se logró implementar exitosamente el prototipo, motivo por el cual se decide realizar una nueva iteración desde la fase análisis del sistema con el fin de identificar los cambios que podría sufrir el prototipo como consecuencia del inconveniente presentado.

8.5. Prototipo 2

8.5.1. Análisis y rediseño

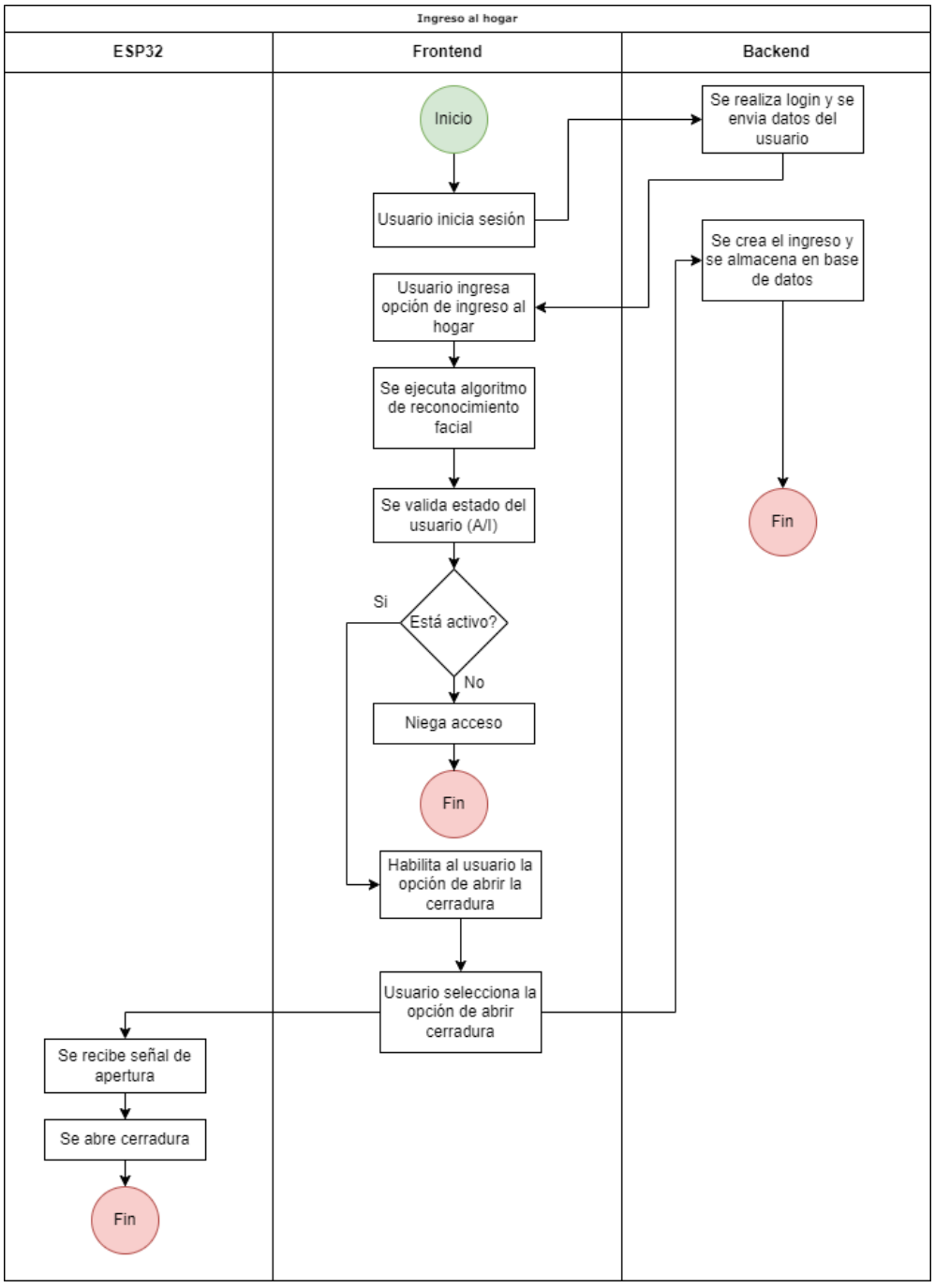
8.5.1.1. Análisis de requerimientos

Para el prototipo 2 los requerimientos del sistema en general no cambian, por lo tanto, en cuanto a procesos se redefine únicamente el diagrama de actividades para el proceso principal del ingreso al hogar.

8.5.1.1.1. Diagrama de actividades

Figura 41

Prototipo 2 - Diagrama de actividades para ingreso al hogar



8.5.1.2. Selección de tecnologías

8.5.1.2.1. Modelo de reconocimiento facial

Previamente en la comparación de modelos de reconocimiento facial se mencionaban librerías del lenguaje Python, sin embargo, se pudo observar que en la implementación anterior estaba presentando problemas al hacerse con la librería face recognition, por lo tanto, se optó por utilizar la librería face api del lenguaje Javascript ya que es el mismo lenguaje que se está utilizando en el frontend y en el backend, lo que podría facilitar la implementación.

8.5.1.2.2. Computadora móvil

Como se observaba en las comparativas de las computadoras móviles, todas tenían un precio considerablemente elevado, además que estas tecnologías se tenían como opción porque inicialmente se pensaba implementar el modelo de reconocimiento facial en la computadora móvil, pero teniendo en cuenta que la librería face api puede ser implementada en el frontend de la aplicación se opta por elegir una computadora que únicamente tenga la función de recibir las solicitudes de apertura y cierre y se encargue de hacer esta labor, además que al ser una computadora con mejor capacidad de cómputo su costo es considerablemente inferior.

Para la implementación de la cerradura se propone un esquema sencillo basado en 3 componentes, modulo relé (Ver figura 42) el cual opera a 5V, 1 canal de relay, señal de control TTL de 3.3 o 5V y tiempo de acción de entre 5 y 10 ms. También se tiene un ESP32 (Ver figura 8) y la cerradura solenoide (Ver figura 9).

Figura 42

Módulo Relé

Nota. Modulo Relé, Tomado de ssdielect, 2023. https://ssdielect.com/rele-relay-o-relevo/992-md-rele-1ch.html

8.5.1.3. Diseño del mockup

Se hace el diseño de una vista adicional, la cual estará en la página del usuario y desde esta se hará el proceso del reconocimiento facial, se tendrá un frame en la izquierda desde el cual se verá la imagen que está capturando la cámara en tiempo real, y en el frame de la derecha se verá la foto que el usuario ha decidido usar para el reconocimiento facial.

