UNIVERSIDAD UTE

FACULTAD DE CIENCIAS DE LA INGENIERÍA E INDUSTRIAS

CARRERA DE INGENIERÍA INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN

DESARROLLO DE UNA APLICACIÓN QUE USANDO RECONOCIMIENTO FACIAL PERMITE BLOQUEAR/DESBLOQUEAR DISPOSITIVOS DE HOGARES INTELIGENTES

TRABAJO PREVIO A LA OBTENCIÓN DEL TÍTULO

DE INGENIERO EN INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN

NICOLÁS PATRICIO JIMÉNEZ DÁVILA

DIRECTOR: ING. RODRIGO PROAÑO

Quito, agosto 2020

FORMULARIO DE REGISTRO BIBLIOGRÁFICO
TRABAJO DE TITULACIÓN

DATOS DE CONTACTO	
CÉDULA DE IDENTIDAD:	1725517393
APELLIDO Y NOMBRES:	JIMÉNEZ NICOLAS
DIRECCIÓN:	JUAN LOPEZ DE VELASCO S11-360 Y EL CANELO
EMAIL:	nicolas.jimenez@ute.edu.ec
TELÉFONO FIJO:	(02)2663562
TELÉFONO MOVIL:	0998413483

DATOS DE LA OBRA	
TÍTULO:	DESARROLLO DE UNA APLICACIÓN QUE USANDO RECONOCIMIENTO FACIAL PERMITE BLOQUEAR/DESBLOQUEAR DISPOSITIVOS DE HOGARES INTELIGENTES
AUTOR O AUTORES:	Nicolás Patricio Jiménez Dávila
FECHA DE ENTREGA DEL PROYECTO DE TITULACIÓN:	
DIRECTOR DEL PROYECTO DE TITULACIÓN:	Rodrigo Proaño
PROGRAMA	PREGRADO [X] POSGRADO [ ]
TÍTULO POR EL QUE OPTA:	Ingeniero En Informática y Ciencias De La Computación
RESUMEN:	El proyecto final de carrera tuvo como propósito, el desarrollo de una aplicación para dispositivos móviles que cuenten con un sistema operativo Android, el cual permita el uso del reconocimiento facial como método de seguridad biométrico, capaz de bloquear y desbloquear dispositivos inteligentes del hogar. Con el uso de la metodología SCRUM y el sistema de seguridad biométrico facial, se pudo elaborar un aplicativo móvil completamente funcional. El reconocimiento facial, es un sistema de seguridad biométrico que posee varias características positivas, entre ellas se pudo observar que posee mayor

	precisión en la identificación de un sujeto, existe menos probabilidad de suplantación de identidad, poca manipulación por parte del usuario final por ser un método no invasivo, alta disponibilidad y facilidad de obtención de los datos almacenados. Obteniendo como resultado un control discreto de la información. En el análisis del reconocimiento facial se tomó en cuenta una secuencia de pasos que se debe seguir para la toma e identificación de un rostro. En las pruebas de la aplicación se usó un dispositivo móvil para la toma secuencial de las fotografías, para agregar, reconocer e identificar el usuario. Se obtuvo como resultado un aplicativo capaz de agregar y eliminar rostros de varios usuarios a un grupo de seguridad determinado, para bloquear y desbloquear vía bluetooth una cerradura eléctrica, con un sistema de seguridad biométrico facial de 27 puntos, brindando una alta disponibilidad del servicio de reconocimiento facial.
PALABRAS CLAVES:	**SCRUM, reconocimiento facial, bluetooth, grupo de seguridad**
ABSTRACT:	The purpose of the final degree project was to develop an application for mobile devices that have an Android operating system, which one allows the use of facial recognition as a security method biometric, capable of locking and unlocking smart home devices. Using SCRUM methodology and the biometric security system facial, a fully functional mobile application could be developed. The facial recognition, is a biometric security system that has several positive characteristics, among them it was observed that it has greater greater precision identifying people, there is less probability of spoofing, less manipulation by the final user because it is a non-invasive method, high availability and ease of obtaining data stored. Achieving as a result a discreet control of the information. The facial

	recognition analysis took into account a sequence of steps to be followed in order to take and identify a face. In the tests of the application a mobile device was used for taking sequential photographs, to add, recognize and identify the user. The result was an application capable of adding and removing faces from multiple users of a security group, to lock and unlock an electric lock via bluetooth, with a security system facial biometric of twenty seven points, providing a high availability of the service of facial recognition.
KEYWORDS	**SCRUM, Facial Recognition, Bluetooth, Security Group**

Se autoriza la publicación de este Proyecto de Titulación en el Repositorio Digital de la Institución.

Jiménez Dávila Nicolás patricio
1725517393

DECLARACIÓN Y AUTORIZACIÓN

Yo, Jiménez Dávila Nicolás Patricio**,** CI 1725517393 autor/a del trabajo de titulación: Desarrollo de una aplicación que usando reconocimiento facial permite bloquear/desbloquear dispositivos de hogares inteligentes. previo a la obtención del título de Ingeniero En Informática Y Ciencias De La Computación en la Universidad UTE.

1. Declaro tener pleno conocimiento de la obligación que tienen las Instituciones de Educación Superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación de grado para que sea integrado al Sistema Nacional de información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.
2. Autorizo a la BIBLIOTECA de la Universidad UTE a tener una copia del referido trabajo de titulación de grado con el propósito de generar un Repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Quito, agosto del 2020

JIMÉNEZ DÁVILA NICOLÁS PATRICIO

1725517393

CERTIFICACIÓN DEL TUTOR

En mi calidad de tutor, certifico que el presente trabajo de titulación que lleva por título Desarrollo de una aplicación que usando reconocimiento facial permite bloquear/desbloquear dispositivos de hogares inteligentes para aspirar al título de Ingeniero En Informática Y Ciencias De La Computación fue desarrollado por Jiménez Dávila Nicolás Patricio**,** bajo mi dirección y supervisión, en la Facultad de Ciencias de la Ingeniería e Industrias; y que dicho trabajo cumple con las condiciones requeridas para ser sometido a las evaluación respectiva de acuerdo a la normativa interna de la Universidad UTE.

Ing. Rodrigo Proaño Escalante

DIRECTOR DEL TRABAJO

C.I. 170854904-1

Yo, Nicolás Patricio Jiménez Dávila, portador(a) de la cédula de identidad 1725517393, declaro que el trabajo aquí descrito es de mi autoría, que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en ese documento.

La Universidad UTE puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normativa institucional vigente.

JIMÉNEZ DÁVILA NICOLÁS PATRICIO

1725517393

ÍNDICE DE CONTENIDOS

PÁGINA

ÍNDICE DE TABLAS

PÁGINA

ÍNDICE DE FIGURAS

PÁGINA

ÍNDICE DE ANEXOS

PÁGINA

RESUMEN

El proyecto final de carrera tuvo como propósito, el desarrollo de una aplicación para dispositivos móviles que cuenten con un sistema operativo Android, el cual permita el uso del reconocimiento facial como método de seguridad biométrico, capaz de bloquear y desbloquear dispositivos inteligentes del hogar.

Con el uso de la metodología SCRUM y el sistema de seguridad biométrico facial, se pudo elaborar un aplicativo móvil completamente funcional. El reconocimiento facial, es un sistema de seguridad biométrico que posee varias características positivas, entre ellas se pudo observar que posee mayor precisión en la identificación de un sujeto, existe menos probabilidad de suplantación de identidad, poca manipulación por parte del usuario final por ser un método no invasivo, alta disponibilidad y facilidad de obtención de los datos almacenados. Obteniendo como resultado un control discreto de la información. En el análisis del reconocimiento facial se tomó en cuenta una secuencia de pasos que se debe seguir para la toma e identificación de un rostro. En las pruebas de la aplicación se usó un dispositivo móvil para la toma secuencial de las fotografías, para agregar, reconocer e identificar el usuario.

Se obtuvo como resultado un aplicativo capaz de agregar y eliminar rostros de varios usuarios a un grupo de seguridad determinado, para bloquear y desbloquear vía bluetooth una cerradura eléctrica, con un sistema de seguridad biométrico facial de veintisiete puntos, brindando una alta disponibilidad del servicio de reconocimiento facial.

Palabras Clave: SCRUM, reconocimiento facial, bluetooth, grupo de seguridad

ABSTRACT

The purpose of the final degree project was to develop an application for mobile devices that have an Android operating system, which one allows the use of facial recognition as a security method biometric, capable of locking and unlocking smart home devices.

Using SCRUM methodology and the biometric security system facial, a fully functional mobile application could be developed. The facial recognition, is a biometric security system that has several positive characteristics, among them it was observed that it has greater greater precision identifying people, there is less probability of spoofing, less manipulation by the final user because it is a non-invasive method, high availability and ease of obtaining data stored. Achieving as a result a discreet control of the information. The facial recognition analysis took into account a sequence of steps to be followed in order to take and identify a face. In the tests of the application a mobile device was used for taking sequential photographs, to add, recognize and identify the user.

The result was an application capable of adding and removing faces from multiple users of a security group, to lock and unlock an electric lock via bluetooth, with a security system facial biometric of twenty seven points, providing a high availability of the service of facial recognition.

Keywords: SCRUM, Facial Recognition, Bluetooth, Security Group

1. INTRODUCCIÓN

1. INTRODUCCIÓN

El estudio de la biometría viene de algunas décadas atrás, se analizaba las maneras de cómo reconocer sujetos que eran perseguidos por la ley, para esto existieron varios métodos de reconocimiento de personas, uno de ellos era el bertillonaje, este método consistía en el reconocimiento de todo el cuerpo de acuerdo a mediciones hechas en ese momento, esto no era nada seguro al tener sujetos de las mismas contexturas, después de varios años, surgió la biométrica aplicada a la sociedad, con esto se dio el aprendizaje en métodos de clasificación y comparación netamente biométricos, esto daba a conocer la huella dactilar, cara, iris, mano y forma de caminar.(Serratosa, 2008)

Actualmente los sistemas biométricos, tienen un rol fundamental en la sociedad para el proceso de reconocimiento de personas, las cuales se basan en procedimientos y políticas públicas de seguridad, tal es así que los gobiernos se apoyan en las diferentes biometrías para el reconocimiento de civiles, existen diferentes usos que se da a los datos biométricos de una persona, como puede ser: la seguridad pública de un aeropuerto, seguridad en ciudades, restringir el acceso a sitios seguros, tanto de manera física como virtual.(Etchart, Luna, Leal, Benedetto, & Alvez, s. f.)

El reconocimiento biométrico facial, es un sistema no invasivo, es decir no tiene intrusión física con la persona y el autenticador. Se puede analizar el reconocimiento facial de un objetivo en movimiento, es decir un video, sin el conocimiento y la colaboración del sujeto.(Costa, 2018)

Existe una tendencia creciente del uso de la biometría facial en los dispositivos móviles para mejorar la seguridad, con sistemas de fácil uso, aplicación y sin necesidad de colocarse ningún elemento en el cuerpo humano.(Lía & Alberto, s. f.)

El avance tecnológico permitió automatizar el reconocimiento facial, haciendo un sistema que se asemeja a la percepción visual de los seres humanos, y poder aplicarlos a diferentes campos en los que se requiera mejorar la seguridad. Esto nos permite tener menos causas de fraude de identidad.

El reconocimiento biométrico facial ofrece una solución con niveles de error casi nulos, es decir una probabilidad mínima de que el sistema sea vulnerado y se tenga acceso a recursos tecnológicos que se usa habitualmente en el hogar. Se puede pensar en dispositivos que tengan algún tipo de conexión directa con un dispositivo móvil, que disponga una cámara integrada.

El reconocimiento facial, se compone principalmente de un software, que realiza el procedimiento de una manera automatizada, en la que compara en tiempo real un rostro y analiza cada uno de los contornos y los rasgos faciales

que la diferencia del resto, el sistema de reconocimiento debe recibir una imagen que se codifica de manera automática y empieza a denotar los rasgos faciales y compararlos con perfiles ya ingresados anteriormente, en el cual lo acepta o rechaza si no existe ninguna similitud.

El sistema de reconocimiento facial generalmente consta de tres fases, la fase de entrenamiento y modelado, la fase de almacenamiento del modelo obtenido en la fase anterior y la fase de pruebas se realiza con comparaciones de un conjunto de imágenes, en las que debe tener un número de parámetros en los que deben coincidir.(Lorente Giménez, 1998)

En el sistema de reconocimiento facial, el proceso que debe seguir se describe por varias fases, primero como se menciona, se debe recibir una imagen del sujeto, el algoritmo debe realizar la detección del rostro que se registrará, el siguiente paso será la extracción de toda la información biométrica que se tenga de la imagen, en la que se incluye las propiedades geométricas que posee el rostro de cada una de las fotografías, y la alineación de la cara. Como último paso se realiza el entrenamiento del sistema, guardando toda la información biométrica para un futuro reconocimiento.

El propósito del proyecto es desarrollar un aplicativo móvil que permita implementar la seguridad biométrica facial y dar a conocer los aspectos positivos que puede brindar. El aplicativo se dará uso en dispositivos inteligentes del hogar, para lo cual deberá haber comunicación entre varios dispositivos, logrando así tener una opción para brindar seguridad. El dispositivo que se utilizará para ejemplificar el uso del aplicativo móvil será una cerradura eléctrica.

El objetivo general del proyecto es desarrollar una aplicación para dispositivos móviles con sistema operativo Android, que por medio de reconocimiento facial pueda abrir o cerrar una cerradura de manera automática.

También se definió los objetivos específicos:

- Investigar modelos y librerías de reconocimiento de patrones que ayuden a realizar el reconocimiento facial.

- Investigar las diferentes metodologías para el control de una cerradura automática.

- Investigar las conexiones y configuraciones necesarias entre la placa electrónica y el dispositivo móvil, para realizar las funcionalidades indicadas.

- Implementar un prototipo de sistema de reconocimiento facial para la apertura/cierre de una puerta usando dispositivos móviles con S.O. Android.

La aplicación de reconocimiento facial se realizará con la cámara frontal de un dispositivo móvil con sistema operativo Android. La imagen del rostro deberá estar previamente registrada para poder proceder al bloqueo y desbloqueo del dispositivo inteligente, en el presente caso se utilizará una placa electrónica que interactuará con la cerradura automática, la misma que realizará las funciones necesarias para abrir o cerrar la puerta. Como medida de contingencia el sistema de reconocimiento facial permitirá el ingreso de una clave para su funcionamiento.

La elaboración de este sistema se lo realizará en un computador que cuenta con un sistema operativo Windows 10 Home, con una memoria instalada RAM de 12 GB, con un procesador Core i7-5500 de 2.40 GHz y un disco de almacenamiento SSD.

El dispositivo móvil que se utilizará para realizar la implementación y las pruebas físicas es un Xiaomi Redmi Note 7, el cual dispone de 4GB de RAM, 64GB de almacenamiento interno y una cámara frontal de 13mpx, con sistema operativo Android 9.0, cabe recalcar que la aplicación se podrá ejecutar con otros dispositivos que cumplan con requerimientos mínimos, un dispositivo que tenga cámara trasera y frontal, conexión bluetooth, conexión a internet, con un sistema operativo Android mayor a la versión 6.0 para su funcionamiento.

El dispositivo inteligente que se usará, para la apertura/cierre será una cerradura diseñada para este propósito la misma que tendrá una conexión directa con una placa Arduino UNO y será instalada en una maqueta para el desarrollo del prototipo.

El presente trabajo tiene los siguientes alcances

- Se usará una placa electrónica para la toma de control de la cerradura eléctrica y sus funcionalidades.
- Se hará un análisis para el tipo de conexión que tendrá la placa electrónica y el dispositivo móvil Android.
- Se analizará el procesamiento de las imágenes para el sistema de reconocimiento biométrico facial con librerías funcionales para Android.
- Se aplicarán los procesos necesarios para realizar la biometría por medio de las librerías analizadas.
- Se analizarán herramientas de manejo de información para los datos biométricos obtenidos.

- El sistema se realizará en la plataforma Android para un dispositivo móvil, con una interfaz simple e intuitiva para el uso.

El sistema operativo Android, es generalmente usado en dispositivos móviles, es capaz de administrar los recursos del dispositivo, para que sea usado eficazmente, el sistema operativo Android es un producto de Google y la OpenHandset Alliance.(Robledo, s. f.)

Es un sistema operativo de plataforma abierta basado sobre el kernel de Linux, diseñada para dispositivos móviles, la cual permite el desarrollo de aplicaciones y la modificación de existentes, que generalmente están basadas en el lenguaje de programación JAVA, es compatible con múltiples dispositivos lanzados al mercado.

Posee un portal llamado Play Store, en donde se encuentran varias aplicaciones desarrolladas por diferentes empresas, permite realizar actualizaciones del sistema operativo, siempre y cuando el dispositivo cumpla con los requerimientos para la actualización. (Malave Polanco & Beauperthuy Taibo, 2011)

Android Studio es un entorno de desarrollo integrado, para desarrollar aplicaciones de dispositivos móviles en el sistema operativo Android, está basado en IntelliJ IDEA(Hohensee, s. f.), Android Studio ofrece algunas herramientas que nos servirá en el momento del desarrollo de aplicaciones móviles, como son:

- Sistema de compilación
- Emulador Rápido
- Entorno Unificado
- Instant Run
- Frameworks y Herramientas de prueba
- Herramientas Lint
- Compatibilidad con c++ y NDK
- Soporte Google cloud plataform

Android estudio es un entorno de desarrollo completo, por todas las funcionalidades que brinda, dispone de igual manera una amplia gama de dispositivos móviles virtuales emuladores de prueba, en los que se puede ejecutar las aplicaciones en desarrollo.

Algunas funcionalidades que se puede tener en este entorno de desarrollo son la posibilidad de desarrollar aplicativos para diferentes dispositivos que posean Sistema Android, como puede ser Android Wear y Android TV.

Android Studio permite la administración de plugins, los que dan facilidades en el desarrollo de un aplicativo.

