

UNIVERSIDAD NACIONAL JOSÉ MARÍA ARGUEDAS

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



Presentado por:

ERVIN LEWIS CÁCERES MARIÑO

**APLICACIÓN MÓVIL DE RECONOCIMIENTO FACIAL EN
PERSONAS CON ANTECEDENTES DE ABUSO SEXUAL EN LA
PROVINCIA DE ANDAHUAYLAS, APURÍMAC - 2018.**

Asesor:

Dr. JULIO CÉSAR HUANCA MARÍN

**TESIS PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO
DE SISTEMAS**

ANDAHUAYLAS – APURÍMAC – PERÚ

2018

APROBACIÓN DEL ASESOR



APROBACIÓN DEL ASESOR

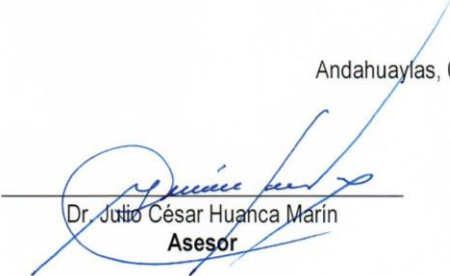
Quién suscribe:


Dr. Julio César Huanca Marín, identificado con CIP N° 095492, por la presente:

CERTIFICA,

Que, el Bachiller en Ingeniería de Sistemas, ERVIN LEWIS CÁCERES MARIÑO ha culminado satisfactoriamente el informe final de tesis intitulado: "APLICACIÓN MÓVIL DE RECONOCIMIENTO FACIAL EN PERSONAS CON ANTECEDENTES DE ABUSO SEXUAL EN LA PROVINCIA DE ANDAHUAYLAS, APURÍMAC - 2018." para optar el Título Profesional de Ingeniero de Sistemas.

Andahuaylas, 07 de Diciembre de 2018.


Dr. Julio César Huanca Marín
Asesor


Bach. Ervin Lewis Cáceres Mariño
Tesista

ACTA DE SUSTENTACIÓN DE TESIS



Universidad Nacional José María Arguedas

Identidad y Excelencia para el Trabajo Productivo y el Desarrollo



FACULTAD DE INGENIERÍA

ACTA DE SUSTENTACIÓN DE TESIS

En la Av. José María Arguedas del Local Académico SL01 (Ccoyahuacho) en el auditorio de la Escuela Profesional de Ingeniería de Sistemas de la Universidad Nacional José María Arguedas ubicado en el distrito de San Jerónimo de la Provincia de Andahuaylas, siendo las 08:00 horas del día 29 de noviembre del año 2018, se reunieron los docentes: Dr. Yalmar Temistocles Ponce Atencio, MSc. Carlos Yínnel Castro Buleje, MSc. Richard Carrión Abollaneda, en condición de integrantes del Jurado Evaluador del Informe Final de Tesis intitulado: "APLICACIÓN MÓVIL DE RECONOMIENTO FACIAL EN PERSONAS CON ANTECEDENTES DE ABUSO SEXUAL EN LA PROVINCIA DE ANDAHUAYLAS, APURÍMAC - 2018", cuyo autor es el Bachiller en Ingeniería de Sistemas **ERVIN LEWIS CÁCERES MARIÑO**, el asesor Dr. Julio César Huanca Marín., con el propósito de proceder a la sustentación y defensa de dicha tesis.

Luego de la sustentación y defensa de la tesis, el Jurado Evaluador **ACORDÓ: APROBAR** por **UNANIMIDAD** al Bachiller en Ingeniería de Sistemas **ERVIN LEWIS CÁCERES MARIÑO**, obteniendo la siguiente calificación y mención:

Nota escala vigesimal		Mención
Números	Letras	
14	CATORCE	BUENO

En señal de conformidad, se procedió a la firma de la presente acta en 03 ejemplares.