Figura 43

Prototipo 2 - Mockup de vista de reconocimiento facial

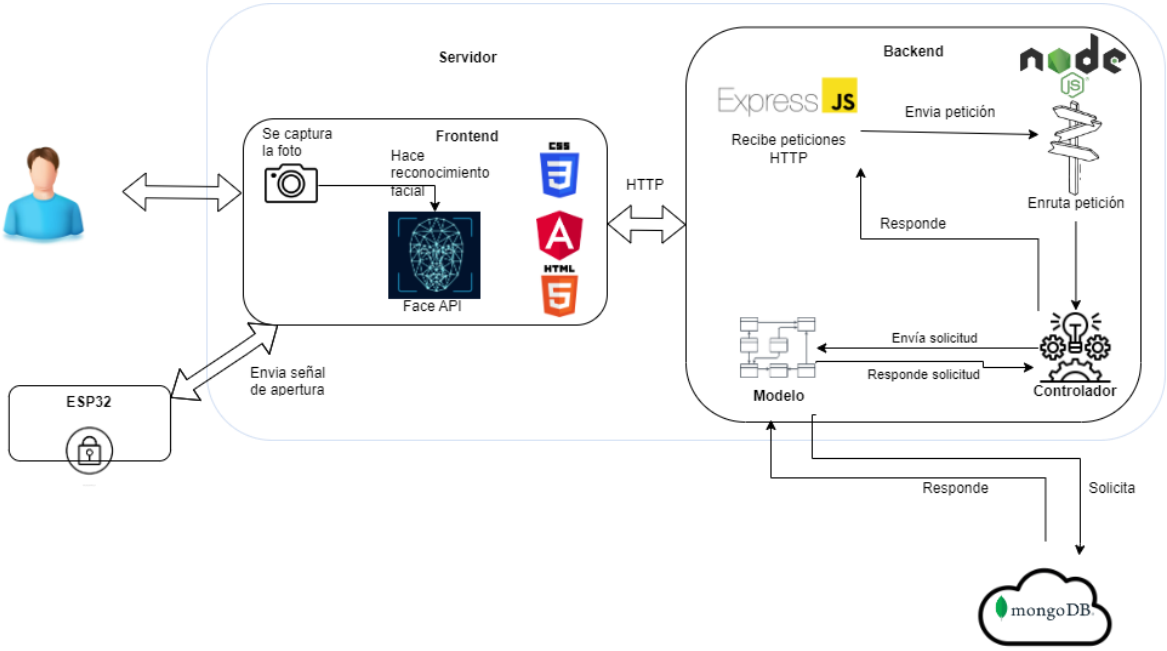
Nota. https://vasanza.blogspot.com/2021/07/especificaciones-del-modulo-esp32.html

8.5.1.4. Diseño de la arquitectura

Se reestructura la arquitectura del sistema donde se incluye el cambio en el hardware y la implementación del reconocimiento facial en el frontend.

Figura 44

Prototipo 2 - Arquitectura del sistema



8.5.2. Software del prototipo 2

Para el prototipo 2 no se hacen modificaciones en el backend, servidor y base de datos, solamente se agrega una nueva vista que es donde el usuario ingresa a la opción del reconocimiento facial para el ingreso al hogar.

8.5.2.1. Frontend

Se implementó la vista del reconocimiento facial, donde inicialmente se captura la foto del usuario (Ver figura 45), luego se inicial el reconocimiento facial donde el sistema hace el cálculo de las similitudes de la imagen capturada y la almacenada del usuario en la base de datos para finalmente dar el resultado del cálculo y habilitar la opción de abrir la cerradura (Ver figura 46).

Figura 45

Prototipo 2 - Captura de foto desde frontend

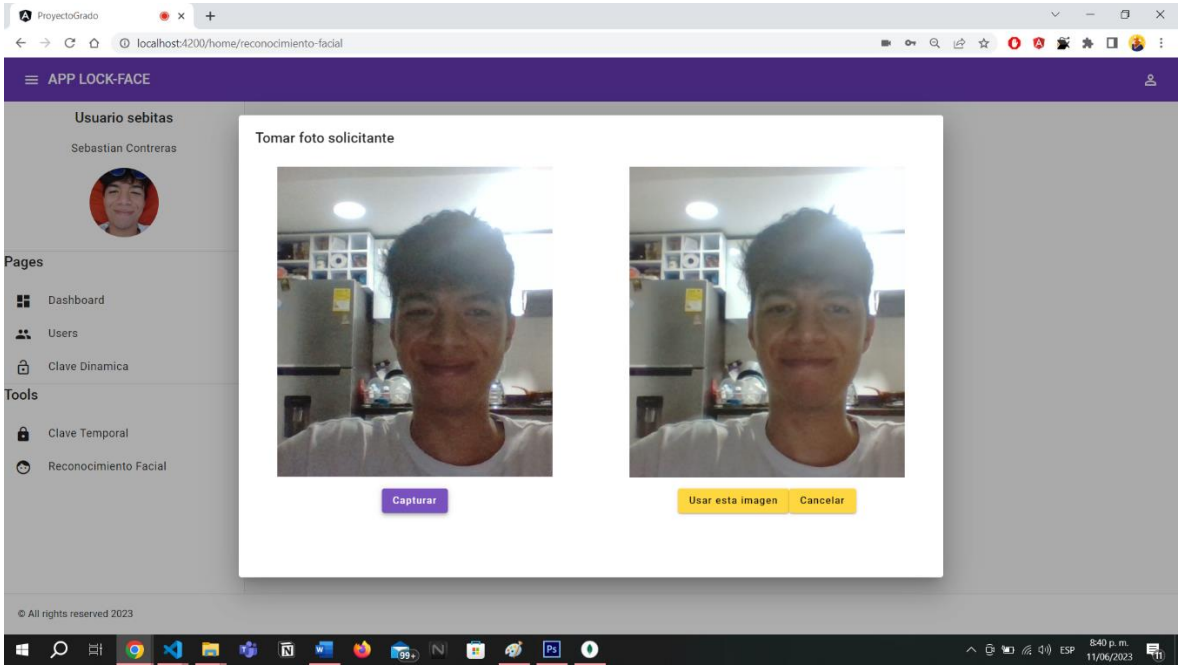
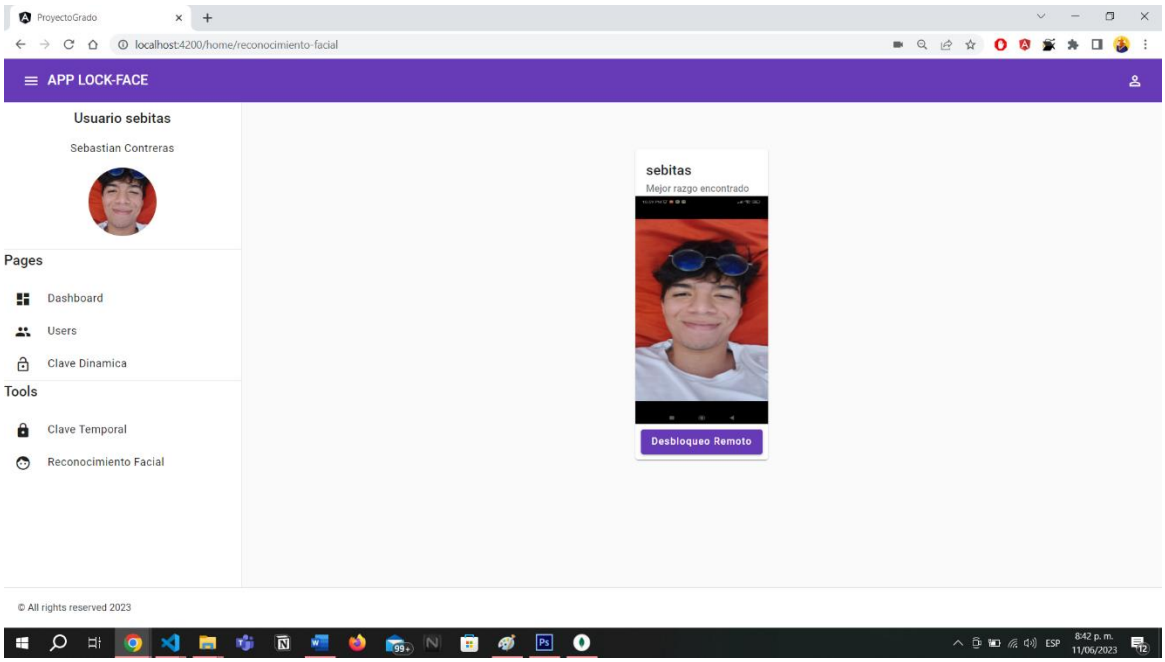


Figura 46

Prototipo 2 - Reconocimiento facial éxitoso



8.5.3. Implementación del modelo de reconocimiento facial en el frontend

Para la implementación del reconocimiento facial se instala la librería “face-api” desde el entorno en el que estemos trabajando, en este caso es Angular y la instalamos desde la línea de comandos. Una vez instalada la importamos como una librería en el entorno en el que queramos trabajar como se muestra en la siguiente imagen.