Android Studio posee además una interfaz intuitiva para la instalación de plug-ins, de igual manera en la misma interfaz se puede corroborar los complementos instalados y cuales están disponibles para actualizar.(Calvachi, 2018)

Java es un lenguaje de alto nivel orientado a objetos el cual se introdujo a en el año de 1991, compilado e interpretado. Hereda la sintaxis de C/C++ y muchas de las características orientadas a objetos de C++.(Groussard)

Java fue construido con varias funcionalidades y capacidades para una interconexión TCP/IP, permite a los desarrolladores adquirir información de la red de una manera simplificada. Java es un lenguaje de programación robusto, permite que se compruebe el código al momento de la compilación, también permite la comprobación del código desarrollado en tiempo de ejecución.

El lenguaje de programación JAVA es multihilo, es decir que puede realizar varias tareas al mismo tiempo. De una forma fácil y de igual manera robusta.(SANABRIA, 1998)

SQL Server, es un sistema de gestión de base de datos relacional, su uso se lo puede hacer desde una portátil, hasta servidores empresariales, fue desarrollado en 1980 por la empresa Sybase, la cual se dedicaba al desarrollo de tecnología de la información. Fueron creados originalmente para sistemas UNIX y posteriormente en sistemas Windows NT para Microsoft, el lenguaje de desarrollo que se usa, es Transact-SQL, el cual es una implementación del lenguaje estándar del lenguaje SQL, es utilizado para la manipulación y la recuperación de datos, así como crear tablas y definir las relaciones correspondientes entre ellas.(Korth & Sudarshan, 2002)

Algunas de las características que se puede dar de Microsoft SQL server es que tiene soporte para transacciones, escalabilidad estabilidad y seguridad, soporta procedimientos almacenados, tiene un potente entorno gráfico para administración, el cual permite la ejecución de varias consultas SQL y ver los resultados en ese instante.(Server, s. f.)

SQL server posee varias herramientas que se usan en una base datos, tales como: desarrollo, consulta, ajuste, verificación y administración de esta, para el desarrollo de la base de datos se tiene varias herramientas que proporcionan mecanismos para diseñar tablas, columnas, claves, índices, relaciones y restricciones, en las herramientas de consulta se tiene la herramienta de analizador de consultas de SQL Server, el cual posee una interfaz intuitiva para realizar consultas a la base de datos deseada(Korth & Sudarshan, 2002).

Arduino es una empresa que desarrolla hardware y software libre, es así que existen placas de desarrollo de hardware de uso libre, uno de los ejemplos que se tiene presente de las placas electrónicas, es el Arduino UNO, la placa un tamaño de 75*53mm, tiene una unidad de procesamiento que consiste en un microcontrolador ATmega328. Esta placa puede ser alimentada mediante USB o también puede tener una fuente externa, en la figura 1 se aprecia el dispositivo Arduino UNO.(Lledó Sánchez, 2012)

Figura 1. Arquitectura Arduino uno(Arduino Uno, 2020)

La plataforma Arduino posee su propio lenguaje que está basado en los lenguajes C/C++, por lo cual soporta algunas funciones del estándar C, y algunas de C++, existe la posibilidad de utilizar otro tipo de lenguajes para la programación de la placa, como puede ser Java Processing, Python, Mathematica, Matlab, Perl, Visual Basic, etc. (Lledó Sánchez, 2012)

El entorno de desarrollo de Arduino es intuitivo para el usuario, además se puede descargar sin costo alguno, desde la página oficial de Arduino para diferentes sistemas operativos. El entorno de desarrollo Arduino está formado por algunas barras de menú en donde se encontrarán las posibles placas a conectarse con el software, también se puede visualizar el puerto que se está usando en la placa.(Lledó Sánchez, 2012)

El reconocimiento facial por medios tecnológicos hoy en día es un área de investigación bastante amplia, la identificación de una persona por medio del rostro emula el proceso cognitivo que realiza el ser humano al reconocer a sus pares. El rostro es un rasgo que diferencia a cada ser humano por el cual se puede identificar a individuos a simple vista.

La automatización de este proceso llega a tener un buen rendimiento, inclusive obteniendo mejores resultados que un reconocimiento facial por parte del ser humano, el uso de este tipo de biometría en tiempo real permite que existan menos intrusiones y la obtención de las características analizadas en poco tiempo.

Con el avance de la tecnología y los dispositivos móviles, es posible que se realicen las funciones de reconocimiento por medio de un smartphone que posea una cámara frontal, lo cual hace factible el uso de esta biometría, inclusive para la seguridad del dispositivo.

El rostro humano brinda gran cantidad de información sobre el sujeto a reconocer, existe una similitud en la posición de los rasgos faciales como se muestra en la figura 2, con el resto de los sujetos, lo que puede ayudar en la automatización y en la obtención de los parámetros como son: Orejas, cejas, ojos, nariz, boca, forma lineal del rostro, mentón y pómulos.(Eslava, 2013)

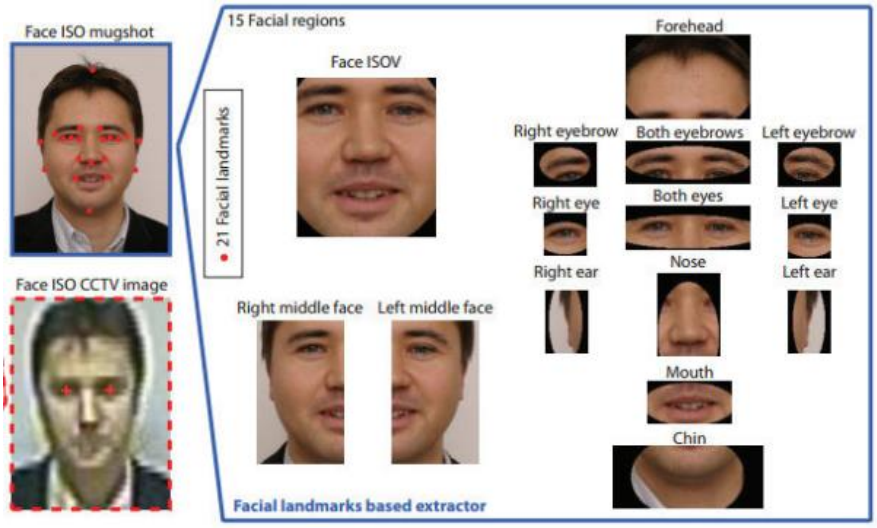

Figura 2. Reconocimiento facial. (elDiario.es, 2020)

El proceso de reconocimiento facial se lo realiza en 5 etapas, la primera es la adquisición de la imagen que se procesará, la segunda es la detección y localización del rostro en la imagen adquirida, la tercera etapa es el procesamiento de la imagen para tener las características singulares del rostro, la cuarta etapa, es la extracción de las características, las cuales se podrá almacenar en una base de datos, es decir las características que se obtuvieron en el entrenamiento del algoritmo, y por último la comparación y reconocimiento del rostro vs la base de datos con las características almacenadas, en la figura 3 se muestran los pasos necesarios.

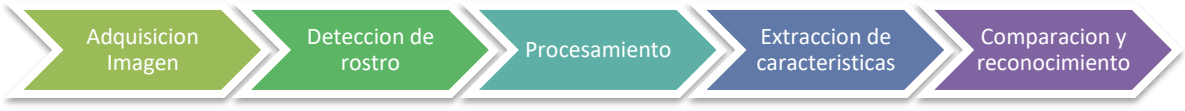

Figura 3. Diagrama reconocimiento facial.

Adquisición de la imagen: La imagen de entrada que se utilizará para la etapa de reconocimiento facial, se la realiza en cualquier dispositivo que sea capaz de tomar imágenes, como puede ser una cámara fotográfica, una cámara de vigilancia o la cámara integrada de un dispositivo móvil. Se debe

considerar que es requerida una buena iluminación para captar los detalles del rostro, es recomendable usar fondos neutros que no contengan rostros, la imagen debe cumplir ciertas especificaciones, el rostro debe ser de al menos 200x200 pixeles y 100 pixeles entre los ojos, se recomienda este tamaño mínimo cuando se da uso al API de Microsoft.(Microsoft, 2019)

Detección: En esta etapa, se analiza la imagen en búsqueda y localización del rostro humano, esta etapa es de vital importancia, si no se realiza con exactitud la localización del rostro, el resto de las etapas no se realizará correctamente.

Procesamiento: Esta etapa depende de la detección realizada, aquí se realiza una serie de transformaciones geométricas en la imagen para posterior extracción de las características.(Fernández, Belmonte, & Ortega, 2015)

Extracción de características: La extracción se emplea para obtener solamente la información relevante del rostro para realizar una comparación, en la extracción de las características también se puede realizar el entrenamiento necesario para cualquier algoritmo que use. Con esto se logra tener una base de conocimiento.(Fernández et al., 2015)

Comparación y reconocimiento: En esta última etapa, se realiza comparaciones con las características anteriormente extraídas, si se obtiene una similitud se podrá dar una validación del rostro que se adquirió en la primera etapa, caso contrario el rostro será rechazado.

Microsoft Azure posee un conjunto de servicios base que son alojados en la nube, pueden usarse en conjunto o también se pueden usar los servicios de manera individual, Microsoft Azure es una plataforma interoperable, permite el desarrollo en varios lenguajes de programación y la comunicación con un entorno externo.(Collier & Robin, 2015)

Microsoft Azure brinda facilidades a los desarrolladores, para aplicar soluciones en aplicaciones Cloud sin agregar complejidad en el uso de las herramientas, esta plataforma ofrece a los desarrolladores herramientas de servicios de ejecución y almacenamiento, definiéndolo, así como un sistema operativo en la nube, se podrá de igual manera gestionar la información en los centros de datos de Microsoft.

Microsoft SQL Azure, el cual hace posible tener una base de datos relacional que se almacena y se maneja desde la nube de Microsoft, es decir un sistema de base de datos similar al SQL server, pero con la única diferencia de ser manejado en la nube. Microsoft SQL Azure es uno de los primeros gestores para la nube, relacional que puede ejecutar consultas SQL.

Se tiene múltiples ventajas a la hora de tener un servicio almacenado en la nube como es Azure, se puede ejecutar procesos genéricos en la nube, crear,

modificar y distribuir aplicaciones escalables con recursos internos mínimos, se puede realizar un almacenamiento de gran cantidad, procesamiento de lotes, creaciones, evaluaciones, depuraciones y distribuciones del servicios web, de una manera eficaz y accesible, logrando reducir costes y esfuerzos en la administración de TI, respondiendo de igual manera a los cambios tecnológicos que surgen con el pasar del tiempo, también reduce la necesidad de administrar hardware propio y reduce el riesgo de pérdida de la información.(Rikelmer, 2012)

Cabe recalcar que los servicios de Azure solamente funcionan con internet, de igual manera el servicio de Microsoft Azure Face API, este servicio de Microsoft Azure tiene varias funciones, la primera es la detección, esta función es la encargada de analizar en una imagen dada, la cantidad de rostros que existen en la misma, es decir, la detección encuentra los rostros humanos en una imagen y devuelve recuadros delimitadores que indican las ubicaciones de los rostros, todas las demás funciones dependen directamente de esta función.

La función de verificación de Microsoft Azure Face API, se basa en la existencia de un identificador, el cual permite buscar, identificar y asociar un rostro en su repositorio privado que puede poseer aproximadamente un millón de personas.

Por último, se tiene la función de identificación, la cual compara una plantilla del rostro identificado, con el resto de las plantillas almacenadas en su repositorio, esta función devuelve una colección de rostros similares a la identificada. Y realiza una coincidencia de “uno a muchos”, haciendo que la respuesta sea la más fiable.(Microsoft, 2019)

El API Microsoft Azure, es un servicio basado en la nube, que brinda al usuario una fluidez y seguridad en sus datos enviados, las características principales de esta API incluyen, la detección de rostros, la identificación de un sujeto obtenido con la detección de rostros, y el reconocimiento de emociones del sujeto.

La API es capaz de identificar a un sujeto en un repositorio privado de hasta 1 millón de personas, esta interfaz de programación de aplicaciones cuenta con una implementación flexible, lo que quiere decir que se puede ejecutar en la nube o en el perímetro de ella usando contenedores. Tiene una seguridad incorporada teniendo una confianza de seguridad de nivel empresarial. Además, garantiza una disponibilidad de su servicio del 99.9%.

Para la aplicación de las funciones de este algoritmo se deben tener en cuenta los puntos que se darán uso. Para esto existen 27 puntos predeterminados para el uso del algoritmo, como se muestra en la figura 4.

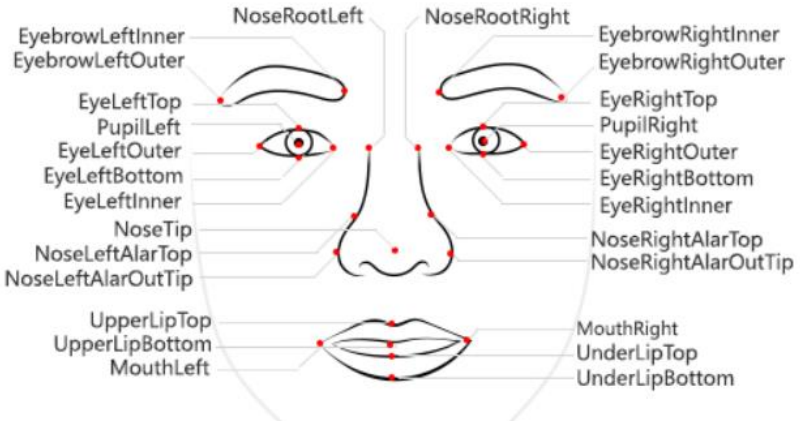

Figura 4 Puntos referenciales reconocimiento facial(Microsoft, 2020)

Las coordenadas obtenidas en dichos puntos se devuelven como unidades de pixeles.

El tratamiento de los rostros se realiza mediante Face-Detect , esta función detecta un rostro y lo devuelve rectángulos de rostros, tomando en cuenta los puntos referenciales, la API no almacena las imágenes de los rostros detectados, solamente enviara los puntos referenciales al servidor.

Faceid, es uno de los identificadores usados, para diferenciar los rostros en el servidor, el cual se ocupará en otras funciones que competen al reconocimiento facial, se pueden detectar hasta 100 rostros en una sola imagen, en las que realiza un ordenamiento del rostro obtenido más grande hasta el más pequeño, el modelo predeterminado para la detección de un rostro es Detection_01, la cual es usado preferencialmente para rostros frontales cercanos(Microsoft, 2020).

Para el siguiente método, correspondiente a la identificación del sujeto, se utiliza predeterminadamente un modelo de aprendizaje automático, el cual es Recognition_01, Face-identify, calculara las semejanzas entre el rostro que fue consultado y el grupo de personas a verificar, esto se realiza de acuerdo a la matriz tomada de Faceids, y devolverá los sujetos clasificados según la confianza de similitud entre ellos, el grupo de personas que se consulta, debe estar entrenado anteriormente los rostros que se deseen consultar para dar un resultado positivo.

Al momento de identificar los rostros el algoritmo predeterminado permite identificarlos de forma independiente hasta menos de 10 rostros, se debe también mencionar que una persona puede tener registrado en un solo grupo hasta 248 rostros del mismo sujeto. Al momento de realizar la detección del rostro y la identificación de este, se debe tener en cuenta el método de aprendizaje automático seleccionado, es decir debe ser la misma versión para las dos funciones.

La metodología de desarrollo indica que métodos y técnicas se usará en el desarrollo de un proyecto, haciendo que el proyecto tenga varias fases y orden, las metodologías de desarrollo de un proyecto generalmente constan de 5 componentes, el primer componente son las fases aplicables, la documentación, las técnicas y herramientas, métodos y por último control y evaluación.

Scrum es una metodología ágil, la cual se inició en el año de 1993 aplicándola como un modelo de desarrollo de software. La metodología Scrum por ser una metodología ágil cumple con la creación de ciclos rápidos de desarrollo, lo que se lo llama Sprint o comúnmente conocido como iteración.

Una de las facilidades a la hora de aplicar Scrum, son las revisiones diarias, el cual es uno de los elementos fundamentales, se tiene un ejemplo en la figura 5. (Trigas Gallego, s. f.).

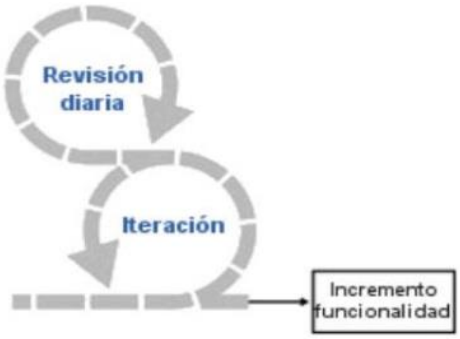

Figura 5. SCRUM(Trigas Gallego, s. f.).

Planificación de backlog: En esta fase se realizará el análisis completo de los requerimientos del aplicativo incluyendo, el análisis de requisitos, funcionales, no funcionales, la factibilidad del sistema, el análisis de costos para el proyecto, los requisitos de hardware y software, es decir se planeará el Sprint 0.

Planeación del sprint: En esta fase se definirá con el cliente el entregable, la forma de realizarlo y se asignará un plazo que se tendrá para tener los resultados de una manera óptima.

Daily scrum: Se puede realizar con el cliente, estados de avance del Sprint conversado anteriormente, con la finalidad de registrar el avance del proyecto y de la fase anterior.

Revisión del sprint: En esta fase se puede analizar los avances, los problemas que se tuvieron durante la implementación, las maneras en que se pueden mejorar, se analiza el producto backlog, para ver el avance final del proyecto, y se realizará un nuevo Sprint para una nueva implementación.(Cadavid, Martínez, & Vélez, 2013)

2.METODOLOGÍA

2. METODOLOGÍA

Para la creación de un sistema informático se debe establecer una metodología que nos ayude en el manejo de los procesos de desarrollo, mejorar las funcionalidades y fases en las que se necesiten resultados óptimos. En este proyecto se utilizó la metodología ágil Scrum, se decidió trabajar con dicha metodología, por la flexibilidad a la hora de realizar cambios y nuevos requisitos que se den a lo largo de la vida del proyecto, también es una metodología que nos permite hacer iteraciones, esto ayuda a la entrega de buenos resultados permitiendo saber el estado del proyecto y obtener una visión global.