.....
Dr. Yalmar Temistocles Ponce Atencio
Presidente del Jurado Evaluador

.....
MSc. Carlos Yínnel Castro Buleje
Primer Miembro del Jurado Evaluador

.....
MSc. Richard Carrión Abollaneda
Segundo Miembro del Jurado Evaluador

APROBACIÓN DEL JURADO DICTAMINADOR




APROBACIÓN DEL JURADO DICTAMINADOR

LA TESIS: APLICACIÓN MÓVIL DE RECONOCIMIENTO FACIAL EN PERSONAS CON ANTECEDENTES DE ABUSO SEXUAL EN LA PROVINCIA DE ANDAHUAYLAS, APURÍMAC - 2018.; para optar el Título Profesional de Ingeniero de Sistemas, ha sido evaluada por el Jurado Dictaminador conformado por:


PRESIDENTE: Dr. Yalmar Temistocles Ponce Atencio
PRIMER MIEMBRO: MSc. Carlos Yinmel Castro Buleje
SEGUNDO MIEMBRO: MSc. Richard Carrión Abollaneda

Habiendo sido aprobado por UNANIMIDAD/MAYORÍA, en la ciudad de Andahuaylas el día 29 del mes de noviembre de 2018


Andahuaylas, 07 de Diciembre de 2018.



Dr. Yalmar Temistocles Ponce Atencio
PRESIDENTE DEL JURADO DICTAMINADOR



MSc. Carlos Yinmel Castro Buleje
PRIMER MIEMBRO DEL JURADO DICTAMINADOR



MSc. Richard Carrión Abollaneda
SEGUNDO MIEMBRO DEL JURADO DICTAMINADOR

DEDICATORIA

A mí querida madre, Luzmila Mariño Gonzales por su apoyo incondicional y mi esposa Anali Diana Guía Gonzales quienes son la razón de mí ser, A mis hermanos, A mi familia Mariños por su comprensión y apoyo en cada uno de los pasos que me he propuesto seguir, y todas las personas que de una y otra manera me han motivado para lograr mi objetivo

Ervin L. Cáceres Mariño

AGRADECIMIENTO

A mis padres por darme la vida y familia por su apoyo incondicional a lo largo de toda la carrera.

A la Universidad Nacional José María Arguedas y la Facultad de Ingeniería de Sistemas por acogerme y permitirme formar parte de esta gran familia y alcanzar mis metas.

Al Dr. Julio César Huanca Marín e Ing. Leónidas Asto por la ayuda en el transcurso del desarrollo de esta investigación.

A mis profesores por todos sus consejos y conocimientos compartidos para mi formación profesional.

ÍNDICE

APROBACIÓN DEL ASESOR	ii
ACTA DE SUSTENTACION	iii
APROBACIÓN DEL JURADO DICTAMINADOR	iv
DEDICATORIA.....	v
AGRADECIMIENTO	vi
RESUMEN.....	xv
ABSTRACT	xvii
CHUMASQA.....	xix
CAPITULO 1: INTRODUCCIÓN.....	1
1.1. Planteamiento del problema.....	3
1.1.1. Formulación del problema	5
1.1.2. Objetivos	5
1.1.2.1. Objetivo General	5
1.1.2.2. Objetivos Específicos.....	5
1.2. Justificación.....	6
1.3. Viabilidad de la investigación.....	8
1.3.1. Viabilidad técnica.....	8
1.3.2. Viabilidad económica.....	8
1.3.3. Viabilidad social.....	9
1.3.4. Viabilidad operativa.....	9
1.4. Limitación del estudio	9
CAPITULO 2: MARCO TEÓRICO.....	10