Figura 47

Importe de librería face api

```
import * as faceapi from 'face-api.js';
```

Luego de haberse importado la librería ya podemos cargar los modelos pre entrenados necesarios para el reconocimiento facial y la detección de rostros que vamos a utilizar en nuestro proyecto, en este caso cargaremos tres modelos que descargaremos desde el repositorio oficial, los cuales son: tinyFaceDetector, faceLandmark68Net y faceRecognitionNet (los tres modelos mencionados fueron explicados en cómo funciona la librería face-api).

Figura 48

Cargue de modelos de reconocimiento facial

```
await faceapi.nets.tinyFaceDetector.loadFromUri('/assets/models');
await faceapi.nets.faceLandmark68Net.loadFromUri('/assets/models');
await faceapi.nets.faceRecognitionNet.loadFromUri('/assets/models');
```

Seguido de haber cargado los modelos ya podremos hacer uso de las funciones que tiene face-api para realizar tareas como detección de rostros, reconocimiento facial, etc.

Ahora el usuario podrá proporcionar una imagen desde la cámara donde este corriendo el software para poder hacer el correcto reconocimiento facial. Para ello haremos uso de la función “face matcher” de face-api que permite comparar las características faciales de dos rostros para determinar si pertenecen a la misma persona.

Los pasos por seguir del procedimiento de reconocimiento facial son los siguientes:

1. Detección y extracción de características: Antes de poder usar el face matcher, primero necesitamos detectar y extraer las características faciales de los rostros que deseemos comparar.
2. Comparación de características: Una vez que tengamos las características faciales de los rostros, podemos compararlas utilizando el face matcher.
3. Resultado de la comparación: El resultado de la comparación es un objeto que contiene información sobre la mejor coincidencia encontrada. Podemos acceder a la etiqueta y el umbral de la mejor coincidencia.

8.5.4. Implementación del hardware e integración con aplicación web

8.5.4.1. Construcción del hardware

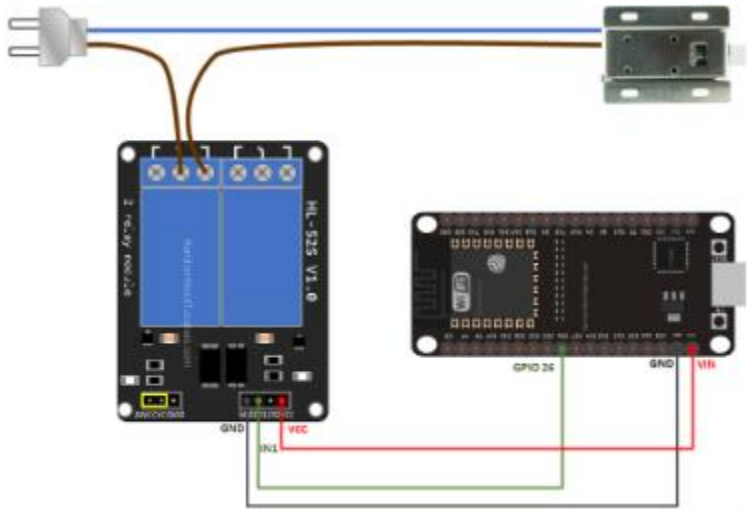
Para la implementación del hardware de la cerradura se hizo la conexión física del relé, la ESP32 y la cerradura solenoide.

La ESP32 enviará una señal desde el pin D26 (En estado OUTPUT) al relé para avisar que pueda abrir el relé y dar paso al flujo de corriente. También se conectarán los pines de voltaje y tierra al relé para el correcto funcionamiento.

En cuanto al relé, se conecta en forma de “normalmente cerrado”, para cuando enviemos la señal abramos la compuerta y demos flujo a la corriente para que se pueda abrir la cerradura correctamente y podamos volver a cerrarla luego de cierto tiempo abierta.

Figura 49

Prototipo 2 - Arquitectura hardware



8.5.4.2. Conexión del hardware con el software

Para hacer la conexión del hardware con el software se utilizó la biblioteca de Arduino “ESPAsyncWebServer”, con la cual podemos crear un servidor web asíncrono que implementaremos en la ESP32.

Los pasos por seguir para la correcta implementación son los siguientes:

1. Configuración del servidor web: incluiremos la biblioteca en el proyecto y crearemos una instancia del servidor web.
2. Manejo de rutas y solicitudes: esto implica asociar una función o un controlador a una ruta específica. Cuando se recibe una solicitud en esa ruta, el servidor ejecutará la

función o el controlador correspondiente. En nuestro caso crearemos una ruta que para poder mandar una señal de salida desde el pin D26.

3. Ejecución del servidor: se debe iniciar el servidor web para que comience a escuchar y procesar las solicitudes. Aquí se destaca que el servidor correrá en la misma conexión Wifi local que este configurada en el ESP32, por lo cual desde el Front es importante conectarse a la red Wifi que este conectada el ESP32.

Envío de peticiones desde el Front: finalmente desde el front creando un servicio importaremos la librería “HttpClient” que es propia de Angular, con ella podemos hacer peticiones Http a cualquier URL. Con esta implementación podremos enviar la petición Http desde la URL que nos proporciona la ESP32 para abrir la cerradura y dar acceso al hogar.

Figura 50

Implementación de biblioteca ESPAsyncWebServer

```
#include <ESPAsyncWebServer.h>

AsyncWebServer server(80);

void handleRootRequest(AsyncWebServerRequest *request) {
  request->send(200, "text/plain", "¡Hola, mundo!");
}

void setup() {
  server.on("/", HTTP_GET, handleRootRequest);
  server.begin();
}

void loop() {

}
```