El uso de la metodología scrum, tiene varios puntos buenos y malos a tratar tales como:

2.1. VENTAJAS:

- Flexibilidad y adaptación respecto a las necesidades del cliente.
- Se logra obtener un alto índice de calidad en el desarrollo.
- El cliente establece expectativas del aplicativo, haciéndolas tangibles en un corto plazo.

2.2. DESVENTAJAS:

- Si una tarea o fase no está bien definida, el proyecto puede tener retrasos en la fecha de entrega definida.
- Un numero alto de reuniones, muchas de las veces pueden dar malos resultados, por los cambios suscitados en cada reunión dada.

Con el uso de esta metodología, se logró obtener un aplicativo, funcional e intuitivo para el usuario, cabe destacar que se realizaron los Sprint necesarios para cada fase de desarrollo, a continuación, se detalla todo lo que se realizó en cada fase de desarrollo, aplicando la metodología escogida

2.3. FASES DE DESARROLLO

2.3.1. SPRINT 1

Para la primera fase, se establecieron los requisitos funcionales, en los que se detalla los sprint a realizarse, esta primera fase de análisis será nuestro Sprint inicial del proyecto de desarrollo, analizando así todas las funcionalidades que tendrá el aplicativo es decir nuestro product backlog. En la tabla 1 se muestra la descripción de los requerimientos de la fase inicial.

Tabla 1. Fase de análisis

Sprint 1, Fase de análisis	Tiempo: 7 Días
Tarea	Tiempo
Recolectar requerimientos funcionales y no funcionales	1 día
Determinar casos de uso necesarios	3 días
Definición de los flujos de interacción en el sistema	3 días

2.3.2. SPRINT 2

En la fase que corresponde el diseño, se define la interfaz gráfica que se presentará al usuario, el diseño técnico que se usará y la base de datos que se requiere, esto se lo realizó tomando en cuenta los requisitos funcionales y no funcionales tomados en la fase anterior, en la tabla 2 se muestran las actividades realizadas.

Tabla 2. Fase de diseño

Sprint 2, Fase de diseño	Tiempo: 15 Días
Tarea	Tiempo
Diseño de interfaz de usuario	10 días
Diseño técnico de aplicativo	4 días
Diseño base de datos	1 día

2.3.3. SPRINT 3

En esta fase de realizó el desarrollo de los módulos necesarios para el funcionamiento del sistema, de acuerdo con el diseño de la interfaz del usuario, de igual manera se tomó en cuenta los requisitos funcionales de la primera fase, en la tabla 3 se podrá encontrar las actividades y los módulos realizados.

Tabla 3. Fase de desarrollo

Sprint 3, Fase de desarrollo	Tiempo: 51 Días
Tarea	Tiempo
Desarrollo módulo de ingreso al aplicativo y registro de usuarios nuevos.	8 días
Desarrollo modulo para agregar rostros de usuarios a grupo de seguridad.	15 días
Desarrollo módulo de eliminación de usuarios de grupo de seguridad.	8 días
Desarrollo módulo de dispositivos de seguridad con verificación de reconocimiento facial.	10 días
Desarrollo módulo de grupos de seguridad registrados, con sistema de seguridad de reconocimiento facial .	10 días

2.3.4. SPRINT 4

En esta fase se realizan las pruebas funcionales del aplicativo, en donde se analiza si se debe realizar nuevas iteraciones para el funcionamiento óptimo y realizando una retroalimentación de los errores y aciertos obtenidos en la tabla 4 se muestra las actividades realizadas en el sprint.

Tabla 4. Fase de pruebas

Sprint 4, Fase de pruebas	Tiempo: 10 Días
Tarea	Tiempo
Pruebas de calidad de software	10 días

2.4. SOFTWARE PARA EL DESARROLLO DEL APLICATIVO:

- Lenguaje de programación Java
- Android Studio
- Gestor de base de datos SQL server
- API Face Microsoft Azure
- Servicios base de datos Microsoft Azure
- Arduino

El software descrito nos servirá en diferentes fases, el lenguaje de programación Java, será el lenguaje que usaremos para el desarrollo de la aplicación en el entorno de desarrollo Android Studio , el Gestor de base de datos de SQL server, nos servirá para la creación de la base de datos a usar para el registro de usuarios, el cual se usara por medio de los servicios de bases de datos de Microsotf Azure, y por último, el entorno de desarrollo Arduino, para la configuración del dispositivo inteligente.

3.RESULTADOS Y DISCUSIÓN

3. RESULTADOS Y DISCUSIÓN

3.1. ANÁLISIS

El aplicativo se basó principalmente en el uso de la biometría como medio de seguridad, en este caso fue el reconocimiento facial, se escogió este método de biometría, por ser un método no invasivo con el usuario al no tener un contacto directo, esta seguridad es utilizada para el manejo de dispositivos bluetooth, para uso concreto se realizaron las pruebas con un módulo HC-05, el cual se conectó a una placa electrónica Arduino, y ella a una chapa eléctrica.

Para la implementación, se realizó un análisis de todas las tecnologías de biometría de reconocimiento facial, haciendo una comparación con todos sus componentes, las tecnologías que se compararon fueron 3 principalmente, Microsoft Azure API Face, Google API Vision AI y OpenCV, en la tabla 5 se muestra el análisis tomado de las tecnologías.

Tabla 5. Análisis de tecnologías de reconocimiento facial

	Microsoft Azure API Face	OpenCV	Google API Vision AI
Detección Facial	SI	SI	SI
Reconocimiento facial en una imagen	SI	NO	NO
Reconocimiento facial en un video	SI	SI	NO
SDK	NO	SI	NO
API	SI	NO	SI
Facilidad de uso	SI	NO	SI
Costo Transacción	0,000665517	0	0.0014
OpenSource	NO	SI	NO
Aplicabilidad al caso de estudio	SI	SI	NO
Uso de hardware	Bajo	Alto	Bajo

En el análisis de las características de cada tecnología, se puede evidenciar que, para el proyecto es más conveniente Microsoft Azure API Face, por tener las características más completas. Después del análisis realizado, Google API Vision no dispone de Reconocimiento facial tanto para imágenes y video, por lo que no brinda el soporte necesario, además el costo por transacción es el más costoso comparando las 3 tecnologías, mientras que OpenCV es una herramienta más compleja, al no contar con un API para la implementación , no dispone un reconocimiento facial por medio de imágenes lo que complica su uso, una de las ventajas que brinda OpenCV tras ser OpenSource es el costo cero de su uso, la documentación para el desarrollo del aplicativo con esta tecnología aplicada a dispositivos Android no son apropiados y tienden a cambiar con las versiones de las librerías según el informe (Soler, 2014), detalla además que para el uso de las librerías de OpenCv es necesario descargar un aplicativo adicional “Opencv Manager”, el cual ya no se

encuentra disponible en la PlayStore, adicionalmente se menciona en muchos trabajos de investigación que el uso de estas librerías tiene una alta demanda de recursos de hardware, tal como el almacenamiento del dispositivo.(CÁCERES, 2018)

Se analizó el costo por transacción que tiene cada tecnología, la API de Microsoft Azure, la que ofrece 30000 transacciones mensuales sin costo, en la duración del proyecto se revisaron las tarifas del año 2020 en el que se dio uso el aplicativo, y no se ha concurrido a pagar costos por el servicio de Reconocimiento Facial como se muestra en la figura 6.

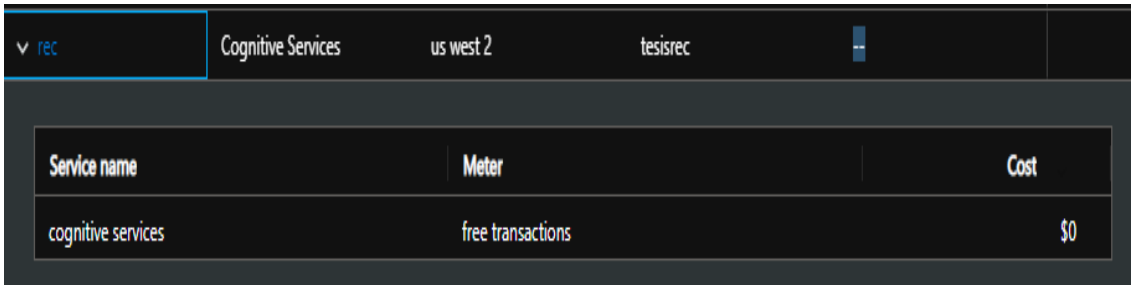

Figura 6 Costos reconocimiento facial Microsoft Azure.

API Face de Microsoft Azure, Brinda muchas facilidades, al contar con buena documentación, tiene una facilidad de uso por tener varios repositorios que ayudan a la compresión de los métodos a aplicar, además tiene el soporte necesario para el reconocimiento facial por medio de una imagen lo que significa que es de gran ayuda a la aplicabilidad en el proyecto.

3.1.1. ARQUITECTURA

La arquitectura usada en este trabajo fue cliente-servidor, el dispositivo móvil actuó como cliente, y los servicios de Microsoft Azure se usaron como servidor, aquí se procesaron las peticiones del ingreso al aplicativo y también los servicios de reconocimiento facial.

En la figura 7 vemos la arquitectura que toma el proyecto, en donde tenemos dos servicios que son administrados por Microsoft Azure, el primer servicio utilizado es la API Face, la cual servirá para el reconocimiento facial, la información obtenida a partir del API se enviará para su debido procesamiento usando los protocolos detallados en el siguiente punto.

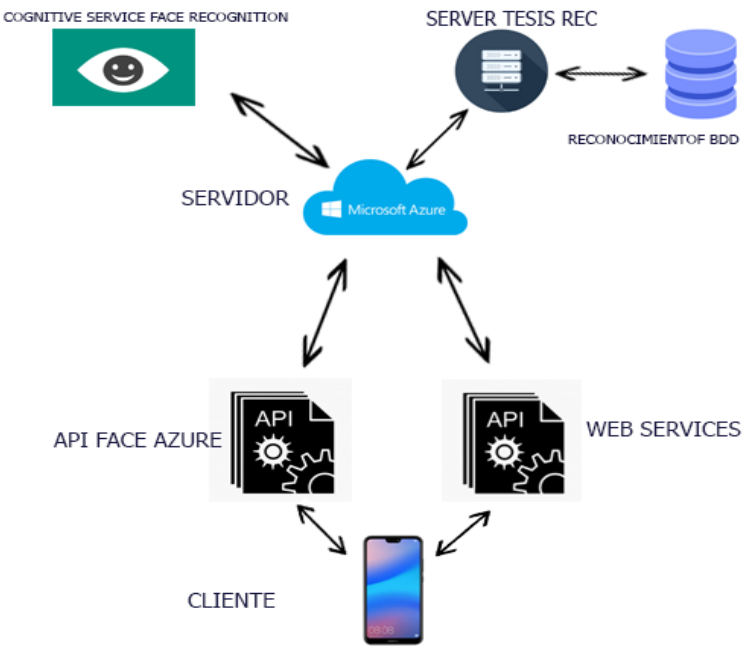

Figura 7. Arquitectura reconocimiento facial.

Además, se utilizará una base de datos, la cual se encuentra alojada en un servidor de Microsoft Azure, en la que podremos enviar la información y recibirla por medio de web services, a continuación, se detallará la base de datos utilizada en el proyecto.

3.1.1.1. Protocolos

Para el desarrollo del aplicativo, se obtuvo varios métodos que son usados por medio de una API de Microsoft Azure, el cual maneja el protocolo de transferencia de hipertexto(HTTP) al ser tratado como un API REST, es decir realiza los envíos, obtención, eliminación de los datos, por medio de los métodos, GET, POST, DELETE, etc.

3.1.1.2. Azure SQL Server

El uso principal de la base de datos es el ingreso de los usuarios registrados en el aplicativo, en esta base se alojará la información del grupo al que pertenece el usuario, las credenciales como el correo, la clave de ingreso y el nombre del usuario. En la figura 8 se aprecia la estructura de la tabla que se encuentra alojada en los servicios de Azure SQL Server.

	Column Name	Data Type
🔑	codigo	int
	nombres	varchar(20)
	correo	varchar(50)
	contrasenia	varbinary(MAX)
	grupo	varchar(50)

Figura 8 Base de datos tabla usuarios.

3.1.1.3. Consideraciones en el uso de API FACE

Para el uso del API FACE de Microsoft Azure, se considera la seguridad que se maneja para el enlace de los métodos, se debe tener un API Key, asegurando el acceso a la API y los datos alojados en ella.

Se debe tomar en cuenta que la precisión que se tenga con API Face, es relativo, depende de cómo se lo implemente en el sistema, la tecnología usada para el desarrollo y la forma como está configurada, y por ultimo los factores y condiciones ambientales que hacen que varíe la precisión. Se debe tener presente, que la herramienta de Microsoft Azure no predice la edad o sexo de un sujeto como rasgo para la identificación. La imagen para el sondeo con los rostros almacenados y su resultado son proporcionales a la calidad de la imagen enviada para su reconocimiento.

3.1.1.4. Cognitive Face Recognition

El servicio que ofrece Microsoft Azure Face, emplea algoritmos para detectar rostros, reconocerlos y analizarlos, este servicio proporciona varias funciones, la primera función es la detección de rostros, el cual realiza el análisis de una imagen y reconoce los rostros humanos que puede haber en ella, devolviendo las coordenadas del rectángulo en donde se encuentren, si se desea, se puede extraer varios elementos de la imagen, como puede ser la posición de la cabeza, el sexo, la edad, emociones faciales, vello facial y accesorios usados en el rostro.

La siguiente función que podemos apreciar es la verificación de los rostros, en donde se realiza la autenticación a partir de un rostro detectado, evaluando si pertenecen a un rostro ya registrado en el sistema, esta función es ocupada en la aplicación.

La búsqueda de caras similares es otra de las funciones que se agregan en este servicio, la cual compara un grupo de rostros buscando el más semejante al sujeto, existen dos formas, la primera es matchFace que devuelve una lista de caras similares que pueden o no pertenecer al mismo sujeto, y la otra función es matchPerson, devuelve todos los rostros parecidos al filtrar entre rostros del mismo sujeto.

La agrupación de caras es la función encargada de dividir un grupo de rostros desconocidos en función de su semejanza, haciendo que los grupos formados sean de un solo sujeto.

3.1.1.5. Detección de imágenes

Para realizar el reconocimiento de un rostro a un sujeto se debe tener un dato de entrada, en el cual se tiene varios lineamientos a seguir:

- Los formatos de la imagen de entrada deben ser JPEG,PNG,GIF(se tomará en cuenta el primer fotograma) y BMP.
- El tamaño del archivo que se usara como dato de entrada no debe exceder de 4MB
- Cuando se crea un objeto Person, para la creación de un usuario, se debe usar datos de entrada que tengan distintos tipos de ángulos e iluminación para una mejor precisión.
- Se debe tomar en cuenta que los datos de entrada deben ser lo más claros y concisos posibles, cualquier afectación, como iluminación extrema, contraluz, obstáculos que bloqueen los ojos, diferencias en el tipo de vello facial, cambios de apariencia por la edad del sujeto y expresiones faciales extremas pueden causar repercusiones en el reconocimiento del sujeto.

3.1.1.6. Recolección de información

Para la recolección de la información, se debe tomar en cuenta las funciones utilizadas en el aplicativo, las cuales extraen los datos necesarios para la identificación del sujeto. En la figura 9, se muestra los pasos que se realiza para la recolección de la información y el envío de la información.

Figura 9 Procedimiento Recolección de datos.

En la captura de imagen, se toma en cuenta que el aplicativo, realiza tres tomas de imágenes para el reconocimiento del sujeto, captura el nombre que tendrá de identificativo el rostro, extrae de las imágenes el rostro centrado en un recorte de la imagen original, se realiza la detección y añade con un nuevo identificativo al rostro encontrado, por último, se debe entrenar nuevamente el grupo al que pertenece este rostro.

3.1.1.7. Almacenamiento de información

En el almacenamiento de la información enviada, en este caso la imagen con el rostro definido, se realiza un procesamiento para la obtención de la información necesaria para el reconocimiento facial, realizado el

procesamiento de la imagen, es eliminada, Microsoft Azure maneja directivas de seguridad para el manejo de la información, siguiendo el RGPD(Reglamento general de protección de datos).

Los datos que son almacenados son alojados en un servidor de preferencia a la región perteneciente, en el proyecto se escogió el servidor que se detalla en la figura 10.

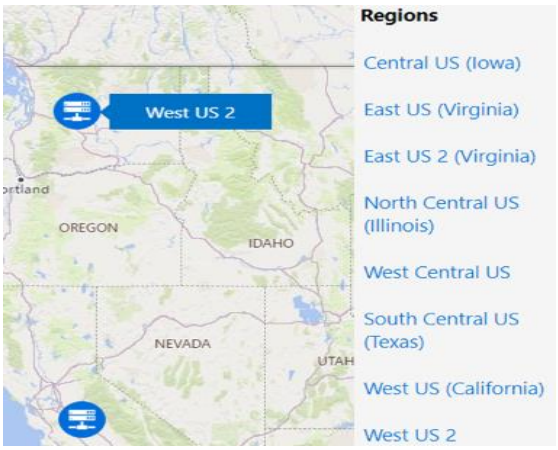

Figura 10 Datacenter Microsoft Azure.(«Azure Datacenters», s. f.)

Se debe tomar en cuenta que los formatos para el envío de la información deben cumplir con los especificados en la captura de la imagen, es decir:

- Los formatos de la imagen de entrada deben ser JPEG,PNG,GIF(se tomará en cuenta el primer fotograma) y BMP.
- El tamaño del archivo que se usara como dato de entrada no debe exceder de 4MB

3.1.1.8. Procesos de aprendizaje automático

Los procesos de aprendizaje automático en el servicio de Microsoft Azure, se usa comúnmente para realizar operaciones en caras humanas de imágenes dispuestas al análisis, en la actualidad existen 3 versiones del servicio de Azure Face, los cuales son netamente compatibles, recognition_01 publicad a en 2017, recognition_02 publicado en 2019 y recognition_03 publicado en 2020 son las versiones que actualmente cuenta los servicios de Microsoft.