2.1. Antecedentes	10
2.1.1. Antecedentes Internacionales.....	10
2.1.2. Antecedentes Nacionales.....	13
2.2. Marco conceptual de la Biometría.....	16
2.2.1. Seguridad Biométrica.....	16
2.2.2. Estructura general de un sistema biométrico.....	17
2.2.3. Aplicaciones de la biometría en seguridad	17
2.3. Visión por computadora.....	18
2.3.1. Procesamiento Digital de Imágenes	18
2.3.2. Análisis de Imágenes	19
2.3.3. Pixel.....	19
2.3.4. Adquisición de imágenes	20
2.4. Reconocimiento facial.....	20
2.4.1. Algoritmos para la detección facial	21
2.4.2. Algoritmo de Viola Jones	21
2.4.3. Redes neuronales	23
2.5. Algoritmo para reconocimiento facial	23
2.5.1. Análisis de componentes principales (PCA)	23
2.5.2. Algoritmo análisis de discriminación lineal (LDA)	25
2.5.3. ORB (RÁPIDO ORIENTADO y ROTO GIRADO)	26
2.5.4. Aprendizaje automático, Máquinas de vectores de soporte SVM	27
2.6. Dispositivos móviles	28
2.6.1. Aplicación móvil	29
2.7. Sistemas operativos móviles	29
2.7.1. Sistema operativo Android.....	29
2.7.2. Sistema operativo iOS	29

2.7.3. Sistema operativo Windows Phone	30
2.8. Cloud computing.....	30
2.8.1. Cloud computing Azure.....	31
2.8.2. Microsoft Face API.....	32
2.8.3. Cloud computing Bluemix	33
2.8.4. Watson Visual Recognition	34
2.8.5. Cloud computing Anaconda.....	35
2.9. OpenCV.....	36
2.10. Luxand, Inc.....	37
2.10.1. Luxand FaceSDK	37
2.11. Base de datos MYSQL	38
2.11.1. Orígenes de MySQL	38
2.11.2. Lenguaje de programación PHP.....	39
2.12. MADAMDM	39
2.12.1. Conclusión de las Metodologías Analizadas	40
2.12.2. Estructura de la Metodología MANDAMDM	40
2.13. Gestión de proyectos PMBOK.....	43
2.13.1. Procesos de la Dirección del proyecto.....	44
2.13.2. Áreas de conocimiento.....	45
CAPITULO 3: MATERIALES Y MÉTODOS	46
3.1. Tecnología de reconocimiento Facial.	46
3.2. Infraestructura y uso de Cloud Computing.....	48
3.3. Lenguajes y tecnologías para la lógica de negocio o Back-End.....	49
3.4. Base de datos.....	50

3.5. Aplicación móvil.....	50
3.6. Dirección del proyecto según las áreas del conocimiento de la guía del PMBOK.	50
3.6.1. Gestión de la integración del proyecto.....	51
3.6.2. Gestión del alcance del proyecto	53
3.6.3. Gestión del tiempo del proyecto	54
3.6.4. Gestión de los costos del proyecto	56
3.6.5. Gestión de la calidad del proyecto.....	57
3.6.6. Gestión de los Riesgos	58
3.6.7. Gestión de los Interesados	61
3.7. Puesta en marcha de la metodología MADAMDM.	61
3.7.1. Fase de Requerimientos	61
3.7.1.1. Requerimientos funcionales.....	62
3.7.1.2. Requerimientos no funcionales.....	62
3.7.2. Fase de Planificación	63
3.7.2.1. Diagramas de casos de uso	63
3.7.2.2. Diagrama de actividades.....	69
3.7.3. Fase de Diseño	71
3.7.3.1. Diseño del sistema.....	72
3.7.3.2. Entrenamiento del sistema.	77
3.7.4. Diseños de la Interfaces Abstractas	79
3.7.5. Funcionamiento.....	82
3.7.5.1. Pruebas de usabilidad.....	82
3.7.5.2. Pruebas funcionales.....	83
3.7.5.3. Pruebas de portabilidad	83
CAPITULO 4: RESULTADOS Y DISCUSIÓN	84
4.1. Análisis de resultados.....	84