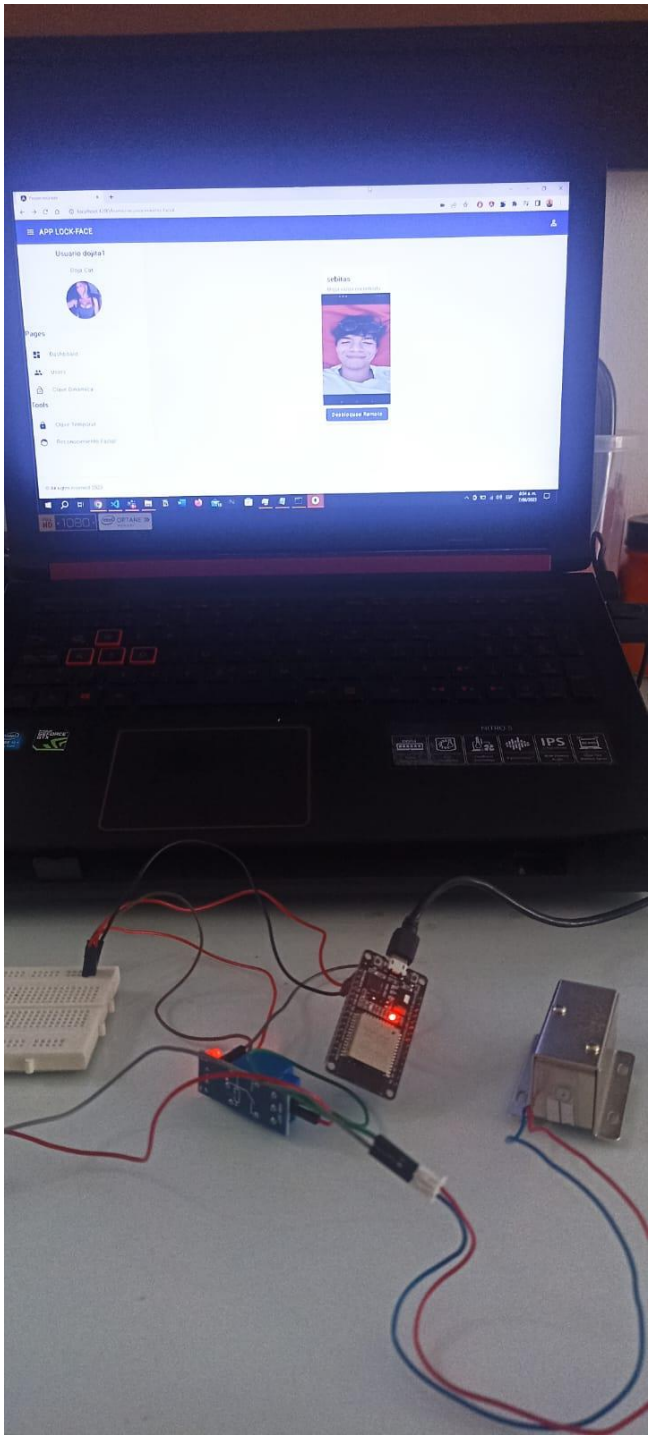
8.5.5. Prototipo funcional final

Una vez se ha finalizado con la configuración del hardware y el software se procede a levantar los servidores y realizar una prueba con el fin de validar el funcionamiento del prototipo, en la figura # se puede observar el hardware conectado a la electricidad y a la espera de recibir

una petición desde el servidor web, además se visualiza en el computador que se está en la vista del aplicativo web.

Figura 51

Prototipo Funcional Final



9. Presentación de resultados

9.1. Pruebas al prototipo

Con el fin de evaluar la eficacia del prototipo se realizan pruebas al prototipo funcional final levantando el servidor en la red local de uno de los autores, se hace una prueba creando un nuevo usuario, cargando la imagen de este, y luego haciendo el proceso para el ingreso al hogar bajo diferentes condiciones.

En la siguiente tabla de presenta el resultado final de las pruebas las cuales fueron realizadas con el mismo usuario registrado.

La finalidad de esta prueba era evaluar el comportamiento del prototipo cuando se intentaba hacer la autenticación del usuario haciendo uso de desbloqueo por reconocimiento facial, se evidencia que en 3 de los cuatro escenarios no se concede el acceso y en un solo caso sí, lo cual hace ver que el algoritmo es estricto en cuanto a las alteraciones que pueda tener el rostro del usuario.

Tabla 8

Resultado de las pruebas de reconocimiento facial

Número de prueba	Descripción	Concede Acceso (Si/No)
1	Prueba con usuario usando un objeto en el rostro (Gafas).	No
2	Prueba con usuario forzando una expresión facial diferente a la de la foto almacenada en el registro.	No
3	Prueba con usuario en condiciones de baja exposición a la luz.	No

4	Prueba con usuario con presentación similar a la de la foto almacenada en el registro.	Si

Se tiene que al usar un elemento como lo son las gafas limitan la capacidad del algoritmo e impide que el reconocimiento se haga de manera exitosa (Ver figuras 52 y 53).

Figura 52

Prototipo funcional - prueba reconocimiento con gafas

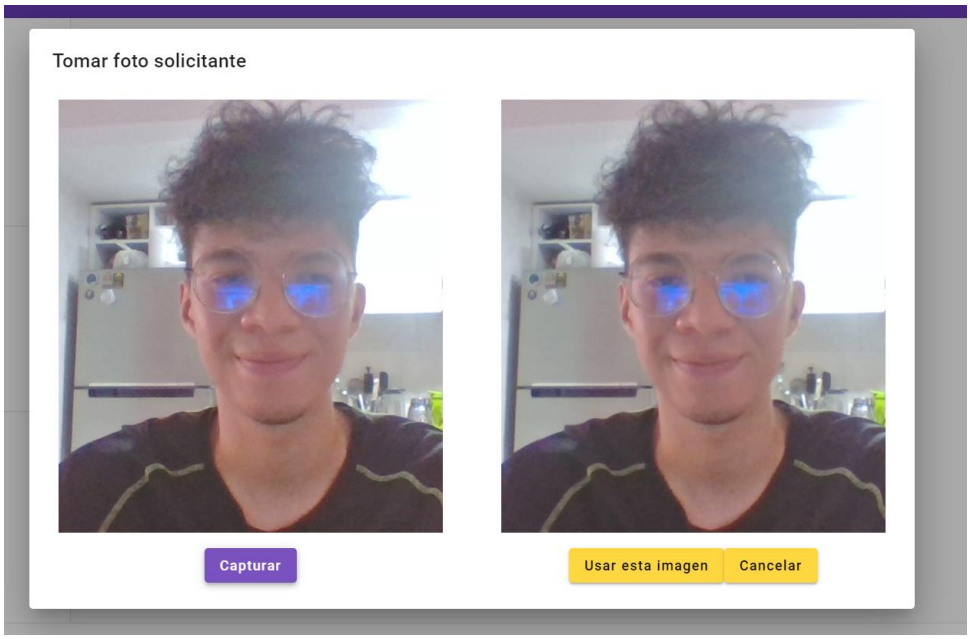
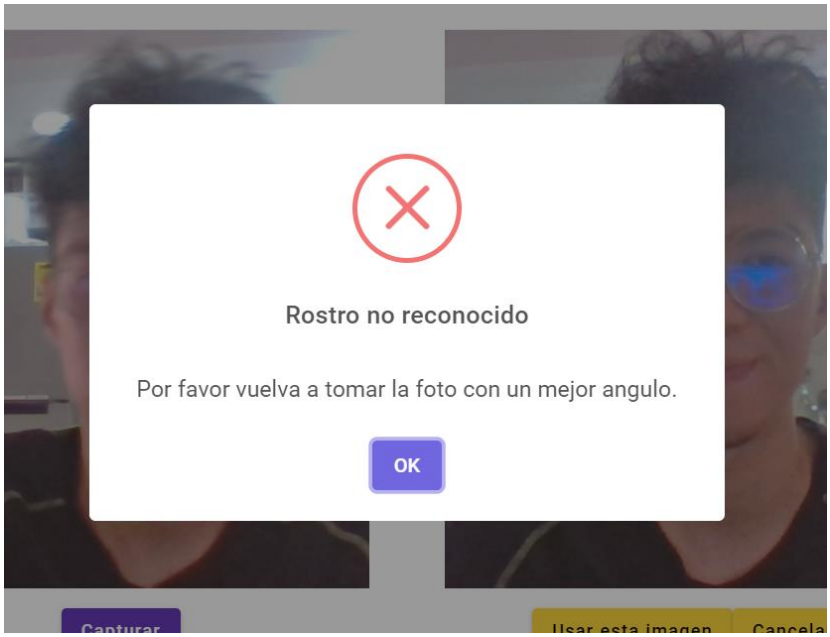


Figura 53

Prototipo funcional - prueba reconocimiento con gafas fallido



Por otra parte, al hacer una nueva validación, pero esta vez modificando los rasgos corporales como las expresiones faciales, se observa que el modelo obtiene los mismos resultados que el caso anterior (Ver figuras 54 y 55).