El proceso de solución de problemas que maneja Microsoft Azure de manera general tiene 4 pasos definidos, los cuales se detallan en la figura 11.

Figura 11 Aprendizaje automático(«Microsoft Azure», s. f.)

En el primer paso del proceso de aprendizaje automático se debe identificar el origen de los datos, los cuales ayudan a determinar que algoritmo de aprendizaje se debe dar uso, en el momento de revisar los datos se identifica alguna anomalía que pueden tener, se procede al desarrollo de una estructura, la cual se encarga de la resolución de los problemas que puede tener de integridad de los datos.

En el siguiente proceso, los datos se deben dividir en dos tipos, en los datos de aprendizaje y los datos que se ingresaran de prueba, gran parte de los datos que se usaran para dicho aprendizaje, son datos usados para el ajuste de los modelos logrando alta precisión.

Continuando con el proceso de aprendizaje automático, cuando está listo el conjunto de datos a usar como modelo, se usa el conjunto de datos de prueba, evaluando el rendimiento y precisión del modelo.

El último paso que se debe seguir es la interpretación de los datos, en los que se revise la información para sacar conclusiones y predecir los resultados.(«Microsoft Azure», s. f.)

3.1.2. MÓDULOS APLICATIVO

Se definieron varios módulos que son esenciales para la funcionalidad completa del aplicativo:

- Inicio de sesión
- Creación de usuario
- Agregar rostros
- Eliminar usuarios
- Agregar dispositivos de seguridad
- Agregar grupos de seguridad
- Cerrar sesión.

Se definen los casos de uso pertenecientes a los módulos descritos.

El módulo de ingreso al aplicativo y registro de usuarios nuevos, permite la verificación del usuario con algoritmos que validan el usuario y contraseña en la base de datos, en la figura 12 se define el diagrama de casos de uso del módulo de ingreso del aplicativo.

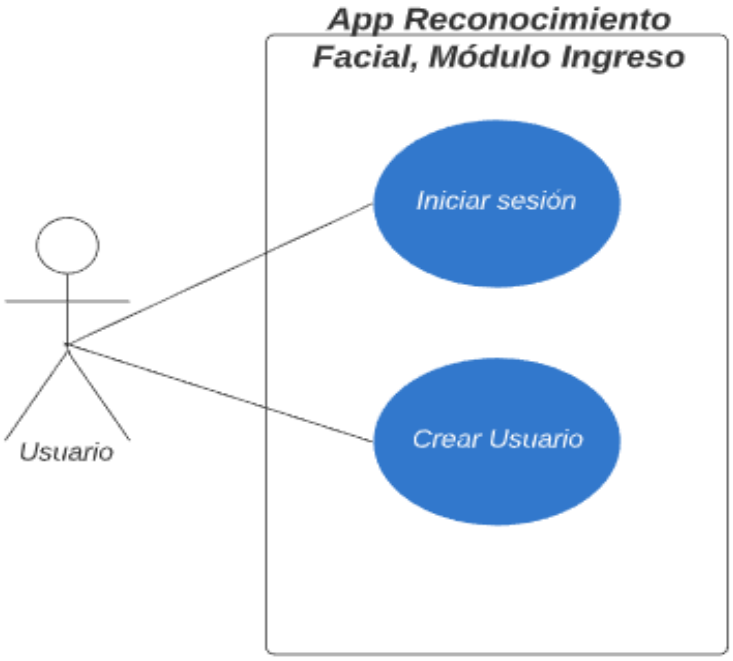

Figura 12. Casos de uso módulo de ingreso

En la tabla 6, se describe las distintas acciones que toma el módulo de inicio de sesión.

Tabla 6. Casos de uso de iniciar sesión

Caso de uso:	**Iniciar sesión**
Resumen: Usuario deberá ingresar su correo y contraseña, para la validación en la base de datos del aplicativo y proceder con el ingreso al menú principal.	
Actor: Usuario	
Responsabilidad de los actores	**Responsabilidades sistema**
1. El usuario deberá ingresar el correo registrado anteriormente en el aplicativo.	
2. El usuario debe ingresar la contraseña definida para el aplicativo.	3. El aplicativo Valida la contraseña ingresada por el usuario comparando con la contraseña alojada en la base de datos
	4. El aplicativo debe validar el correo electrónico ingresado con la variable de la base de datos.
	5. El aplicativo debe guardar nombre de usuario ingresado en un archivo temporal para el uso posterior.
	6. El aplicativo debe consultar a que grupo es perteneciente el usuario que inicio la sesión.
	7. El aplicativo, realizada la validación de los datos, debe guardar credenciales para iniciar de manera automática en la próxima apertura del aplicativo.
	8. Enviar a menú de aplicación

En la tabla 7 se describe el funcionamiento del módulo de creación de usuario, las credenciales ingresadas se dan uso en varios módulos.

Tabla 7. Casos de uso creación usuario

Caso de uso:	**Crear Usuario**
Resumen: Módulo que crea una cuenta nueva en la aplicación y la creación de un grupo de seguridad para el uso del reconocimiento facial y los dispositivos de seguridad.	
Actor: Usuario	
Responsabilidad de los actores	**Responsabilidades sistema**
1. Ingresar el nombre que tendrá el usuario nuevo en el aplicativo .	
2. Ingresar el correo electrónico que servirá como credencial.	
3. Ingresar una contraseña nueva.	
4. Repetir la contraseña ingresada anteriormente para validar los campos.	5. El aplicativo valida que los campos estén llenos con la información requerida.
	6. El aplicativo valida si la cuenta ya existe, verificándola con la base de datos y el correo electrónico.
	7. El aplicativo valida si las contraseñas coinciden para el registro en la base de datos
	8. El aplicativo, procede a registrar los datos en la base, previa validación.
	9. El aplicativo envía al módulo de iniciar sesión.

En la figura 13 se definen todos los casos de uso del aplicativo.

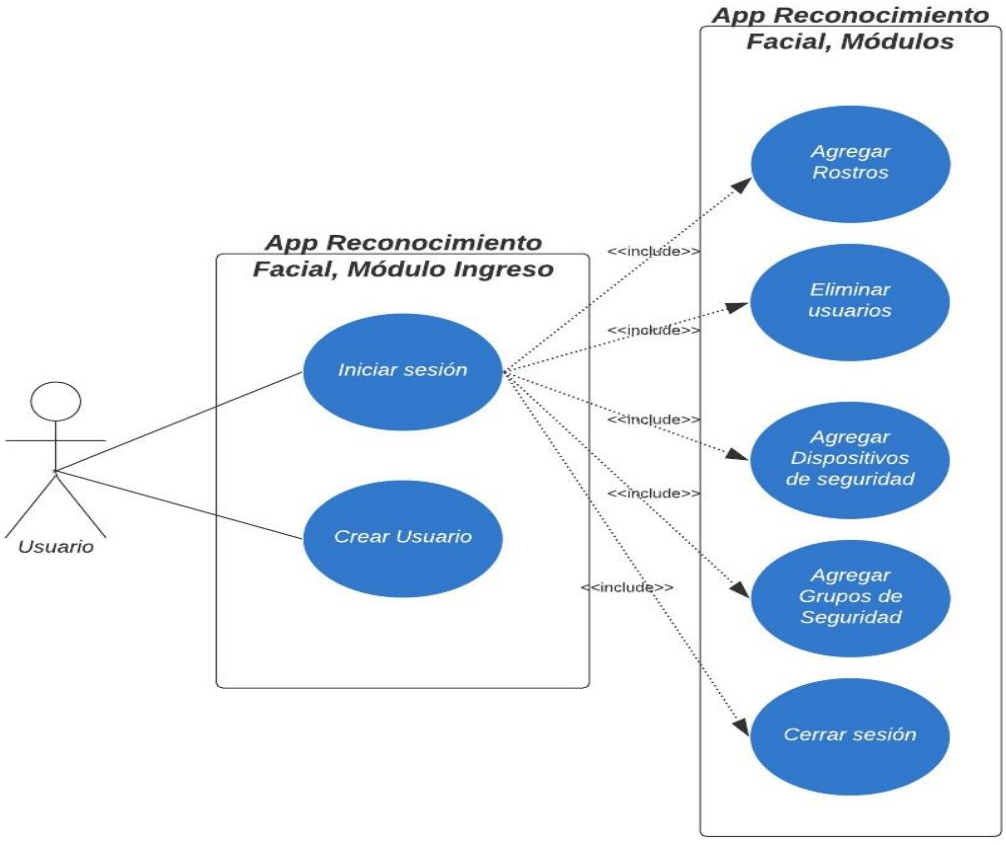

Figura 13. Casos de uso, módulos del aplicativo

En la tabla 8 se observa el uso del módulo de agregar rostros, en este módulo se inicia el reconocimiento facial.

Tabla 8. Casos de uso agregar rostro

Caso de uso:	**Agregar Rostros**
Resumen: Módulo que registra rostros para el uso de los dispositivos de seguridad, en el cual se debe ingresar 3 rostros y el nombre de la persona para la identificación.	
Actor: Usuario	
Responsabilidad de los actores	**Responsabilidades sistema**
1. El usuario debe ingresar el nombre que tomara el rostro nuevo.	
2. Usuario debe registrar la cara del usuario nuevo en el primer apartado.	
3. Usuario debe registrar la cara del usuario nuevo en el segundo apartado.	
4. Usuario debe registrar la cara del usuario nuevo en el tercer apartado.	5. El aplicativo debe validar que los campos no estén vacíos.
	6. El aplicativo detecta los rostros en las tres imágenes que servirán de entrenamiento.
	7. El aplicativo añade el rostro al grupo de seguridad para identificarlo.
	8. El aplicativo realiza el entrenamiento del rostro agregado al grupo de seguridad.

En la tabla 9, se describe las acciones correspondientes a la eliminación de los rostros agregados en el grupo de seguridad.

Tabla 9. Caso de uso eliminar usuario

Caso de uso:	**Eliminar usuarios**
Resumen: Módulo que elimina el usuario seleccionado, el cual pertenece al grupo de seguridad, en el que se registra su rostro.	
Actor: Usuario	
Responsabilidad de los actores	**Responsabilidades sistema**
1. El usuario debe seleccionar a que rostro desea eliminar del grupo de seguridad	
	2. El aplicativo verifica a que grupo pertenece usuario que se eliminara.
	3. El aplicativo verifica la identificación del usuario en el grupo que se eliminara
	4. El aplicativo elimina el rostro del usuario del grupo de seguridad.

En la siguiente tabla 10, se describen las acciones del módulo de dispositivos de seguridad.

Tabla 10. Casos de uso agregar dispositivos de seguridad

Caso de uso:	Agregar dispositivos de seguridad
Resumen: Módulo que permite el uso de dispositivos de seguridad, los cuales están vinculados al teléfono vía bluetooth.	
Actor: Usuario	
Responsabilidad de los actores	**Responsabilidades sistema**
1. El usuario debe tomar foto de su rostro para identificarlo.	
	2. El aplicativo Detecta el rostro del usuario a identificar.
	3. El aplicativo Verifica el rostro del usuario en el grupo de seguridad
	4. El aplicativo envía a la actividad de los dispositivos vinculados.
5. El usuario debe seleccionar el dispositivo que se usara.	
6. El usuario procede a Bloquear o desbloquear el dispositivo de seguridad.	7. El aplicativo envía una señal al dispositivo de seguridad seleccionado para realizar la acción pertinente.

En la tabla 11 se describe las acciones del módulo grupos de seguridad.

Tabla 11. Casos de uso agregar grupos de seguridad

Caso de uso:	Agregar grupos de seguridad
Resumen: Módulo que permite vincular grupos de seguridad a los que pertenece un usuario y compartir el grupo de este por medio de código QR.	
Actor: Usuario	
Responsabilidad de los actores	**Responsabilidades sistema**
1. Usuario debe tomar una foto de su rostro para identificarlo	
	2. El aplicativo detecta el rostro a identificar el usuario.
	3. El aplicativo verifica el rostro en un grupo de seguridad que pertenezca el usuario.
	4. El aplicativo envía a nueva actividad de grupos de seguridad
5. El usuario debe pulsar en compartir grupo de seguridad.	6. El aplicativo muestra un código QR para poder realizar el vínculo con otro dispositivo.
7**.** El usuario debe pulsar en Ingresar a grupo de seguridad.	8. El aplicativo escanea el código QR para vincular usuario en grupo de seguridad.

En la tabla 12 se describe las acciones usadas en el módulo de cierre de sesión.

Tabla 12. Casos de uso cerrar sesión

Caso de uso:	Cerrar sesión
Resumen: Cierra la sesión, limpiando las credenciales almacenadas en archivos temporales del aplicativo, y envía al módulo de iniciar sesión	
Actor: Usuario	
Responsabilidad de los actores	**Responsabilidades sistema**
1. El usuario debe pulsar en cerrar sesión	
	2. El aplicativo limpia las credenciales almacenadas en archivos temporales de la sesión
	3. El aplicativo envía nuevamente al módulo de inicio de sesión para ingreso de usuario nuevo.

3.2. DISEÑO DE LA INTERFAZ

El diseño de la interfaz se realizó lo más simple posible para el usuario, haciéndolo más amistoso y funcional, se toma en cuenta los colores y la eficiencia en la legibilidad en los textos usados.

como se muestra en la figura 14 el aplicativo posee una ventana correspondiente al SPLASH SCREEN, se presenta el logo del aplicativo.

Figura 14. Interfaz splash screen

En la figura 15, se muestra el inicio del aplicativo, donde se ingresará el correo y la contraseña registrados anteriormente.

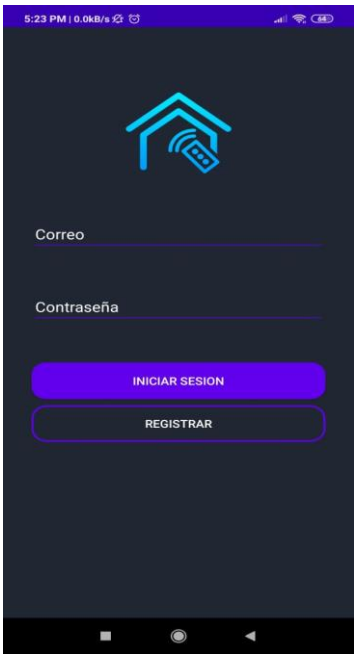

Figura 15. Interfaz inicio sesión

En la figura 16, se registran los datos para iniciar un nuevo usuario en el aplicativo, se pide el nombre del usuario, la contraseña y un correo electrónico para poder ingresar al aplicativo.

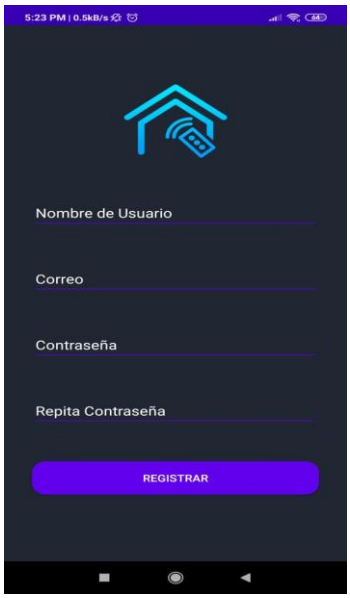

Figura 16. Interfaz creación usuario

En la figura 17, se muestra la pantalla inicial que sirve de menú, para todas las opciones que tendrá el aplicativo, tal como los dispositivos de seguridad, los grupos de seguridad, agregar usuarios, modificar grupo y por último nuestra barra de navegación que tendrá un Icono de ayuda y cerrar sesión.

Figura 17. Interfaz Menú aplicativo

En la figura 18,se observa la interfaz de identificar un usuario, se pedirá una fotografía del rostro para el uso.

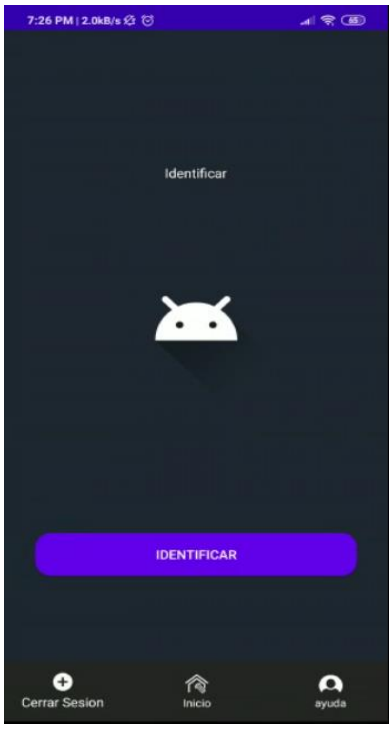

Figura 18. Interfaz identificación usuario

En la figura 19, se muestra los dispositivos vinculados para su selección.

Figura 19. Interfaz mis dispositivos

En la figura 20, se observa las funcionalidades que tenemos dentro del dispositivo de seguridad seleccionado, tenemos el bloqueo y desbloqueo del dispositivo.

Figura 20. Interfaz bloqueo desbloqueo dispositivo

En la figura 21, se muestran los grupos de seguridad a los que fue emparejado la aplicación, tenemos la opción de compartir nuestro grupo e ingresar a uno.

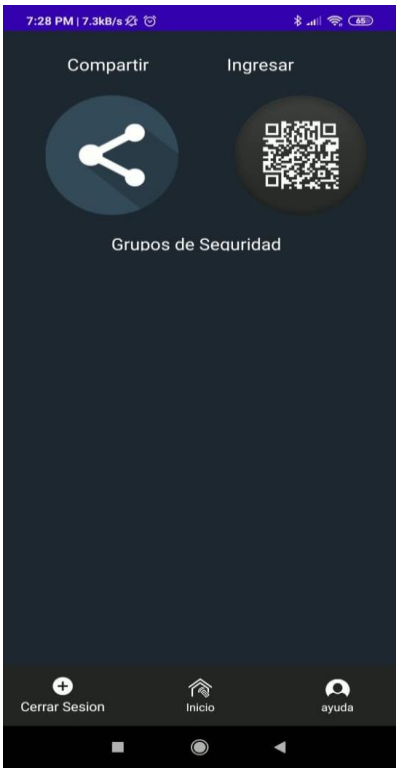

Figura 21. Interfaz grupos de seguridad

En la figura 22, se muestra la interfaz de código QR perteneciente al grupo de seguridad.