4.2. Efectividad y confiabilidad en el sistema.....	84
CONCLUSIONES.....	87
RECOMENDACIONES.....	89
REFERENCIAS BIBLIOGRÁFICAS.....	90
Anexo A. Requisitos necesarios de pre-instalación.....	94
Anexo B: Manual de usuario.....	95
Anexo C: Manual de programador.....	98
Anexo D: tabla de personas con fotografías para pruebas en MySQL.	104
Anexo E: Errores encontrados al momento del reconocimiento con OpenCV.	105
Anexo F: Sujetos de prueba para reconocimiento Facial.	106
Anexo G: Código Fuente	112

Índice de tablas

Tabla 1: Comparativo de SIFT, SURF y ORB	26
Tabla 2: Comparativo de ID's para desarrollo de aplicación móvil.....	50
Tabla 3. Actividades para el desarrollo de la investigación.	51
Tabla 4. Roles y responsabilidades de la investigación.....	55
Tabla 5. Gestión de calidad del proyecto	57
Tabla 6. Descripción de calidad de proyecto.....	58
Tabla 7. Plan de gestión de riesgos	58
Tabla 8. Requerimientos funcionales	62
Tabla 9. Requerimientos no funcionales	62
Tabla 10. Especificaciones de captura de foto	66
Tabla 11. Especificaciones de detector de rostro	67
Tabla 12. Especificaciones de identificación de la persona.....	67
Tabla 13. Especificaciones de reporte de persona.....	68
Tabla 14. Base de datos del diseño	71
Tabla 15. Resultados obtenidos durante el día	85
Tabla 16. Resultados obtenidos durante la noche.....	86
Tabla 17. Resultados totales.....	86
Tabla 18. Requisitos de hardware y software.....	94

Índice de figuras

Figura 1: Estadística de mujeres que sufrieron violencia sexual a nivel mundial	3
Figura 2: cifra estadístico de abuso contra la mujer de todas las edades en Estados Unidos.	4
Figura 3: Arquitectura de un sistema de reconocimiento biométrico.....	17
Figura 4: Separar el objeto de estudio del fondo de la imagen	19
Figura 5: Análisis de la imagen radioscópica	19
Figura 6: Extracción de los pixeles.....	20
Figura 7: Esquema algoritmo Viola-Jones.....	22
Figura 8: Salida luego de usar el sistema de detector de rostros de viola jones.....	22
Figura 9: Procedimiento de implementación de PCA.	25
Figura 10: múltiples líneas que ofrecen una solución al problema	27
Figura 11: Solución al problema de la figura 10 con SVM.	28
Figura 12: Proveedores computación en la nube, proveedores SaaS / usuarios de la nube.	31
Figura 13: Verificación facial en FaceApi	33
Figura 14: Demo de detección de rostro con Watson.....	35
Figura 15: Metodología ágil para el diseño de aplicaciones multimedia-dispositivos móviles	40
Figura 16: Estructura MANDAMDM	41
Figura 17. Catálogos de acuerdo a sus roles	64
Figura 18.Catálogos de captura de rostro	64
Figura 19. Catálogos de detectar rostro de personas.....	65
Figura 20. Catálogos de identificación de persona.....	65
Figura 21. Catálogos de reporte de estado de persona.....	66
Figura 22. Diagrama de actividades.....	69
Figura 23. Diagrama de actividades – captura de fotos.....	70
Figura 24. Diagrama de actividades – detectar rostro	70

Figura 25. Diagrama de actividades – Identificar rostro.....	71
Figura 26. Modelo lógico.....	72
Figura 27. Modelo conceptual de la solución	73
Figura 28. Pasos de detección de rostro.....	74
Figura 29. Puntos de referencia.....	76
Figura 30. Software Entrenar sistema.....	78
Figura 31. Adicionar persona de la Base de datos BDdelincuentes.....	78
Figura 32. Imágenes de prueba.....	79
Figura 33. Sistema login	80
Figura 34. Interfaz gráfica Tomar Foto.....	80
Figura 35. Interfaz gráfica “Detectar rostro.....	81
Figura 36. Interfaz gráfica Identificar Persona.....	81
Figura 37. Interfaz gráfica Alerta de persona.....	82
Figura 38. Representación grafica	85
Figura 39. Representación gráfica -Resultados obtenidos durante la noche	86
Figura 40. Representación gráfica de resultado final.....	86