Figura 54

Prototipo funcional - prueba reconocimiento con cambio en expresión facial

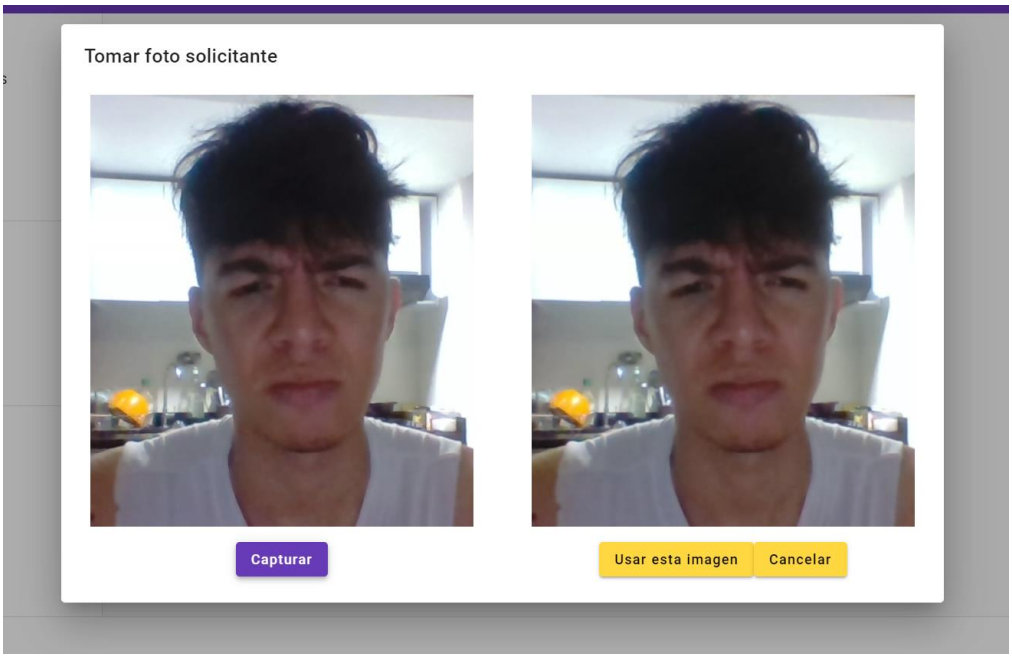
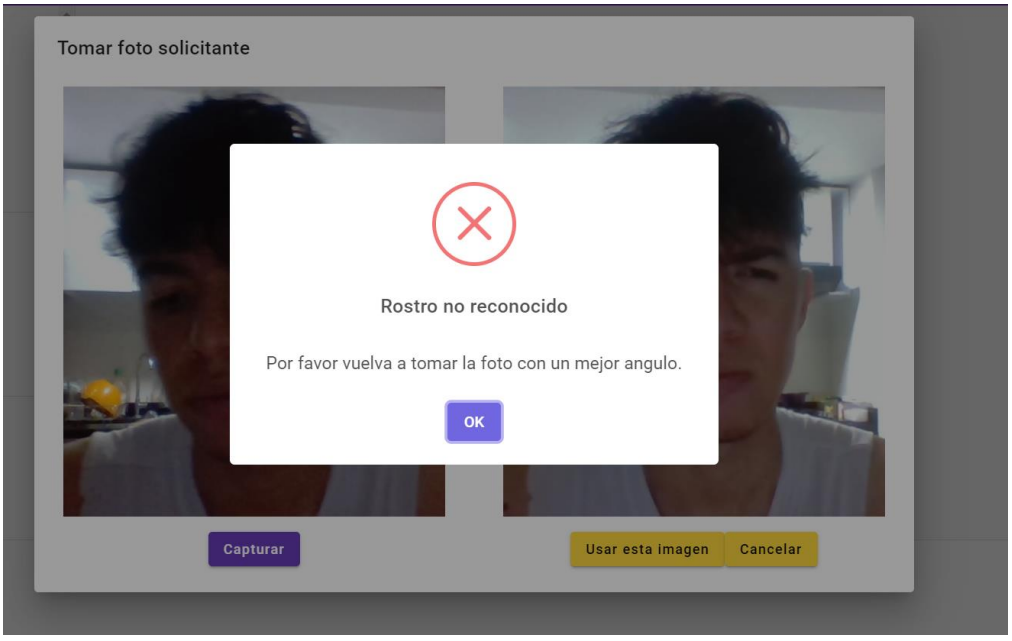


Figura 55

Prototipo funcional - prueba reconocimiento facial con cambio en expresión facial fallido



Se hace una tercera prueba donde se intenta hacer el reconocimiento facial en un lugar donde hay condiciones de poca luz, se observa que en este caso el modelo tampoco es capaz de reconocer al usuario que está intentando autenticarse (Ver figuras 56 y 57).