Figura 22. Interfaz compartir grupos de seguridad

En la figura 23, se muestra la interfaz de ingresar a un grupo de seguridad escaneando un código QR.

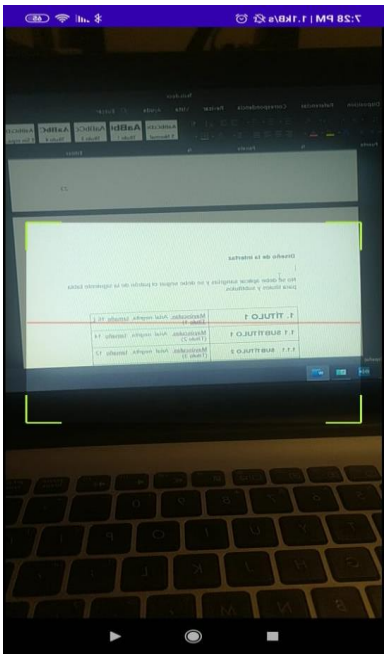

Figura 23. Interfaz ingresar grupos de seguridad

En la figura 24,se observa la interfaz en la que almacena el grupo escaneado con el código QR.

Figura 24. Interfaz enlace grupo de seguridad

En la figura 25, se observa la interfaz de agregar un usuario, en el cual nos pedirá el nombre para identificar el rostro y las tres imágenes para el reconocimiento del rostro.

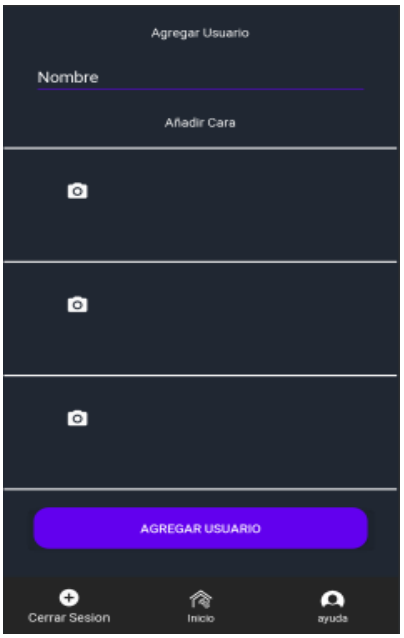

Figura 25. Interfaz agregar usuario

En la figura 26, se observa la interfaz perteneciente a la opción de modificar grupo.

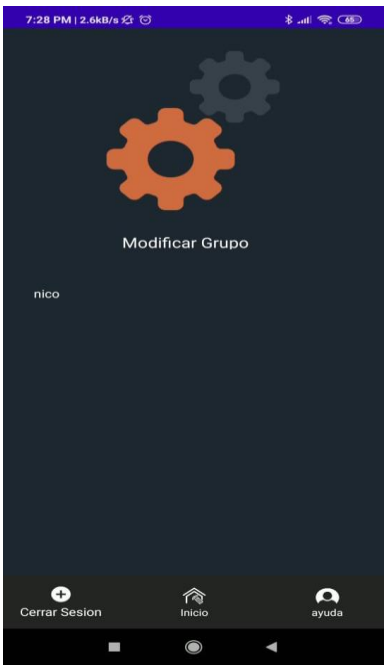

Figura 26. Interfaz modificar grupo

En la figura 27, se observa la interfaz perteneciente a la opción de eliminar el usuario seleccionado en el módulo de modificar grupo.

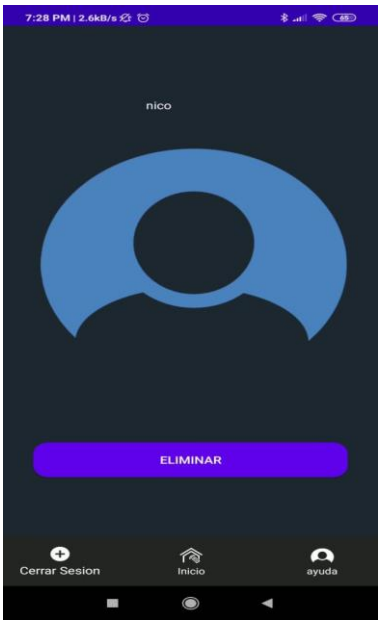

Figura 27. Interfaz eliminar usuario

3.3. DESARROLLO

3.3.1. MÓDULO DE INGRESO AL SISTEMA Y CREACIÓN DE USUARIO

En la figura 28, se muestra el diagrama de flujo correspondiente al ingreso al sistema de un usuario registrado en el aplicativo. En el anexo 1, se puede ver el algoritmo usado para su funcionalidad.

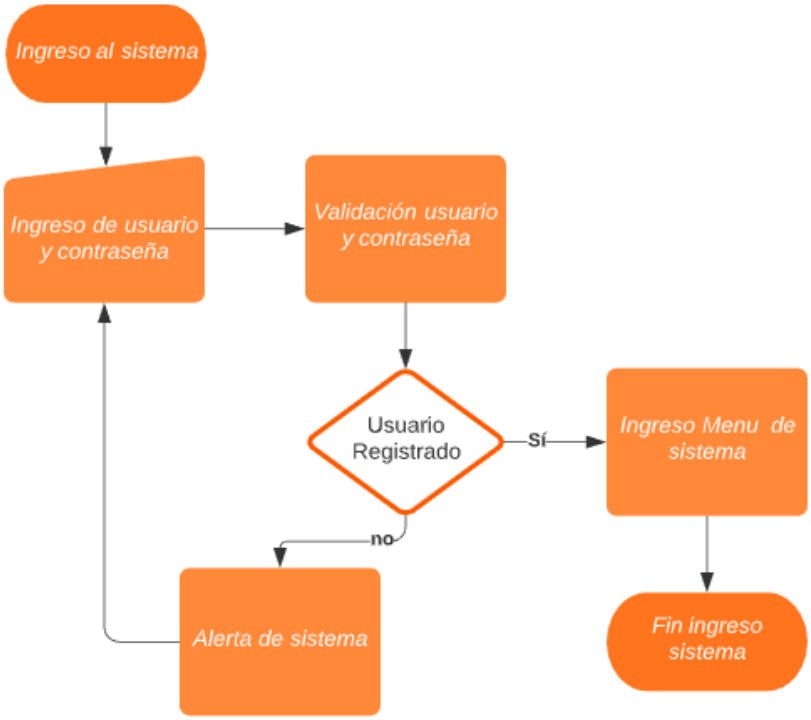

Figura 28. Diagrama de flujo ingreso al sistema

En la figura 29 se muestra, el diagrama de flujo de la creación de un usuario en el aplicativo, en el anexo 2 se muestra además el algoritmo usado para la creación de usuario.

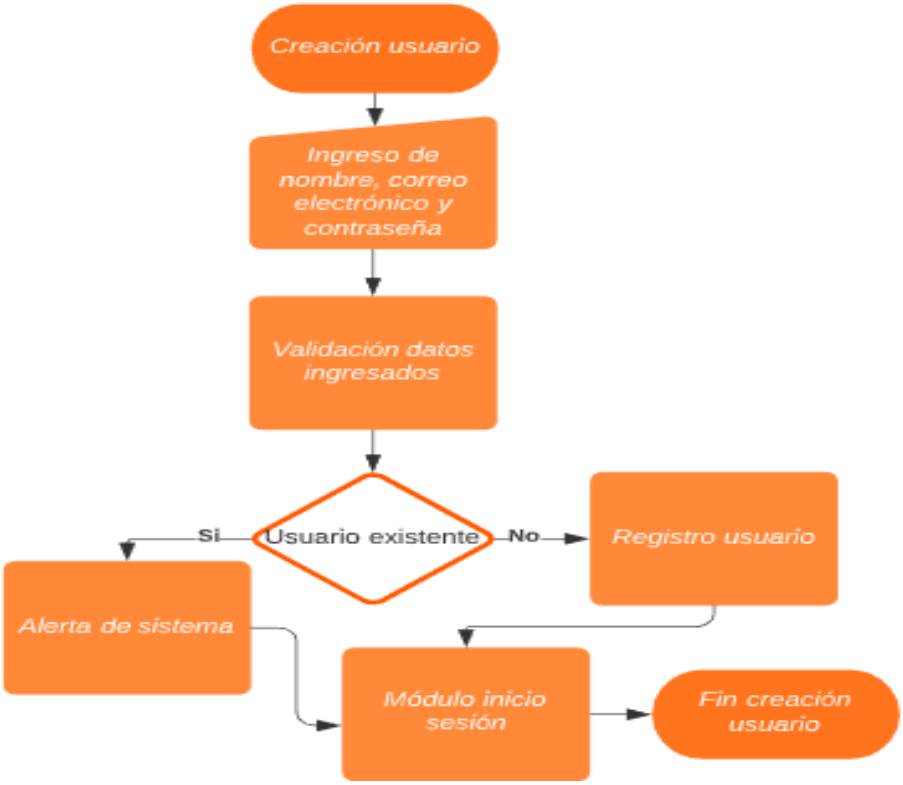

Figura 29. Diagrama de flujo creación de usuario.

3.3.2. MÓDULO PARA AGREGAR ROSTRO DE USUARIO

En la figura 30, se muestra el diagrama de flujo de la creación de un rostro en el grupo de seguridad, en el anexo 3, se muestra el algoritmo pertinente para la funcionalidad.

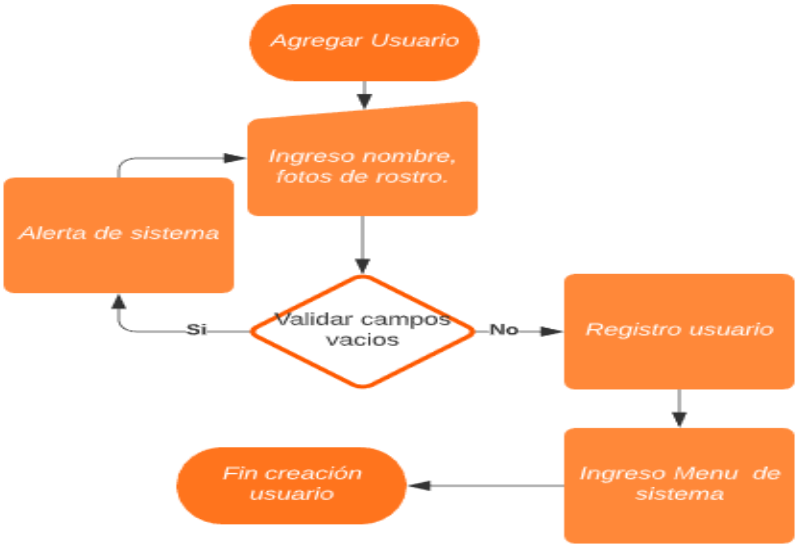

Figura 30. Diagrama de flujo agregar usuario.

3.3.3. MÓDULO MODIFICAR GRUPO DE SEGURIDAD

En la figura 31, se muestra el diagrama de flujo correspondiente a la modificación de grupo de seguridad, para la eliminación de un rostro de usuario, en el anexo 4, se muestran los algoritmos necesarios para su funcionamiento.

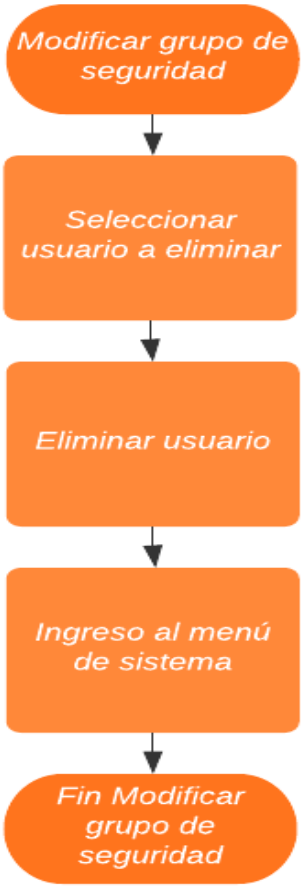

Figura 31. Diagrama de flujo modificar grupo de seguridad

3.3.4. MÓDULO DE DISPOSITIVOS DE SEGURIDAD

En la figura 32, se muestra el diagrama de flujo, que realiza la funcionalidad de los dispositivos de seguridad, además, en el anexo 5 se muestran los algoritmos necesarios para su funcionamiento.

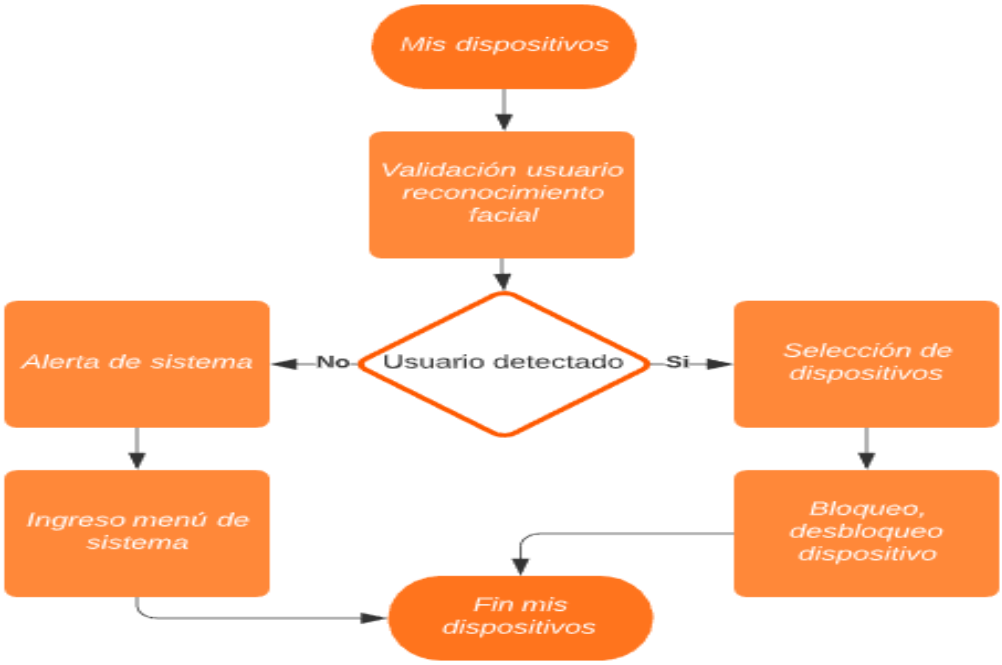

Figura 32. Diagrama de flujo mis dispositivos

3.3.5. MÓDULOS DE GRUPOS DE SEGURIDAD

En la figura 33 se muestra el diagrama de flujo correspondiente al módulo de grupos de seguridad, en el anexo 6, se muestran los algoritmos necesarios para el funcionamiento de este.

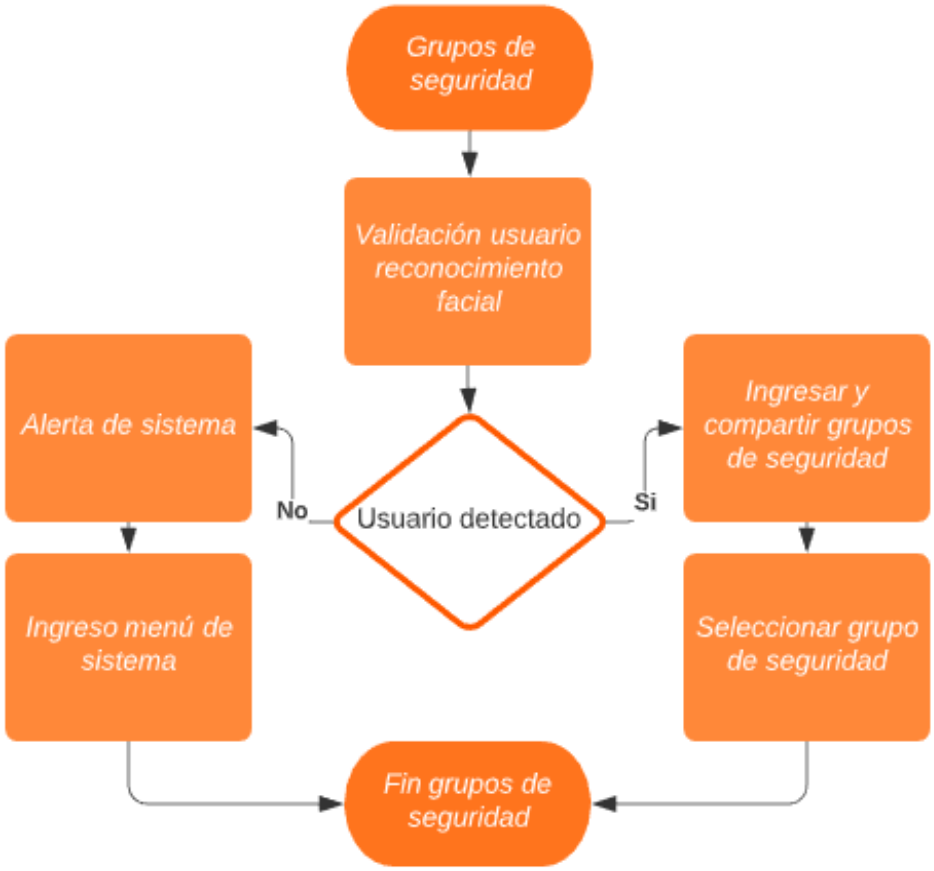

Figura 33. Diagrama de flujo grupos de seguridad

3.4. PRUEBAS

En esta fase, se hicieron las pruebas de funcionalidad necesarias para el aplicativo, es así como se definieron las pruebas por módulos para lograr mejor control y calidad de los procesos, en la tabla 13 se muestra las pruebas realizadas para el ingreso al sistema.