RESUMEN

El presente proyecto de investigación propone una aplicación móvil de reconocimiento facial en personas buscadas por la justicia en especial por el delito de abuso sexual, se centra en este pequeño grupo para medir la eficacia de la aplicación. Este será una aplicación de reconocimiento facial destinado a la identificación y reconocimiento de personas en tiempo real, llevado a cabo en la dirección zonal de Andahuaylas, para ser aplicado a la localización de individuos que se encuentran en una lista denominada “personas requisitorados por la justicia”, con el objetivo principal de incrementar los índices de seguridad colectiva brindando un sistema robusto que pueda ser utilizado en diversos lugares concurridos como: terminales, Carreteras transoceánicas, avenidas concurridas de personas, etc.

La aplicación móvil por medio de una web cervices se conectara al Face API y base de datos de la policía y pueda realizar las consultas, este permitirá que el intercambio de información sea en un menor tiempo posible en formato JSON. con el uso de un celular con cámara integrada de 5mp y sistema operativo Android 6.0 en adelante, realizara la adquisición de la imagen del rostro de una persona posterior realizara un análisis de la imagen adquirida mediante el uso de funciones y librerías para el procesamiento de imágenes si dentro de ella existe rostros humanos y luego, realizara la comparación de las características faciales que hacen únicas a las persona como el tamaño de los ojos, boca, quijada, entre otros puntos principales o patrones que tenemos las personas comparándolas con los rostros de delincuentes entrenados en el Face API, permitiendo identificar a la persona buscada y de esta forma se envía una alerta en tiempo real con un reporte del tipo de delito, en qué lugar esta requisitorado por la justicia, Alias con el que se hace conocer, y el nombre de la persona reconocida, de esta forma permitir al efectivo policial proceder con la captura del sujeto.

Por medio de un análisis de las técnicas existentes utilizadas en el reconocimiento facial, se estableció la aplicación de servicios cognitivos de Microsoft Azure, siendo aplicada por medio de

la herramienta de programación Visual Studio y Android Studio, usando formato de llamadas JSON para lograr comunicar la aplicación móvil de reconocimiento facial con la plataforma en la nube donde estará alojado el algoritmo de reconocimiento facial y rostros de personas entrenadas para su reconocimiento.

Palabras Clave: Inteligencia artificial, reconocimiento facial, servicios cognitivos, Azure Face API.

ABSTRACT

This research proposes a mobile application for facial recognition in people wanted by the police, especially for those who committed the crime of sexual abuse. The application focuses on this smaller group due to the fact that we are interested in studying the effectiveness of the method. This application, for facial recognition, is oriented to the identification and recognition in a real-time of wanted people in the police station in Andahuaylas, by using a list called "people wanted by the police". The principal aim is to increase the collective safety indexes, by offering a system that could be used in many crowded places such as terminals, collective transport vehicles, crowded avenues, etc.

The mobile application through web services will connect to the Microsoft Face API and the police database so that queries can be made, allowing the exchange of information in less time than before. With the use of a modern cell phone with high quality integrated camera and Android operating system (> 6.0), the acquisition of images of people faces is uploaded on the application, then analyze the images acquired through the use of functions and libraries for the image processing. Then, a comparison is made of the facial features that makes the person unique, such as the size of the eyes, mouth, jaw, among other characteristics, comparing them with the faces of individuals who has been requested by justice, those who was already trained in the Face API, thus allowing the identification of wanted people. In this way it is possible to send an alert in real time with a report of the place where the recognized person has been located, allowing police officers to proceed with the capture.

Analyzing the existing techniques used in facial recognition, the cognitive application services of Microsoft Azure with Visual Studio and Android Studio programming tools were used. In order to communicate the mobile application with the cloud platform through JSON format queries, where the face recognition application and the trained database of wanted people are hosted.