Figura 56

Prototipo funcional - prueba reconocimiento con poca luz

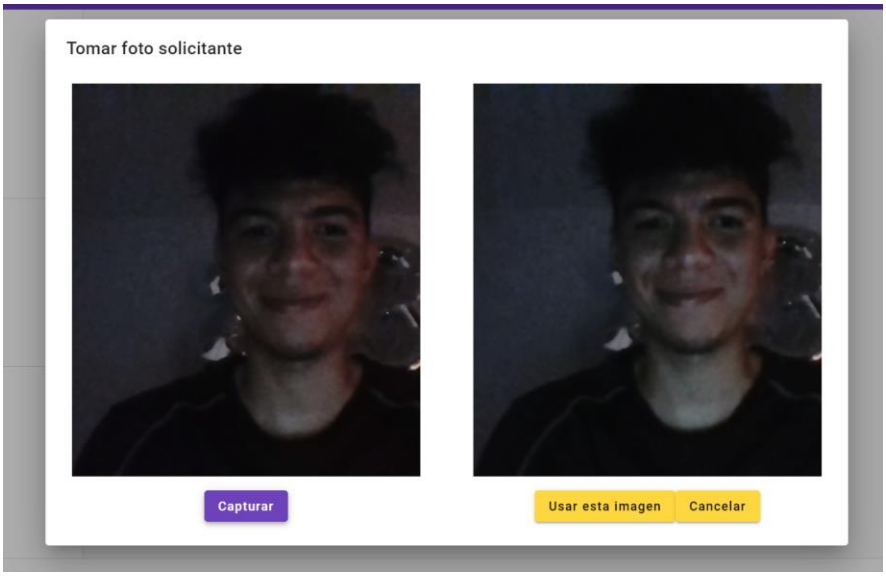
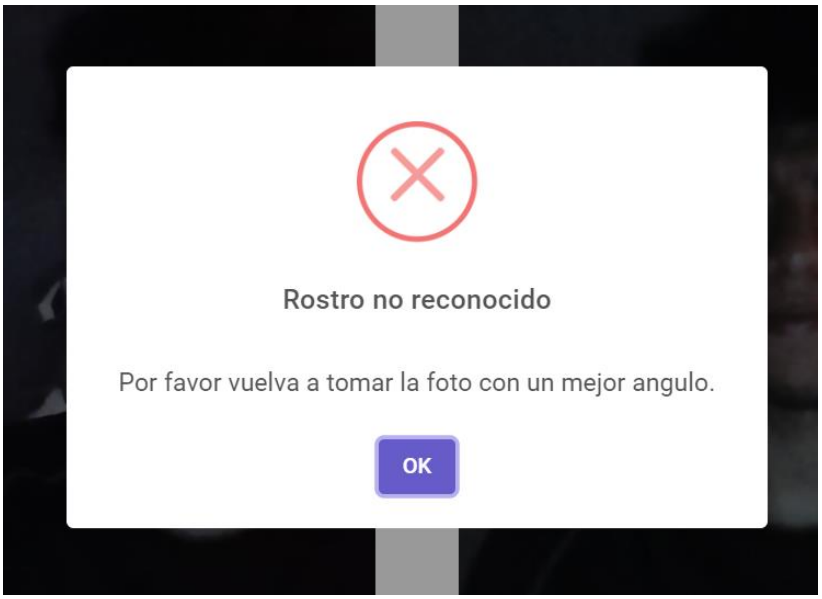


Figura 57

Prototipo funcional - prueba reconocimiento con poca luz fallido



Finalmente se hace una prueba en condiciones óptimas de luz, y expresiones faciales lo más parecidas a la foto que se cargó inicialmente, se observa que el prototipo en este caso sí reconoce a la persona que está frente a la cámara exitosamente y además haya las coincidencias con la foto cargada en el sistema concediendo así la posibilidad de hacer la apertura de la cerradura (Ver figuras 58 y 59).

Figura 58

Prototipo funcional - prueba reconocimiento en condiciones optimas

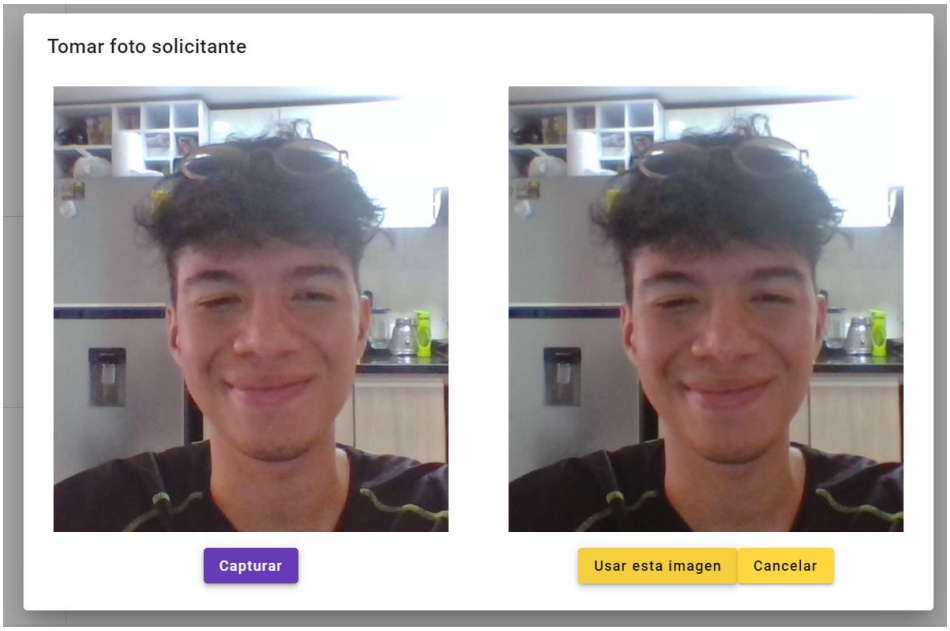


Figura 59

Prototipo funcional - prueba reconocimiento en condiciones óptimas éxitoso

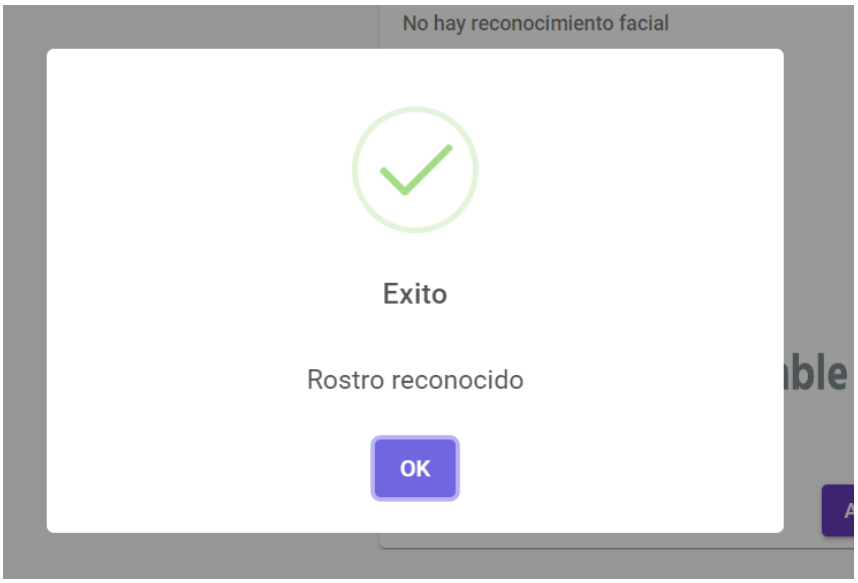
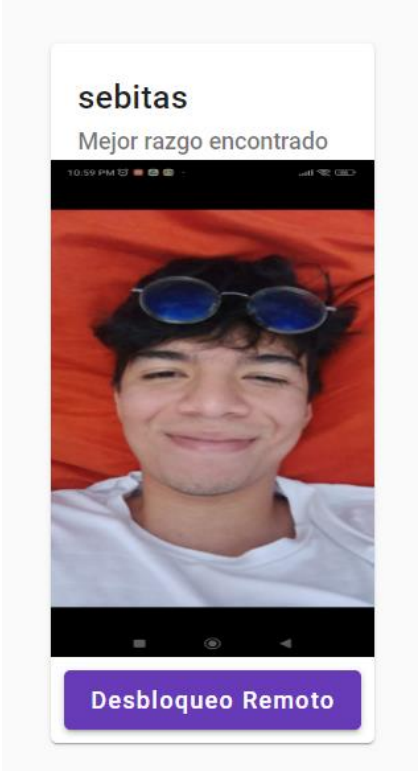


Figura 60

Prototipo funcional - prueba reconocimiento exitoso con mejor rasgo encontrado



9.2. Informe de resultados

La finalidad del presente proyecto era implementar un prototipo funcional que cumpla con la tarea de reconocer un usuario registrado en la plataforma web y concediera el acceso al hogar dando apertura a una cerradura electrónica, además de brindar una herramienta de control y gestión de usuarios.