Tabla 13. Pruebas funcionales Ingreso sistema

Id caso de prueba	Módulo a probar	Descripción de caso	Pre-requisitos	Resultado esperado	Resultado obtenido	Estado
CP001	Ingreso sistema	Verificar el inicio de sesión	Ingresar credenciales	OK	OK	Concluido
CP002	Ingreso sistema	Verificar la creación de usuario	Ingresar datos para registro	OK	OK	Concluido
CP003	Ingreso sistema	Verificar que se graben archivos temporales	Ingresar datos en Inicio sesión	OK	OK	Concluido
CP004	Ingreso sistema	Verificar campos vacíos	Ingresar datos e ignorar una casilla	OK	OK	Concluido
CP004	Ingreso sistema	Creación usuario repetido	Ingresar datos ya registrados	OK	OK	Concluido
CP004	Ingreso sistema	Verificar que se cierre sesión y credenciales	Cerrar sesión	OK	OK	Concluido

En la tabla 14 se pueden ver las pruebas que se realizaron al módulo de agregar un rostro de usuario, en donde se validaron y verificaron los campos descritos en esta.

Tabla 14. Pruebas funcionales agregar rostro

Id caso de prueba	Módulo a probar	Descripción de caso	Pre-requisitos	Resultado esperado	Resultado obtenido	Estado
CP001	Agregar Rostro	Verificar campos vacíos	Ingresar datos requeridos e ignorar una casilla	OK	OK	Concluido
CP002	Agregar Rostro	Verificar ingreso de rostros	Ingresar datos faciales	OK	OK	Concluido
CP003	Agregar Rostro	Verificar detección de rostro		OK	OK	Concluido
CP004	Agregar Rostro	Verificar Reconocimiento Facial	Detección de rostros exitoso	OK	OK	Concluido
CP005	Agregar Rostro	Verificar usuario creado	Reconocimiento facial exitoso	OK	OK	Concluido

En la tabla 15 se describen las pruebas realizadas en el módulo de grupos de seguridad, en donde seleccionamos como tal el usuario a modificar, se detallan las pruebas realizadas.

Tabla 15. Pruebas funcionales modificar grupo de seguridad

Id caso de prueba	Módulo a probar	Descripción de caso	Pre-requisitos	Resultado esperado	Resultado obtenido	Estado
CP001	Modificar Grupo	Verificar usuarios registrados en grupo	Iniciar sesión	OK	OK	Concluido
CP002	Modificar Grupo	Selección de usuario a modificar	Seleccionar de lista usuario	OK	OK	Concluido
CP003	Modificar Grupo	Eliminar usuario de grupo de seguridad	-	OK	OK	Concluido

En la tabla 16 se muestran las pruebas funcionales realizadas en los dispositivos de seguridad, con los que son necesarios tener acciones en las cuales se debe tener control y una seguridad optima, es así como se realizan pruebas de identificación de usuarios registrados en el grupo de seguridad, los cuales se detallan en esta.

Tabla 16. Pruebas funcionales dispositivos de seguridad

Id caso de prueba	Módulo a probar	Descripción de caso	Pre-requisitos	Resultado esperado	Resultado obtenido	Estado
CP001	Dispositivos de seguridad	Identificar usuario	Tomar Foto de rostro para identificación	OK	OK	Concluido
CP002	Dispositivos de seguridad	Activar Bluetooth dispositivo móvil	Identificación de usuario correcto	OK	OK	Concluido
CP003	Dispositivos de seguridad	Selección de dispositivo a usar	Tener registrado dispositivos vinculados	OK	OK	Concluido
CP004	Dispositivos de seguridad	Bloqueo de dispositivo de seguridad	Selección de dispositivo exitoso	OK	OK	Concluido
CP005	Dispositivo de seguridad	Desbloqueo de dispositivo de seguridad	Selección de dispositivo existo	OK	OK	Concluido

En la tabla 17 se muestra las pruebas funcionales realizadas en el módulo de dispositivos de seguridad, en donde se probó las formas de compartir el grupo de seguridad y la forma de ingresar a un grupo de seguridad, el usuario debe estar anexado a un grupo de seguridad y tener registrado su rostro para que su funcionamiento sea válido.

Tabla 17. Pruebas funcionales Grupos de seguridad

Id caso de prueba	Módulo a probar	Descripción de caso	Pre-requisitos	Resultado esperado	Resultado obtenido	Estado
CP001	Grupos de seguridad	Identificar usuario	Tomar foto de rostro para identificación	OK	OK	Concluido
CP002	Grupos de seguridad	Compartir grupo de seguridad	Usuario identificado correctamente	OK	OK	Concluido
CP003	Grupos de seguridad	Agregar Grupo de seguridad	Usuario identificado Correctamente	OK	OK	Concluido
CP004	Grupos de seguridad	Selección de Grupo a desbloquear dispositivo	Grupo de seguridad agregado correctamente	OK	OK	Concluido

3.4.1. CONEXIONES PLACA ARDUINO

Para efecto práctico del aplicativo, se realizó un diagrama correspondiente a las conexiones del Arduino uno, con el módulo hc-05 Bluetooth y la chapa eléctrica, en la figura 34 se describe la conexión.

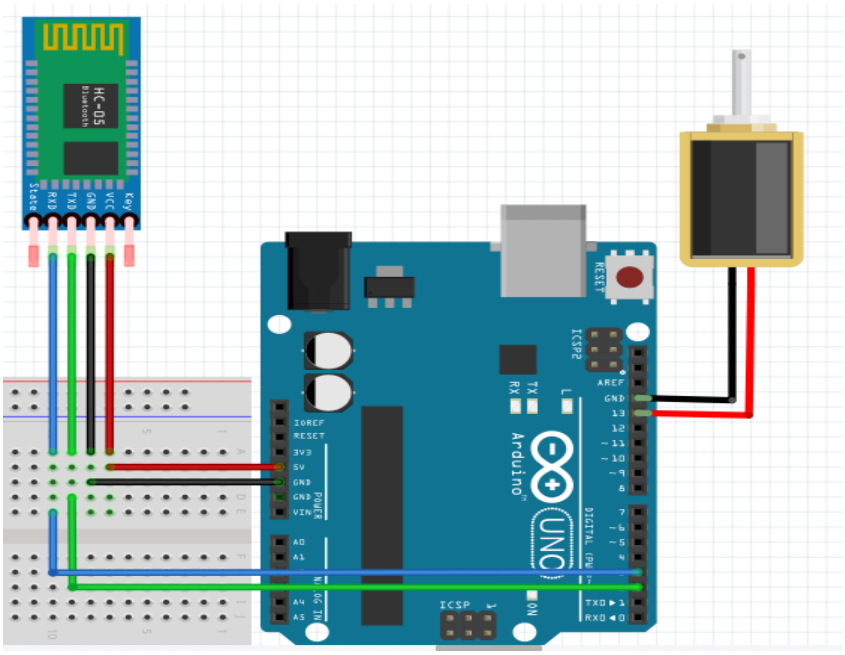

Figura 34 Diagrama conexión Arduino

3.4.2. CÓDIGO ARDUINO

Para el funcionamiento de la chapa eléctrica por medio de bluetooth, la cual se conecta a un Arduino uno, debemos tener dos estados de recepción, el primer estado bloqueara el dispositivo, y el segundo lo desbloqueara o liberara la chapa eléctrica, en la figura 35 se describe el código usado.

```
#include <SoftwareSerial.h>
SoftwareSerial ModBluetooth(2,3); // RX | TX
void setup()
{
    pinMode(13, OUTPUT);
    digitalWrite(13, LOW);
    ModBluetooth.begin(9600);
    Serial.begin(9600);
    ModBluetooth.println("MODULO CONECTADO");}
void loop()
{
    if (ModBluetooth.available()){
        char VarChar;
        VarChar = ModBluetooth.read();
        if(VarChar == '1'){
        digitalWrite(13, HIGH);
        delay(100);
        ModBluetooth.print("Chapa desbloqueada");
        Serial.print("Chapa desbloqueada");
        ModBluetooth.print("#");
        }
        if(VarChar == '0')
        {
        digitalWrite(13, LOW);
        delay(100);
        ModBluetooth.print("Chapa bloqueada#");
        Serial.print("Chapa bloqueada#");
        } } }
```

Figura 35 Código Arduino

3.4.3. PRUEBAS FUNCIONALES

En la figura 36, se muestra la conexión física del Arduino uno, el bluetooth y la chapa eléctrica.

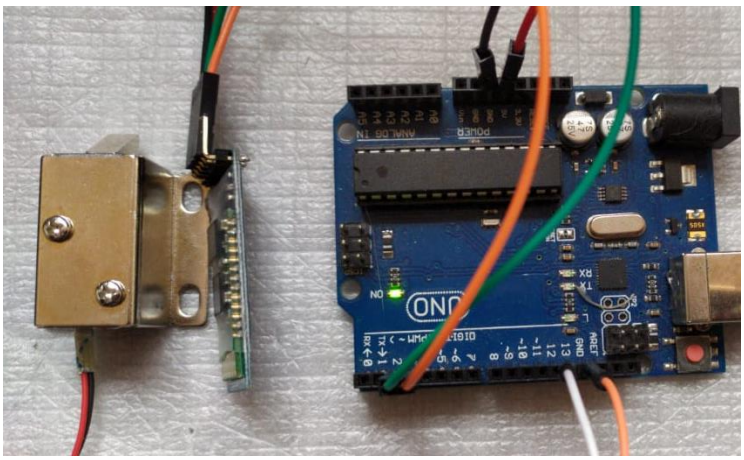

Figura 36 Conexión Arduino

En la figura 37 se puede notar el uso de la chapa eléctrica, este será el estado inicial de la chapa, la cual se mantiene bloqueada.

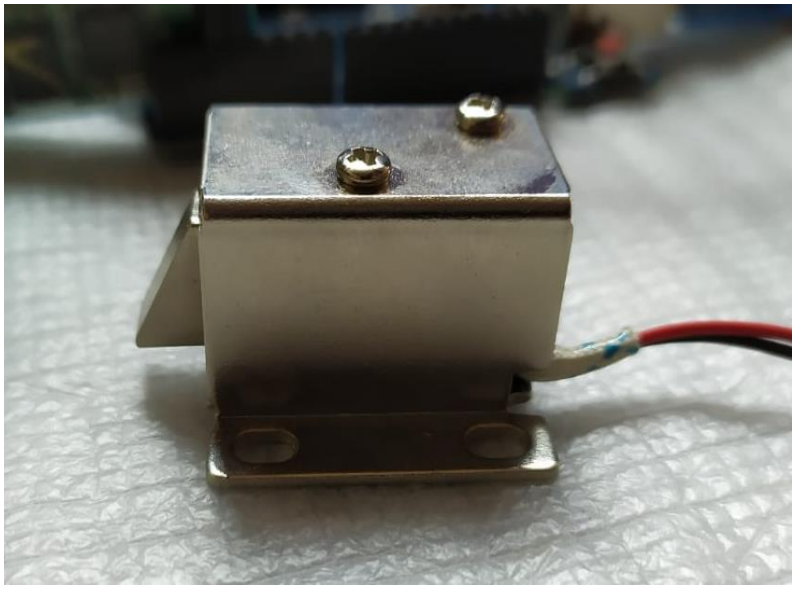

Figura 37 Chapa bloqueada

En la figura 38 se observa el funcionamiento de la chapa eléctrica, cuando el aplicativo envía una señal para el desbloqueo del dispositivo.

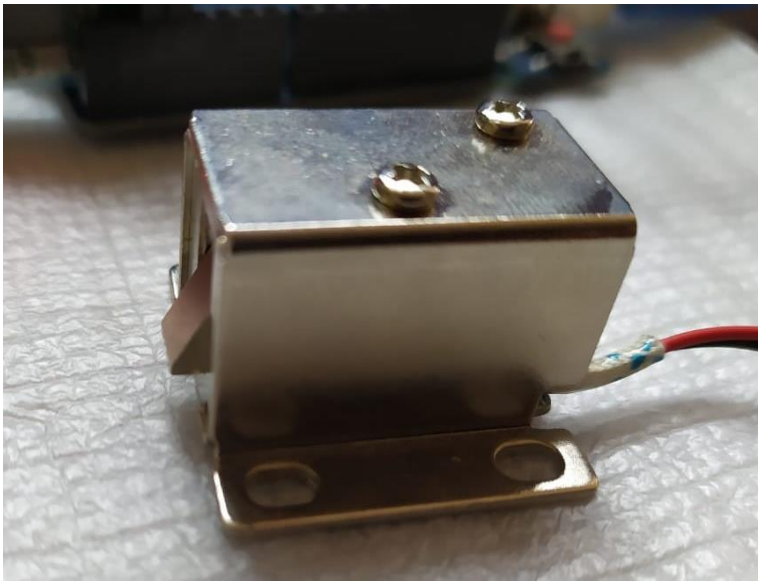

Figura 38 Chapa desbloqueada

3.5. DISCUSIÓN

La aplicación necesita los servicios de Microsoft Azure face API, y una base de datos SQL server para el registro e inicio de sesión en el aplicativo, de igual manera se necesita el uso de la cámara frontal y posterior, el desarrollo del aplicativo de reconocimiento facial permite brindar seguridad a dispositivos que se encuentren en el hogar, permitiendo tener un control de las personas que darán uso al dispositivo y restringirlo.

El aplicativo de reconocimiento facial, permite realizar varias acciones dentro del grupo de seguridad, tal es así como ingresar nuevos rostros y usuarios con los que podemos ingresar a los dispositivos de seguridad y grupos de seguridad, bloquear y desbloquear dispositivos que se tenga vinculados en el aplicativo, compartir e ingresar a grupos de seguridad, en el que esté registrado nuestro rostro, logrando dar uso a los dispositivos de seguridad del grupo seleccionado, y por último permite eliminar usuarios del grupo de seguridad, con esto negamos el ingreso a los dispositivos de seguridad y también a los grupos de seguridad.

El aplicativo permite tener un control de seguridad de usuarios para dispositivos que tengan como uso la conexión vía bluetooth logrando un método de seguridad menos invasivo.

4.CONCLUSIONES Y RECOMENDACIONES

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

En el desarrollo del trabajo de titulación se encontraron las siguientes conclusiones:

- El análisis de las tecnologías de reconocimiento facial permitió elegir la herramienta más adecuada, logrando tener un servicio de reconocimiento facial que se acople a los servicios en la nube que ofrece Microsoft.
- La metodología ágil Scrum permitió tener un control de todos los procesos que se llevaron a cabo, logrando un resultado óptimo y un desarrollo rápido para el aplicativo logrando cumplir con los objetivos de cada sprint.
- Es de vital importancia, la documentación del uso del reconocimiento facial. Se debe tener en cuenta una secuencia de pasos a realizar para el registro de un rostro y la identificación.
- El uso de servicios en la nube permite un ahorro en costos de recursos de hardware propio y permite una alta disponibilidad de los servicios prestados, como es en el caso el servicio de reconocimiento facial y la base de datos.
- Para usar los recursos de hardware correspondientes a la conexión como Bluetooth, se deben otorgar varios permisos en el aplicativo.

4.2. RECOMENDACIONES

Entre las recomendaciones se detalla lo siguiente:

- Integrar más herramientas para la conexión de dispositivos que tengamos en nuestro hogar, tales como la conexión wi-fi, para tener así una aplicación más global.
- Integrar más funcionalidades al módulo de modificación de grupos de seguridad para lograr una mejor personalización del usuario.
- Optimizar el método de reconocimiento facial, logrando un tiempo de respuesta más eficaz.
- Implementar métodos de seguridad alternativos, para trabajar en modo offline.

BIBLIOGRAFÍA

BIBLIOGRAFÍA

Arduino Uno. (2020, julio 15). Recuperado 15 de julio de 2020, de https://store.arduino.cc/usa/arduino-uno-rev3

CÁCERES, E. (2018). *Aplicación Móvil De Reconocimiento Facial En Personas Con Antecedentes De Abuso Sexual En La Provincia De Andahuaylas, Apurímac - 2018.*

Cadavid, A. N., Martínez, J. D. F., & Vélez, J. M. (2013). Prospectiva (Universidad Autonoma del Caribe - Colombia). *Prospectiva*, *11*(2). Recuperado de http://www.redalyc.org/comocitar.oa?id=496250736004

Calvachi, L. V. B. (2018). *ESTUDIO DE LA HERRAMIENTA "ANDROID STUDIO" CON APLICATIVO DE GESTIÓN DE PROVEEDORES, CLIENTES Y GESTIÓN DE PROFORMAS PARA EL TALLER MECÁNICO "EL GOLPE MÁGICO".*

Collier, M., & Robin, S. (2015). Microsoft Azure Essentials - Fundamentals of Azure. En *Microsoft Press*. Recuperado de https://books.google.es/books?hl=es&lr=&id=EfFxBgAAQBAJ&oi=fnd&pg=PA3&dq=microsoft+azure+&ots=1uSOZQW_07&sig=fwc2xxQaaT5D7NEVmgPKjMltFcI#v=onepage&q=microsoft azure&f=false

Costa, D. (2018). *Reconocimiento facial no invasivo en dispositivos móviles*.

elDiario.es. (2020, julio 15). Reconocimiento facial. Recuperado 15 de julio de 2020, de Reconocimiento facial website: https://www.eldiario.es/hojaderouter/tecnologia/software/reconocimiento-facial-funcionamiento-algoritmos_1_2537923.html

Eslava, J. (2013). *Reconocimiento facial en tiempo real*.

Etchart, G., Luna, L., Leal, C., Benedetto, M., & Alvez, C. (s. f.). *del uso de estándares en entes estatales Contexto Introducción Líneas de Investigación y Desarrollo*. (345).

Fernández, A., Belmonte, R., & Ortega, J. (2015). *Seguridad Biométrica*.