Keywords: Artificial intelligence, facial recognition, cognitive services, Azure Face API.

CHUMASQA

Kay aplicación móvil uya riksiq chaynapi maskanapaq warmi abusaq mañasqa runakunata, kayqa teléfonos móviles Smartphone nisqanpaqmi kanqa. Chayraykum rurasq huk prototipo de sistema nisqanta hinay ratulla abusakuq runapa uyan riqsiqta, kayqa rurakunqa Andahuaylas llaqtapim, chaynapi justiciapa maskasqan runakunata tarinanpaq, paykunaqa kachkan kayna sutiyuk listapim “mayna maskasqa warmikunata abusariq”. Kaywanqa tariytam munani llapanchikpaq allin kawsay kanantam, kay sumaq sistema nisqanqa servikunqa qina purina lugarpim: terminalkunapim, ñankunapim, transoceánicas nisqanpim, hinallataq siqis purina ñanpi.

Kay aplicación móvil tarikunqa servicio web nishgampi, chaynallataq kanqa qapinasqa face api, hinaspa policiapa hatun yachay allchasqanwan, kay servikunqa tapukuy ruwanapaq, chaynapi hinayratulla tapukuyta yacharunapaq. Kay musuq celular kanqa allin foto qurqunayuq chaynallataq sistema operativo Android nisqayuq suqta versionmanta wichayman, kaywanqa tarikusun runakunapa uya richkaqninta, chaynapi yachaspa hinayataq ukunchaspa tuparichinanpaq uyankuta. Chaymantataq tuparichinqa ñawinta, siminta, kaquichunta quk richkarininkuntawan maskasqa runapa face api yachasqanwan. Kaynapi tariruspa hutqayllaman aparachinqa chay runa maylawpim tarikusqanta chaynapi policiakuna chay runata ratulla qapirunanpaq.

Hukuncharispa kay yachaykunata uyarisqiyman, hapiruni kay ruranaypaqa microsoft azure nisqanta, visual studio nisqanta chaynallataq Android studio nisqanwan kaykunawanmi hayparuni kay aplicación móvil uyarisiq rurayta, hinaspataq allcharuni puyupi kay rurasqayta.

Rimay kichariq: Inteligencia artificial, uyariqsiq, servicios cognitivos, Azure Face API.

CAPITULO 1: INTRODUCCIÓN

Con el avance de los años, el reconocimiento facial se ha categorizado como una de las aplicaciones más estudiadas en áreas como la biometría y otros campos como la psicología, reconocimiento de patrones y procesamiento de imágenes. El prototipo de sistema de reconocimiento facial de personas buscadas tiene como objetivo principal el incrementar los índices de seguridad y vigilancia.

A nivel global se han desarrollado proyectos relacionados con sistemas de visión por computadora en especial china con el famoso proyecto “ojo de dios”, para ser utilizados en diversos lugares como, por ejemplo: aeropuertos, terminales terrestres, centros comerciales, etc. Permitiendo asistir en la mejora de la seguridad colectiva, brindando alertas en la identificación de individuos considerados peligrosos.

El trabajo investigativo consta de 4 capítulos distribuidos de la siguiente manera:

Capítulo 1, corresponde a la introducción del proyecto haciendo un análisis sobre el planteamiento del problema existente en la actualidad. (En el Perú no existe ningún sistema de búsqueda de personas peligrosas) No existe ningún sistema de búsqueda de personas peligrosas para solucionar este problema en Perú, de la misma forma la justificación del porque se busca realizar un prototipo de sistema de reconocimiento facial y se plantean objetivos que servirán como base en el desarrollo del prototipo.

Capítulo 2, detalla el marco teórico en el cual se detallan los aspectos teóricos para comprender los diferentes sistemas de visión artificial existentes, realizando un detallado estudio de estas diferentes técnicas de reconocimiento facial en el cual se analiza cada una de las etapas que implica el reconocimiento facial.