Se realiza un análisis final del estado del sistema con el fin de identificar qué se cumplió con el segundo prototipo respecto a los requerimientos funcionales propuestos obteniendo lo siguiente:

Tabla 9

Prototipo 2 - Cumplimiento de requerimientos funcionales del aplicativo web

N° Requerimiento	Descripción	Estado
RFW-1	El aplicativo web debe ser capaz de autenticar y autorizar el acceso del usuario.	Cumplido
RFW-2	Se debe poder crear un usuario con sus respectivos datos.	Cumplido
RFW-3	Se debe poder actualizar (cambiar) el estado de usuario en el portal web.	Cumplido
RFW-4	Se debe poder eliminar usuarios.	Cumplido
RFW-5	Se deben poder visualizar una vista general de los usuarios con sus datos.	Cumplido
RFW-6	Se debe poder visualizar una gráfica con los ingresos de un usuario específico en un lapso determinado.	Cumplido
RFW-7	Se debe poder visualizar una gráfica con el total de ingresos de todos los usuarios en un lapso determinado.	Cumplido

Tabla 10

Prototipo 2 - Cumplimiento de requerimientos funcionales del sistema

N° Requerimiento	Descripción	Estado
RFS-1	El sistema debe ser capaz de procesar los datos como la imagen, nickname y el estado del usuario.	Cumplido
RFS-2	El sistema debe ser capaz de detectar e identificar la cara del usuario registrado.	Cumplido
RFS-3	El sistema debe ser capaz de autenticar y autorizar el acceso del usuario al hogar.	Cumplido
RFS-4	El sistema debe ser capaz de captar datos de ingreso una vez se conceda el acceso al hogar.	Cumplido

Se observa que se logró cumplir con los objetivos y requerimientos planteados inicialmente.

Además, en el análisis se identifican una serie de características y pormenores que hacen falta cubrir, por lo que es necesario hacer unas observaciones al trabajo realizado.

Inicialmente se observó que al intentar realizar el reconocimiento facial modificando ciertas cualidades del usuario, el modelo se comporta de manera estricta, impidiendo y restringiendo el acceso al hogar bajo ciertas alteraciones que tenga el rostro del usuario como se pudo observar en la evaluación del prototipo al modificar las condiciones de luz, añadir elementos al rostro o al sufrir un cambio en la apariencia.

Por otro lado, se tiene que el hardware encargado de la apertura de la cerradura funciona con electricidad, si bien el consumo es bajo, se tiene que al sufrir una interrupción del suministro eléctrico bloquearía por completo el ingreso al hogar, lo que hace necesario que para una implementación del prototipo en un escenario real se cuente con una opción de desbloqueo o

apertura manual. Además, también es necesario mantener conectada la ESP32 en la misma red wifi en que está instalado el servidor para abrir la cerradura.

10. Conclusiones

Se finaliza el desarrollo del proyecto dando cumplimiento al objetivo general obteniendo como resultado un prototipo funcional que cumple la tarea de hacer el control de acceso al hogar por medio de reconocimiento facial y además brinda una herramienta de gestión de usuarios como lo es la aplicación web.

Además, tendiendo en cuenta los objetivos específicos del proyecto tenemos que se cumplió el primer objetivo identificando correctamente los requerimientos funcionales y no funcionales de un sistema con estas características, donde se analizó que cualidades debía tener un sistema de control de acceso al hogar, incluyendo las restricciones, los roles que pueden cumplir las personas en un hogar, la información en forma gráfica que podría ser relevante para el habitante o dueño del hogar a la hora de tomar decisiones relacionadas al ingreso, además, se logró cumplir con el segundo objetivo específico del proyecto donde basados en los requerimientos anteriormente planteados se propuso una arquitectura donde se pudiera hacer tangible la solución a esas necesidades identificadas y finalmente se pudo evaluar el funcionamiento del prototipo mediante la implementación del prototipo funcional, donde se pudieron identificar fortalezas y debilidades del prototipo con el fin de proponer mejoras al mismo.

En segundo lugar, el desarrollo del prototipo de plataforma IoT servirá para la gestión de usuario y control de acceso al hogar, además de servir como base para trabajos futuros relacionados con el control de acceso a cualquier lugar que lo requiera.

En tercer lugar, gracias al desarrollo del proyecto se afianzaron conocimientos obtenidos a lo largo de la carrera, donde se reforzaron y pusieron en práctica conceptos de la ingeniería del software, inteligencia artificial e IoT, así como también en el área de la gestión de proyectos.

Por último, se concluye que el proyecto fue un éxito al dar cumplimiento a los objetivos planteados obteniendo un prototipo funcional validado.

11. Trabajo futuro

El prototipo puede servir como base para otros proyectos que requieran o pueda integrar un sistema de control de acceso con reconocimiento facial, por lo que una integración interesante y se propone como trabajo futuro es la integración del presente proyecto con SmartCampus, el cual es un proyecto que se viene adelantando en la Universidad y se basa en plataformas IoT.

Para futuros trabajos en la optimización del prototipo, se sugiere implementar una serie de mejoras y optimizaciones al modelo, por ejemplo, en términos de seguridad se podría analizar la posibilidad de implementar el algoritmo de conteo de parpadeos o análisis de profundidad con el fin de evitar posibles riesgos de vulneración y violación del sistema.

También se podría analizar el comportamiento del sistema cuando se intenta realizar el reconocimiento facial desde un dispositivo donde la calidad de la cámara es baja.

Se sugiere analizar la posibilidad de unificar la ejecución del reconocimiento facial desde un dispositivo o computadora móvil con la finalidad de no depender de un dispositivo de uso individual como lo es un celular.

En términos de operabilidad del prototipo, como se mencionó, el hardware requiere de electricidad para funcionar, por ello se propone para futuras implementaciones o mejoras, suministrar energía eléctrica por medio de una batería, además, buscar la forma de hacer la apertura remota cuando no se está conectado a la misma red wifi, lo que obligaría a hallar la manera de alojar el proyecto en un servidor en la nube y conectar la ESP32 u otro hardware que se quiera utilizar a estar conectado a internet.

Finalmente, en lo que se refiere a la seguridad del prototipo en general, podría considerarse un análisis exhaustivo y profundo de cómo mejorar estas características con una serie desarrollos adicionales en los intermedios de las comunicaciones entre el hardware y el software.