Groussard, T. *JAVA 7: Los fundamentos del lenguaje Java* . Recuperado de https://books.google.es/books?hl=es&lr=&id=JaPTzKZxbN4C&oi=fnd&pg=PA9&dq=lenguaje+Java&ots=pV8GobDj2i&sig=h4EWSGWQ5V79KvesL-W9IMb04S4#v=onepage&q=lenguaje Java&f=false

Hohensee, B. (s. f.). *Introducción A Android Studio*. Recuperado de https://books.google.es/books?id=4dkuBQAAQBAJ&printsec=frontcover&hl=es#v=onepage&q&f=false

Korth, H. F., & Sudarshan, S. (2002). *FUNDAMENTOS DE BASES DE DATOS* (4.ª ed.; C. F. Madrid, Ed.).

Lledó Sánchez, E. (2012). *Diseño sistema de control domotico usando plataforma arduino*.

Malave Polanco, K., & Beauperthuy Taibo, J. (2011). Android, el sistema operativo de google para dispositivos móviles. *NEGOTIUM*, (1), 18.

Microsoft. (2019). *Transparency Note Azure Cognitive Services: Face API*.

Microsoft. (2020). Conceptos de atributos y detección de caras - Azure Cognitive Services. Recuperado 11 de agosto de 2020, de https://docs.microsoft.com/es-es/azure/cognitive-services/face/concepts/face-detection

Microsoft Azure. (s. f.). Recuperado 7 de agosto de 2020, de https://azure.microsoft.com/es-es/overview/what-is-machine-learning-platform/#process

Rikelmer, M. M. (2012). *Microsoft Windows Azure Como Plataforma Para Prestación de servicios, soluciones y computación en la nube*. UNIVERSIDAD TECNOLOGICA ISRAEL.

Robledo, D. (s. f.). *Desarrollo de aplicaciones para Android II* (H. Alvarez, Ed.). Recuperado de https://books.google.es/books?hl=es&lr=&id=lwLXAwAAQBAJ&oi=fnd&pg=PT10&dq=sistema+operativo+android&ots=qE5Nj9CQLW&sig=BdJaS7I3zNTJflybCZRYc6xQUhA#v=onepage&q=sistema operativo android&f=false

SANABRIA, E. C. (1998). *Introducción a Java*.

Server, M. S. Q. L. (s. f.). *Microsoft SQL Server*. 1-6.

Soler, M. (2014). *Informe de OpenCV y Tratamiento de Imágenes*. 10.

Trigas Gallego, M. (s. f.). *Metodología Scrum .* (p. 56). p. 56.

ANEXOS

ANEXOS

ANEXO 1 INICIO SESIÓN

```
public class MainActivity extends AppCompatActivity {
    EditText edtNombre,edtcorreo,edtcontrasenia;
    Metodos metodos;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        edtNombre=(EditText)findViewById(R.id.edtnombre);
        edtcorreo=(EditText)findViewById(R.id.edtcorreo);
        edtcontrasenia=(EditText)findViewById(R.id.edtcontrasenia);
        if (Lectura("validacion").equals("aprobado")){
            Intent = new Intent(MainActivity.this, Menu.class);
            startActivity(intent);
            finish();
        }
    }
    public void crearUsuario(View view){
        Intent intent = new Intent(MainActivity.this, CrearUsuario.class);
        startActivity(intent);
        finish();
    }
    public void validarusuario(View view){
        try {
            Statement stm=conexionDB().createStatement();
```

```
        ResultSet rs= stm.executeQuery("SELECT * from usuarios WHERE codigo=(select codigo from usuarios where correo ='"+edtcorreo.getText().toString()+"') and PWDCOMPARE('"+edtcontrasenia.getText().toString()+"',contrasenia) = 1");

        if (rs.next()){

            if (rs.getString(3).equals(edtcorreo.getText().toString())){

                grabar("validacion","aprobado");

                grabar("credenciales",rs.getString(3));

                grabar("usuario",rs.getString(2));

                grabar("grupo","");

                Intent intent = new Intent(MainActivity.this, Menu.class);

                startActivity(intent);

                finish();

            }else{

                Toast.makeText(getApplicationContext(),"Usuario o contraseña incorrecta",Toast.LENGTH_SHORT).show();

            }

        }else{

            Toast.makeText(getApplicationContext(),"Usuario o contraseña incorrecta",Toast.LENGTH_SHORT).show();

        }

      }catch (SQLException e) {

        e.printStackTrace();

        Toast.makeText(getApplicationContext(),"Error Consulta:",Toast.LENGTH_SHORT).show();

      }

  }

  public String Lectura(String nombre){

      String[] archivos = fileList();
```

```
        String todo="";
        if (existe(archivos, nombre+".txt"))
            try {
                InputStreamReader archivo = new
InputStreamReader(openFileInput(nombre+".txt"));
                BufferedReader br = new BufferedReader(archivo);
                String linea = br.readLine();
                while (linea != null) {
                    todo = todo + linea;
                    linea = br.readLine();
                }
                br.close();
                archivo.close();
            } catch (IOException e) {
            }
        return todo;
    }
    private boolean existe(String[] archivos, String archbusca) {
        for (int f = 0; f < archivos.length; f++)
            if (archbusca.equals(archivos[f]))
                return true;
        return false;
    }
    public void grabar(String nombre,String Credenciales) {
        try {
            OutputStreamWriter archivo = new
OutputStreamWriter(openFileOutput(nombre+".txt",
Activity.MODE_PRIVATE));
```

```
            archivo.write(Credenciales);
            archivo.flush();
            archivo.close();
        } catch (IOException e) {
        }
    }
    public Connection conexionDB(){
        Connection conexion=null;
        try {
            StrictMode.ThreadPolicy policy= new
StrictMode.ThreadPolicy.Builder().permitAll().build();
            StrictMode.setThreadPolicy(policy);
            Class.forName("net.sourceforge.jtds.jdbc.Driver");
            conexion=
DriverManager.getConnection("jdbc:jtds:sqlserver://database.windows.net:14
33;DatabaseName=reconocimientof;user=admin@re;password=mast65465$
$2020;encrypt=true;trustServerCertificate=false;hostNameInCertificate=*.dat
abase.windows.net;loginTimeout=30;");
        } catch (ClassNotFoundException | SQLException e) {
            e.printStackTrace();
Toast.makeText(getApplicationContext(),"Error1",Toast.LENGTH_SHORT).s
how();
        }
        return conexion;
    }
}
```

ANEXO 2 CREACIÓN USUARIO

```
public class CrearUsuario extends AppCompatActivity {
    EditText contrasenia,contrasenia2,correo,nombre;
    Button Creacuenta;
    String Grupo;
    Metodos metodos= new Metodos();
    Azure azure= new Azure();
    private FaceServiceClient faceServiceClient = azure.conexionAzure();
    @Override
    public void onBackPressed() {
        Intent i = new Intent(CrearUsuario.this, SplashScreen.class);
        startActivity(i);
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_crear_usuario);
        contrasenia= (EditText)findViewById(R.id.edtcontrasenia);
        contrasenia2=(EditText)findViewById(R.id.edtcontrasenia2);
        correo = (EditText)findViewById(R.id.edtcorreo);
        nombre =(EditText)findViewById(R.id.edtnombre);
        Creacuenta=(Button)findViewById(R.id.btnCrearcuenta);
        Creacuenta.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                agregarUsuario();
            }
```

```
        });
    }
    public static boolean valemail(String email) {
        boolean isValid = false;
        String expression = "^[\\w\\.-]+@([\\w\\-]+\\.)+[A-Z]{2,4}$";
        CharSequence inputStr = email;
        Pattern pattern = Pattern.compile(expression,
Pattern.CASE_INSENSITIVE);
        Matcher matcher = pattern.matcher(inputStr);
        if (matcher.matches()) {
            isValid = true;
        }
        return isValid;
    }
    public boolean duplicadoCorreo(String correo){
        boolean test=false;
    try {
        Statement stm=metodos.conexionDB().createStatement();
Toast.makeText(getApplicationContext(),correo,Toast.LENGTH_LONG).sho
w();
        ResultSet rs= stm.executeQuery("select * from usuarios where correo
='"+correo+"'");
        if (rs.next()){
        if (rs.getString(3).equals(correo)){
            test= false;
          }else{
            test= true;
          }
```

```
        }else{
            System.out.println("no entro al if");
            test=true;
        }
    }catch (SQLException e) {
        e.printStackTrace();
    }
    return test;
    }
    public boolean contraVacia(String contra1,String contra2){
        boolean respuesta=false;
        if (contra1.equals("" )|| contra2.equals("")){
            respuesta=true;
        }else{
            respuesta=false;
        }
        return respuesta;
    }
    public void agregarUsuario(){
        boolean validacion=valemail(correo.getText().toString());
        boolean duplicado= duplicadoCorreo(correo.getText().toString());
        boolean vacio=
contraVacia(contrasenia.getText().toString(),contrasenia2.getText().toString()
);
        if
(contrasenia.getText().toString().equals(contrasenia2.getText().toString()) &&
validacion==true && duplicado==true && vacio==false){
            new AddPersonGroupTask().execute();
        }else{
```

```
        if (duplicado==false){

            Toast.makeText(getApplicationContext(),"esta cuenta ya existe",Toast.LENGTH_SHORT).show();

        }else {

            Toast.makeText(getApplicationContext(), "Las contraseñas no coinciden o Correo mal ingresado ", Toast.LENGTH_SHORT).show();

        }   }   }

    class AddPersonGroupTask extends AsyncTask<String,String,String> {

        AlertDialog alertDialog = new SpotsDialog.Builder().setContext(CrearUsuario.this).setCancelable(false).build();

        @Override

        protected String doInBackground(String... params) {

            try {

                UUID uuid = UUID.randomUUID();

                String clave = uuid.toString();

                Grupo=clave;

                publishProgress("Creando Usuario...");

                metodos.conexionDB();

                faceServiceClient.createPersonGroup(clave, Grupo, Grupo);

                PreparedStatement pst=metodos.conexionDB().prepareStatement("Insert into usuarios values(?,?,PWDENCRYPT(?),?)");

                pst.setString(1,nombre.getText().toString());

                pst.setString(2,correo.getText().toString());

                pst.setString(3,contrasenia.getText().toString());

                pst.setString(4,clave);

                pst.executeUpdate();

                return "Exito";
```

```
            } catch (Exception e) {
                publishProgress(e.getMessage());
                e.printStackTrace();
                return null;
            }       }
        @Override
        protected void onPreExecute() {
            alertDialog.show();
        }
        @Override
        protected void onProgressUpdate(String... values) {
            alertDialog.setMessage(values[0]);
        }
        @Override
        protected void onPostExecute(String result) {
            alertDialog.dismiss();
            if (result != null) {
                Toast.makeText(getApplicationContext(), "Usuario Creado.", Toast.LENGTH_LONG).show();
                Intent intent= new Intent(CrearUsuario.this, MainActivity.class);
                startActivity(intent);
            }else {
                Toast.makeText(getApplicationContext(), "Error al Crear Usuario, Intente Nuevamente", Toast.LENGTH_LONG).show();
                Intent intent= new Intent(CrearUsuario.this, MainActivity.class);
                startActivity(intent);
            }       }   } }
```