Referencias Bibliográficas

0837L DC 12V 8W Solenoide de Tipo de Abierto para Cerradura de Puerta electrico4 6031021981058. eBay. (s.f.). https://www.ebay.es/itm/185803204153

Aguayo, P. (2004). Introducción al Microcontrolador. *Artıculo, noviembre, tomado de https://d1wqtxts1xzle7.cloudfront.net/39407044/micro-with-cover-page-v2.pdf?Expires=1666367299&Signature=fAuICduRDqbDOE4M9K9-THTlGkvCZWoXiqrzLvEMaLtDGQVQ5kZW2H7JxNavEMkQ-dceJWJFBJ8QZGJxGBgDlXOVD1jQysK-MWS8xaMK1025Tm-Th-Xpx2Kln22FA0shjct4Z5YEnbSsf~SMycMdCSwTl4UItJKYfpHXhCj5aQOXwjzkRcB-9YBrdXRckampxXI372TvwulHy82P7HDgwCxiQLCgz1EA2Vnlb8H3y0IMgJv8WG6jbuZ0yf eOv3cqYbFiOaN-DAX8dBaHi8zOE6pDz~A1lh1nlmODLOGz3qoFA7Q4-~9k5RVpaD52st3GDUpGLzd7ZErjwEewiXn59A__&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA*

Cerradura Electrónica Facelock. lockey Colombia. (2019). Recuperado en octubre 21, 2022, de https://lockeycolombia.com/product/cerradura-electronica-facelock/

Cómputo móvil y dispositivos móviles. Uso de los componentes Físicos de la computadora. (2011). http://cca.org.mx/cca/cursos/cucfc/modulo1/tema4b.html

Especificaciones del módulo ESP32 - blogger. (n.d.). https://vasanza.blogspot.com/2021/07/especificaciones-del-modulo-esp32.html

Espinoza Mina, M. A., & Sierra Cedeño, A. Y. (2018). Análisis comparativo entre ASP.NETY PHP. *INNOVA Research Journal*, 25–43. https://doi.org/10.33890/innova.v3.n4.2018.474

Escate, K. H., Olortegui, L. H., Rodriguez, M. M., Castillo, J. P., & Panana, M. R. (2019). *Cerradura Electrónica Inteligente para el Hogar*. USIL Repositorio Institucional. Retrieved October 21, 2022, from

Face-recognition. PyPI. (s.f.). https://pypi.org/project/face-recognition/

Fisteus, J. A. (2022). *El protocolo HTTP - UC3M*. Universidad Carlos III De Madrid. http://www.it.uc3m.es/jaf/aw/transparencias/http.pdf

He, K., Zhang, X., Ren, S., & Sun, J. (2015, December 10). *Deep residual learning for image recognition*. arXiv.org. https://arxiv.org/abs/1512.03385

Hurto a residencias 2022. Policía Nacional de Colombia. (2023, January 6). https://www.policia.gov.co/contenido/hurto-residencias-2022-0

Kit para desarrolladores nvidia jetson nano. NVIDIA. (2023). https://www.nvidia.com/es-la/autonomous-machines/embedded-systems/jetson-nano/education-projects/

Labeled faces in the wild home. LFW Face Database : Main. (2018). http://vis-www.cs.umass.edu/lfw/

Luján, S. (2002, October 31). *Programación de Aplicaciones web: Historia, principios básicos y clientes web*. RUA. https://rua.ua.es/dspace/handle/10045/16995

López Garzón, W; Cárdenas López, J. 2019. Tecnología IoT (Internet of Things) y El Big Data. Mare Ingenii. Ingenierías 1(1). Disponible en http://cipres.sanmateo.edu.co/index.php/mi/article/view/183/162

Montoro, A. F. (2012). *Python 3: Al Descubierto*. Alfaomega.

Moreno, G. (s.f.). *Boletín Científico - JAVA como lenguaje universal de programación*. UAEH. https://repository.uaeh.edu.mx/revistas/index.php/xikua/article/download/332/4434?inline=1#refe1

Moreno Latorre, J. P. *(2016). Prototipo para el control de una cerradura electrónica por medio de reconocimiento facial. Recuperado de: http://hdl.handle.net/11349/4687.*

Módulo Relay 1CH 5VDC. Naylamp Mechatronics - Perú. (s.f.). https://naylampmechatronics.com/drivers/297-modulo-relay-1-canal-5vdc.html

Módulo rele DE 1 Canal 5V 1CH. SSDIELECT ELECTRONICA SAS. (s.f.). https://ssdielect.com/rele-relay-o-relevo/992-md-rele-1ch.html

Orange Pi. Orangepi. (s.f.). http://www.orangepi.org/html/hardWare/computerAndMicrocontrollers/details/orange-pi-3-LTS.html

Patel, A. (2019). *Face recognition for the happy house*. GitHub. https://github.com/ashishpatel26/Andrew-NG-Notes/blob/master/Deep%20Learning%20Notebooks%20by%20Andrew%20NG/Convolutional%20Neural%20Networks/Week4/Face%20Recognition/Face%20Recognition%20for%20the%20Happy%20House.ipynb

Pérez, S., Quispe, J., Mullicundo, F., & Lamas, D. (2021, Abril 15). *Sedici - Repositorio de la universidad nacional de la plata*. XXIII Workshop de Investigadores en Ciencias de la Computación 347. http://sedici.unlp.edu.ar/bitstream/handle/10915/120476/Ponencia.pdf-PDFA.pdf?sequence=1

Pisco Gómez, Á., Regalado Jalca, J. J., Gutiérrez García, J., Quimis Sánchez, O., Marcillo Parrales, K., & Marcillo Merino, J. (2017). *Fundamentos Sobre La Gestión de Base de Datos*. https://doi.org/10.17993/ingytec.2017.23

Poveda, M., & Merchán, F. (2015, December). *Implementación de un sistema de control de acceso basado en reconocimiento facial*. Vista de Implementación de un sistema de control de Acceso Basado en reconomiento facial. https://revistas.utp.ac.pa/index.php/prisma/article/view/609/630

Raspberry Pi. (s.f.). *Buy A raspberry pi 4 model B*. Raspberry Pi. https://www.raspberrypi.com/products/raspberry-pi-4-model-b/

Reconocimiento facial. INTERPOL. (2023). https://www.interpol.int/es/Como-trabajamos/Policia-cientifica/Reconocimiento-facial

Rosebrock, A. (2023, June 8). *Face recognition with opencv, python, and Deep Learning*. PyImageSearch. https://pyimagesearch.com/2018/06/18/face-recognition-with-opencv-python-and-deep-learning/

Saez, A. S. (2021, febrero 1). Qué son las cerraduras electrónicas, Todos los tipos y las mejores Marcas del Mercado. LinkedIn. Accedido octubre 21, 2022, tomado de https://www.linkedin.com/pulse/qu%C3%A9-son-las-cerraduras-electr%C3%B3nicas-todos-los-tipos-y-soria-saez/?originalSubdomain=es

Sánchez, M. (2012). Javascript. INNOVACIÓN Y CUALIFICACIÓN, S.L. https://books.google.es/books?hl=es&lr=&id=3x09sewjaHIC&oi=fnd&pg=PA7&dq=que+es+javascript&ots=YgPINFrQsF&sig=zTrkQjoxvKj9l5k0dR7IdeusLz0#v=onepage&q=que%20es%20javascript&f=false

Sustaita, B. E., Mazcorro, M. R., & Huerta, A. F. (2013). Año I, no. 01 enero-junio 2013 ISSN: En trámite. Universidad Autónoma de Nuevo León. Recuperado en octubre 21, 2022, de http://eprints.uanl.mx/9836/1/Desarrollo%20de%20tecnologia.pdf

VIM4. Khadas. (2022). https://www.khadas.com/vim4

Apéndices

Anexo A

Figura 61

Configuración OpenCV sin CUDA en Jetpack

Nota. Tomado de Q-engineering, 2023. https://qengineering.eu/install-opencv-4.5-on-jetson-nano.html

Figura 62

Configuración de OpenCV con CUDA

Nota. Tomado de Q-engineering, 2023. https://qengineering.eu/install-opencv-4.5-on-jetson-nano.html