ANEXO 3 AGREGAR ROSTRO DE USUARIO

```
public class AgregarUsuario extends AppCompatActivity {
    Bitmap mBitmap;
    int opcion=0,flag=0;
    String personId,nombre,Name, Group;
    ImageView img1,img2,img3;
    private static String API_KEY="930f765465465470047897987987982321 3df";
    private static String API_LINK="https://cognitiveservices.azure.com/face/v1.0";
    private static FaceServiceClient faceServiceClient = new FaceServiceRestClient(API_LINK,API_KEY);
    FaceGridViewAdapter mFaceGridViewAdapter;
    @Override
    public void onBackPressed() {
        Intent i = new Intent(AgregarUsuario.this, Menu.class);
        startActivity(i);
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_agregar_usuario);
        img1=(ImageView)findViewById(R.id.img1);
        img2=(ImageView)findViewById(R.id.img2);
        img3=(ImageView)findViewById(R.id.img);
        Bundle parametros = this.getIntent().getExtras();
        if(parametros !=null){
            Group= parametros.getString("Grupo");
```

```
            System.out.println("el grupo que se trajo en agregar usuario fue linea 83"+Group);
        }else{
            System.out.println("Error al agregar usuario linea 85");
            Error();
        }
    }
    public void grabar(String nombre,String Credenciales) {
        try {
            OutputStreamWriter archivo = new OutputStreamWriter(openFileOutput(nombre+".txt", Activity.MODE_PRIVATE));
            archivo.write(Credenciales);
            archivo.flush();
            archivo.close();
        } catch (IOException e) {
            e.printStackTrace();
        }
    }
    public void enviar(View view){
     EditText nombre=(EditText)findViewById(R.id.edtnombre);
     Name= nombre.getText().toString();
    if (Name.isEmpty() || img1.getDrawable()==null || img2.getDrawable()==null || img3.getDrawable()==null){
         Toast.makeText(getApplicationContext(), "Nombre del usuario o imagen de identificacion vacia",Toast.LENGTH_SHORT).show();
     }else{
         new AddPersonTask().execute();
        }
```

```
    }
    public void reconocerCara(){
        Bitmap cara1= recuperar("cara1");
        InputStream face1= Detectar(cara1);
        mBitmap=cara1;
        new detectTask().execute(face1);
        Bitmap cara2= recuperar("cara2");
        mBitmap=cara2;
        InputStream face2= Detectar(cara2);
        new detectTask().execute(face2);
        Bitmap cara3= recuperar("cara3");
        mBitmap=cara3;
        InputStream face3= Detectar(cara3);
        new detectTask().execute(face3);
    }
    class AddPersonTask extends AsyncTask<String, String, String> {
        AlertDialog alertDialog= new
SpotsDialog.Builder().setContext(AgregarUsuario.this).setCancelable(false).b
uild();
        @Override
        protected String doInBackground(String... params) {
            try{
                CreatePersonResult
result=faceServiceClient.createPerson(Group,Name,Name);
                return result.personId.toString();
            } catch (Exception e) {
                publishProgress(e.getMessage());
                e.printStackTrace();
```

```
                return null;
            }
        }
        @Override
        protected void onPostExecute(String result) {
            alertDialog.dismiss();
            if (result != null) {
                personId = result;
                System.out.println("se creo persona agregar usuario linea 140 con id: "+personId);
                reconocerCara();
            }else{
                Error();
            }
        }
        @Override
        protected void onPreExecute() {
            alertDialog.show();
        }
        @Override
        protected void onProgressUpdate(String... values) {
            alertDialog.setMessage(values[0]);
        }
    }
    private void openCamera(String nombre, int opcion) {
        this.nombre=nombre;
        this.opcion=opcion;
```

```
        Intent cameraIntent= new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        cameraIntent.putExtra("android.intent.extras.CAMERA_FACING", 1);
        startActivityForResult(cameraIntent,1);
    }
    @Override
    protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
        super.onActivityResult(requestCode, resultCode, data);
        if(requestCode== 1 && resultCode== RESULT_OK){
            Bundle extras =data.getExtras();
            mBitmap= (Bitmap)extras.get("data");
            if (opcion==1){img1.setImageBitmap(mBitmap);}
            if (opcion==2){img2.setImageBitmap(mBitmap);}
            if(opcion==3){img3.setImageBitmap(mBitmap);}
            try {
                ByteArrayOutputStream stream = new ByteArrayOutputStream();
                mBitmap.compress(Bitmap.CompressFormat.JPEG, 100, stream);
                byte[] byteArray = stream.toByteArray();
                FileOutputStream outputStream = getApplicationContext().openFileOutput(nombre+".jpg", Context.MODE_PRIVATE);
                outputStream.write(byteArray);
                outputStream.close();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
    }
```

```
    public void cara1(View view){
       openCamera("cara1",1);
    }
    public void cara2(View view){
        openCamera("cara2",2);
    }
    public void cara3(View view){
        openCamera("cara3",3);
    }
    public Bitmap recuperar(String nombre){
        Bitmap bitmap = null;
        try{
            FileInputStream fileInputStream =new
FileInputStream(getApplicationContext().getFilesDir().getPath()+
"/"+nombre+".jpg");
            bitmap = BitmapFactory.decodeStream(fileInputStream);
        }catch (IOException io){
            io.printStackTrace();
        }
        return bitmap;
    }
    public InputStream Detectar(Bitmap bitmap){
        ByteArrayOutputStream stream = new ByteArrayOutputStream();
        bitmap.compress(Bitmap.CompressFormat.JPEG, 100, stream);
        InputStream imageInputStream = new
ByteArrayInputStream(stream.toByteArray());
        return imageInputStream;
    }
```

```
    class detectTask extends AsyncTask<InputStream, String, Face[]> {

        AlertDialog alertDialog= new SpotsDialog.Builder().setContext(AgregarUsuario.this).setCancelable(false).build();

        @Override

        protected Face[] doInBackground(InputStream... inputStreams) {

            try{

                publishProgress("Detectando");

                Face[] result=faceServiceClient.detect(inputStreams[0],true,false,null);

                return result;

            }catch(IOException | edmt.dev.edmtdevcognitiveface.Rest.ClientException e){

                e.printStackTrace();

                return null;

            }

        }

        protected void onPostExecute(Face[] faces){

            alertDialog.dismiss();

            if(faces == null){

                System.out.println("Cara no detectada linea 241");

                Error();

                finish();

            }else{

                System.out.println("cara detectada linea 242");

                setUiAfterDetection(faces, true);

            }

        }

        @Override
```

```
    protected void onPreExecute(){
        alertDialog.show();
    }
    @Override
    protected void onProgressUpdate(String... values){
        alertDialog.setMessage(values[0]);
    }
}
private void setUiAfterDetection(Face[] result, boolean succeed) {
    mFaceGridViewAdapter = new FaceGridViewAdapter(result);
    if (mFaceGridViewAdapter != null) {
        List<Integer> faceIndices = new ArrayList<>();
        for (int i = 0; i < mFaceGridViewAdapter.faceRectList.size(); ++i) {
          if (mFaceGridViewAdapter.faceChecked.get(i)) {
            faceIndices.add(i);
        }
        if (faceIndices.size() > 0) {
            new AddFaceTask(faceIndices).execute();
        } else {
            Error();
            System.out.println("cero en agregar usuario");
        }
    }else{
        Error();
    }
}
```

```
public Bitmap generateFaceThumbnail(Bitmap originalBitmap, FaceRectangle faceRectangle) throws IOException {

    FaceRectangle faceRect =calculateFaceRectangle(originalBitmap, faceRectangle, 1.3);

    return Bitmap.createBitmap(originalBitmap, faceRect.left, faceRect.top, faceRect.width, faceRect.height);

}

private FaceRectangle calculateFaceRectangle(Bitmap bitmap, FaceRectangle faceRectangle, double faceRectEnlargeRatio) {

    double sideLength = faceRectangle.width * faceRectEnlargeRatio;

    sideLength = Math.min(sideLength, bitmap.getWidth());

    sideLength = Math.min(sideLength, bitmap.getHeight());

    double left = faceRectangle.left- faceRectangle.width * (faceRectEnlargeRatio - 1.0) * 0.5;

    left = Math.max(left, 0.0);

    left = Math.min(left, bitmap.getWidth() - sideLength);

    double top = faceRectangle.top- faceRectangle.height * (faceRectEnlargeRatio - 1.0) * 0.5;

    top = Math.max(top, 0.0);

    top = Math.min(top, bitmap.getHeight() - sideLength);

   double shiftTop = faceRectEnlargeRatio - 1.0;

    shiftTop = Math.max(shiftTop, 0.0);

    shiftTop = Math.min(shiftTop, 1.0);

    top -= 0.15 * shiftTop * faceRectangle.height;

    top = Math.max(top, 0.0);

    FaceRectangle result = new FaceRectangle();

    result.left = (int)left;

    result.top = (int)top;

    result.width = (int)sideLength;
```

```
        result.height = (int)sideLength;
        FaceRectangle test=result;
        return result;
    }
    private class FaceGridViewAdapter extends BaseAdapter {
        List<UUID> faceIdList;
        List<FaceRectangle> faceRectList;
        List<Bitmap> faceThumbnails;
        List<Boolean> faceChecked;
        FaceGridViewAdapter(Face[] detectionResult) {
            faceIdList = new ArrayList<>();
            faceRectList = new ArrayList<>();
            faceThumbnails = new ArrayList<>();
            faceChecked = new ArrayList<>();
            if (detectionResult != null) {
                List<Face> faces = Arrays.asList(detectionResult);
                for (Face face : faces) {
                    try {
                        faceThumbnails.add(generateFaceThumbnail(mBitmap,
face.faceRectangle));
                        faceIdList.add(null);
                        faceRectList.add(face.faceRectangle);
                        faceChecked.add(false);
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                }
```

```
            }
        }
        @Override
        public int getCount() {
            return faceRectList.size();
        }
        @Override
        public Object getItem(int position) {
            return faceRectList.get(position);
        }
        @Override
        public long getItemId(int position) {
            return position;
        }
        @Override
        public View getView(final int position, View convertView, ViewGroup parent) {
            convertView.setId(position);
            faceChecked.set(position, true);
            return convertView;
        }
    }
    class AddFaceTask extends AsyncTask<Void, String, Boolean> {
        AlertDialog alertDialog= new SpotsDialog.Builder().setContext(AgregarUsuario.this).setCancelable(false).build();
        List<Integer> mFaceIndices;
        AddFaceTask(List<Integer> faceIndices) {
```

```
            mFaceIndices = faceIndices;
        }
        @Override
        protected Boolean doInBackground(Void... params) {
            try{
                publishProgress("Añadiendo Rostro...");
                ByteArrayOutputStream stream = new ByteArrayOutputStream();
                UUID personI = UUID.fromString(personId);
                mBitmap.compress(Bitmap.CompressFormat.JPEG, 100, stream);
                InputStream imageInputStream = new
ByteArrayInputStream(stream.toByteArray());
                for (Integer index: mFaceIndices) {
                    FaceRectangle faceRect =
mFaceGridViewAdapter.faceRectList.get(index);
                    AddPersistedFaceResult result
=faceServiceClient.addPersonFace(
                            Group,personI,imageInputStream,"User data",faceRect);
                    mFaceGridViewAdapter.faceIdList.set(index,
result.persistedFaceId);
                }return true;
            }
            catch (Exception e) {
                publishProgress(e.getMessage());
                e.printStackTrace();
                return false;
            }
        }
        @Override
        protected void onPreExecute() {alertDialog.show(); }
```

```
        @Override
        protected void onProgressUpdate(String... values) {alertDialog.setMessage(values[0]);}
        @Override
        protected void onPostExecute(Boolean result) {
            alertDialog.dismiss();
            if (result) {
                System.out.println("Se envio Imagen");
                 flag=flag+1;
                if (flag==2){
                    new Entrenar().execute();
                }
            }
        }
    }
    public class Entrenar extends AsyncTask<Void,Void,String>  {
        private Context httpContext;
        private String resultadoapi="";
        private String linkrequestAPI="";
        @Override
        protected void  onPreExecute(){
            super.onPreExecute();
        }
        protected void onPostExecute(String s){
            super.onPostExecute(s);
                resultadoapi=s;
                Toast.makeText(getApplicationContext(), "Persona agregada con Exito", Toast.LENGTH_LONG).show();
```

```
            eliminar();
            Intent intent= new Intent(AgregarUsuario.this, Menu.class);
            startActivity(intent);
        }
        @Override
        protected String doInBackground(Void... voids) {
            String result=null;
            HttpClient httpclient = HttpClients.createDefault();
            try
            {
                URIBuilder builder = new URIBuilder("https://cognitiveservices.azure.com/face/v1.0/persongroups/"+Group+"/train");
                URI uri = builder.build();
                HttpPost request = new HttpPost(uri);
                request.setHeader("Ocp-Apim-Subscription-Key", "930f65675d6546500470284dca6a654654");
                StringEntity reqEntity = new StringEntity("{body}");
                request.setEntity(reqEntity);
                HttpResponse response = httpclient.execute(request);
                HttpEntity entity = response.getEntity();
                if (entity != null)
                {
                    System.out.println(EntityUtils.toString(entity));
                }
            }
            catch (Exception e){
                System.out.println("error"+e.getMessage());
```

```
            }
            return result;
        }
    }
    public void eliminar() {
        for (int i = 0; i < 4; i++) {
            File file = new File(getApplicationContext().getFilesDir().getPath() + "/cara" + (i + 1) + ".jpg");
            if (file.exists()) {
                boolean eliminado = file.delete();
            }
        }
    }
    int contador=0;
    public void Error(){
        if (contador==0){
            Toast.makeText(getApplicationContext(), "Error al agregar usuario, por favor intente de nuevo", Toast.LENGTH_LONG).show();
            new AgregarUsuario.Eliminar().execute();
            contador++;
            Intent intent = new Intent(AgregarUsuario.this, Menu.class);
            startActivity(intent);
            finish();
        }
    }
    class Eliminar extends AsyncTask<UUID, String, edmt.dev.edmtdevcognitiveface.Contract.Person> {
        AlertDialog alertDialog= new SpotsDialog.Builder()
```

```
            .setContext(AgregarUsuario.this)
            .setCancelable(false)
            .build();
        @Override
        protected void onPreExecute(){
            alertDialog.show();
        }
        @Override
        protected void onProgressUpdate(String... values){
            alertDialog.setMessage(values[0]);
        }
        @Override
        protected edmt.dev.edmtdevcognitiveface.Contract.Person
doInBackground(UUID... uuids) {
            try {
                faceServiceClient.deletePerson(Group,
UUID.fromString(personId));
            } catch (ClientException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
            return null;
        }
        @Override
        protected void
onPostExecute(edmt.dev.edmtdevcognitiveface.Contract.Person person ){
            alertDialog.dismiss();        }    }}
```

ANEXO 4 MODIFICAR GRUPO DE SEGURIDAD

```
public class EliminarUsuario extends AppCompatActivity {
    String Grupo,Codigo,Nombre;
    TextView Usuario;
    @Override
    public void onBackPressed() {
        Intent i = new Intent(EliminarUsuario.this, Menu.class);
        startActivity(i);
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_elminar_usuario);
        Bundle parametros = this.getIntent().getExtras();
        if(parametros !=null){
            Grupo= parametros.getString("Grupo");
            Codigo= parametros.getString("Id");
            Nombre= parametros.getString("Nombre");
            Usuario= (TextView)findViewById(R.id.edtUsuario);
            Usuario.setText(Nombre);
        }else{
            Toast.makeText(getApplicationContext(),                    "error",
Toast.LENGTH_LONG).show();
        }
     }
    public void grabar(String nombre,String Credenciales) {
        try {
```

```
            OutputStreamWriter archivo = new
OutputStreamWriter(openFileOutput(nombre+".txt",
Activity.MODE_PRIVATE));
            archivo.write(Credenciales);
            archivo.flush();
            archivo.close();
        } catch (IOException e) {
        }
    }
    public void eliminar(View view){
        new EliminarUsuario.Eliminar().execute();
    }
    private FaceServiceClient faceServiceClient = new
FaceServiceRestClient(API_LINK,API_KEY);
    class Eliminar extends AsyncTask<UUID, String,
edmt.dev.edmtdevcognitiveface.Contract.Person> {
        AlertDialog alertDialog= new SpotsDialog.Builder()
                .setContext(EliminarUsuario.this)
                .setCancelable(false)
                .build();
        @Override
        protected void onPreExecute(){
            alertDialog.show();
        }
        @Override
        protected void onProgressUpdate(String... values){
            alertDialog.setMessage(values[0]);
        }
        @Override
```

```
        protected edmt.dev.edmtdevcognitiveface.Contract.Person
doInBackground(UUID... uuids) {
            try {
                faceServiceClient.deletePerson(Grupo, UUID.fromString(Codigo))
            } catch (ClientException e) {
                e.printStackTrace();
            } catch (IOException e) {
                e.printStackTrace();
            }
            return null;
        }
        @Override
        protected                                                    void
onPostExecute(edmt.dev.edmtdevcognitiveface.Contract.Person person ){
            alertDialog.dismiss();
            Toast.makeText(getBaseContext(),"Usuario
eliminado",Toast.LENGTH_LONG).show();
            Intent i = new Intent(EliminarUsuario.this,Menu.class);
            startActivity(i);
        }
    }
}
```

ANEXO 5 MIS DISPOSITIVOS

```
public class DispositivosBT extends AppCompatActivity {
    private static final String TAG= "DispositivosBT";
    ListView idlista;
    public static String EXTRA_DEVICE_ADDRESS="device_address";
    private BluetoothAdapter mBadapter;
    private ArrayAdapter<String> mPairedDevicesArrayAdapter;
    @Override
    public void onBackPressed() {
        Intent i = new Intent(DispositivosBT.this, Menu.class);
        startActivity(i);    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_dispsitivos_b_t);          }
    public void grabar(String nombre,String Credenciales) {
        try   {                                  OutputStreamWriter   archivo   =   new
OutputStreamWriter(openFileOutput(nombre+".txt",
Activity.MODE_PRIVATE));
            archivo.write(Credenciales);
            archivo.flush();
            archivo.close();
        } catch (IOException e) {    }    }
    @Override
    public void onResume(){
        super.onResume();
        VerificarEstadoBT();
```

```
mPairedDevicesArrayAdapter = new ArrayAdapter<String>(this,R.layout.nombre_dispositivos_bt);

idlista=(ListView)findViewById(R.id.idlista);

idlista.setAdapter(mPairedDevicesArrayAdapter);

idlista.setOnItemClickListener(mDeviceClickListener);

mBadapter=BluetoothAdapter.getDefaultAdapter();

Set<BluetoothDevice>pairedDevices=mBadapter.getBondedDevices();

if (pairedDevices.size()>0){

for (BluetoothDevice device: pairedDevices){ mPairedDevicesArrayAdapter.add(device.getName()+"\n"+device.getAddres s()); } } }

private AdapterView.OnItemClickListener mDeviceClickListener = new AdapterView.OnItemClickListener() {

public void onItemClick(AdapterView av, View v, int arg2, long arg3) {

String info = ((TextView) v).getText().toString();

String address = info.substring(info.length() - 17);

Intent i = new Intent(DispositivosBT.this, principalbluetooth.class);

i.putExtra(EXTRA_DEVICE_ADDRESS, address);

startActivity(i); } };

private void VerificarEstadoBT(){

mBadapter=BluetoothAdapter.getDefaultAdapter();

if (mBadapter==null){

Toast.makeText(getBaseContext(),"El dispositivo no soporta Bluetooth",Toast.LENGTH_LONG).show();}else{

if (mBadapter.isEnabled()){

Log.d(TAG,"... Bluetooth Activado"); }else{

Intent enableBtIntent= new Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);

startActivityForResult(enableBtIntent,1); } } }}
```

ANEXO 6 GRUPOS DE SEGURIDAD

```
public class escanear extends AppCompatActivity implements ZXingScannerView.ResultHandler {
    private ZXingScannerView mScannerView;
    String cadena_a,cadena_b,cadena_c;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_escanear);
        scan();
    }
    public void scan(){
        mScannerView = new ZXingScannerView(this);
        setContentView(mScannerView);
        mScannerView.setResultHandler(this);
        mScannerView.startCamera();
    }   @Override
    public void handleResult(Result result) {
        Log.v("HandleResult",result.getText());
        cadena_c=result.getText();
        cadena_a = cadena_c.substring(0,36);
        cadena_b = cadena_c.substring(36);
        if (cadena_c.length()==41 && cadena_b.equals("tesis"))
        {
            mScannerView.removeAllViews();
            mScannerView.stopCamera();
            Intent intent  = new Intent(escanear.this, AgregarMisGrupos.class);
```

```
            Bundle extras=new Bundle();
            extras.putString("Grupo",cadena_a);
            intent.putExtras(extras);
            startActivity(intent);
            }else{
                Toast.makeText(getApplicationContext(),"Error al escanear codigo", Toast.LENGTH_LONG).show();
                Intent i = new Intent(escanear.this, Menu.class);
                startActivity(i);
        }
    }
}
public class compartir extends AppCompatActivity {
    String TAG="GenerateQRCode";
    Bitmap bitmap;
    QRGEncoder qrgEncoder;
    ImageView qrimg;
    String Grupo;
    @Override
    public void onBackPressed() {
        Intent i = new Intent(compartir.this, Menu.class);
        startActivity(i);
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_compartir);
```

```
        qrimg=(ImageView) findViewById(R.id.qrcode);
        Bundle parametros = this.getIntent().getExtras();
        if(parametros !=null){
            Grupo= parametros.getString("Grupo");
            System.out.println("El grupo que se trajo fue "+Grupo);
        }else{
            System.out.println("No se trajeron los parametros de grupo e ID necesarios en identificador");
        }
        crear();
    }
    public void grabar(String nombre,String Credenciales) {
        try {
            OutputStreamWriter archivo = new OutputStreamWriter(openFileOutput(nombre+".txt", Activity.MODE_PRIVATE));
            archivo.write(Credenciales);
            archivo.flush();
            archivo.close();
        } catch (IOException e) {
        }
    }
    public void crear(){
        String valor=Grupo+"tesis".trim();
        if (valor.length()>0){
            WindowManager manager= (WindowManager)getSystemService(WINDOW_SERVICE);
            Display display= manager.getDefaultDisplay();
            Point point= new Point();
```

```
            display.getSize(point);
            int width=point.x;
            int height= point.y;
            int smallerdimension=width<height ? width:height;
            smallerdimension= smallerdimension*3/4;
            qrgEncoder= new QRGEncoder(valor, null,
QRGContents.Type.TEXT,smallerdimension);
            try{
                bitmap=qrgEncoder.encodeAsBitmap();
                qrimg.setImageBitmap(bitmap);
            }catch(WriterException e)   {
                Log.v(TAG,e.toString());
            }
        }else{
            System.out.println("request failed");
        }
    }
}
```

Quito D.M., 13 de agosto de 2020

Ingeniero

Juan Carlos Rivera, MSc.

DECANO DE LA FACULTAD DE CIENCIAS DE LA INGENIERÍA E INDUSTRIAS

UNIVERSIDAD UTE

Presente.-

De mi consideración,

Por medio del presente, me dirijo a usted para informarle que el trabajo de titulación

"**DESARROLLO DE UNA APLICACIÓN QUE USANDO RECONOCIMIENTO FACIAL PERMITE BLOQUEAR/DESBLOQUEAR DISPOSITIVOS DE HOGARES INTELIGENTES"**, desarrollado por el estudiante **NICOLÁS PATRICIO JIMÉNEZ DÁVILA**, previa a la obtención del título de **INGENIERO EN INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN**, ha concluido bajo mi dirección y tutoría, el trabajo de titulación constituye un documento realizado según los formatos establecidos por la Facultad de Ciencias de la Ingeniería e Industrias, consta de: una sección con la introducción, los capítulos correspondientes a metodología, discusión, conclusiones y recomendaciones, bibliografía y anexos que contienen 17 tablas, 38 figuras, 6 Anexos y 24 referencias bibliográficas.

En este trabajo se ha realizado el desarrollo de una aplicación para dispositivos móviles con sistema operativo Android, que por medio de reconocimiento facial pueda abrir o cerrar una cerradura de manera automática.

El texto fue verificado por el sistema URKUND y contiene los siguientes indicadores: similitud del texto es de 6%, para lo cual adjunto el enlace respectivo:

https://secure.urkund.com/old/view/74532452-979135-697344#DcQxDoAgDAXQu3T+MS3QWriKcTBEDYMsjMa7yxveS8+gsjEEMpsHBENwREXMSFAYVjg87 aDR7t6uVo9eTyq8sKoxS8yS1ZIz6/cD

Por la atención que se digne dar a la presente, anticipo mi consideración y estima.

Atentamente.

Ing. Rodrigo Proaño Escalante
Director de trabajo de titulación

Ingeniería Informática y Ciencias de la Computación

Facultad de Ciencias de la Ingeniería e Industrias