Capítulo 3, señala los materiales y métodos a utilizar que indica de manera detallada el desarrollo del prototipo de sistema de reconocimiento facial para la identificación de personas consideradas peligrosas, además se detalla de manera clara el desarrollo de la propuesta planteada: Los casos de uso del manejo de la aplicación, los diagramas de actividades de

interacción entre el usuario y la aplicación, el diseño de la interfaz gráfica de usuario, el diseño de la base de datos, la programación de la aplicación, e identificar las actividades necesarias para desarrollar una solución.

Capítulo 4, Señala las conclusiones y recomendaciones obtenidas durante el desarrollo del sistema.

1.1. Planteamiento del problema.

Las mujeres son las más vulnerables a ser víctimas sexuales en todo el mundo, según la Organización Mundial de la Salud (OMS). El 35% de las mujeres en todo el mundo han sufrido al menos alguna vez violencia sexual por parte de sus parejas, familiares o de desconocidos y solo se estima que el 5% de ellas se atreven a denunciar, una situación insostenible a nivel mundial que se agrava en los casos de algunos países.

En la figura 1 se muestra el porcentaje de mujeres de 15 a 49 años de edad que alguna vez tuvieron parejas y que informaron haber sufrido violencia sexual, como se observa nuestro país ocupa unos de los primeros lugares.

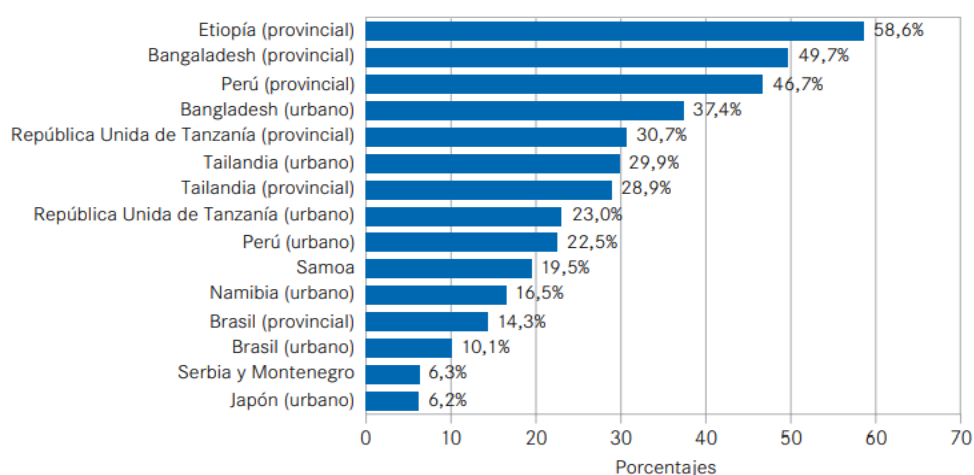


Figura 1: Estadística de mujeres que sufrieron violencia sexual a nivel mundial

Fuente: Organización Mundial de la Salud publicado en el 2017.

El sistema de detección y reconocimiento facial es utilizado por varios países europeos y asiáticos como China, Canadá, Estados Unidos entre otros países en vías de desarrollo. De esta manera se pretende disminuir la inseguridad ciudadana en todos sus aspectos. Permitiendo determinar con rapidez la identidad de una persona luego de capturar la imagen de su rostro y verificar características de su fisonomía comparándolas con las que están registradas en una base de datos.

La página de (Fabien, 2012) contienen las estadísticas de casi todos los países respecto a su economía, salud y lo más importante seguridad. Extrayendo como fuente de la oficina de las naciones unidas contra la droga y el delito (UNODC) el cual

nos muestra en la figura 2, por ejemplo cómo ha disminuido los casos de violaciones contra la mujer en Estados Unidos, el índice de violaciones bajo considerablemente debido a que la policía (específicamente el departamento del FBI) usa recursos como el de reconocimiento facial entre otras tecnologías de última generación.



Figura 2: cifra estadístico de abuso contra la mujer de todas las edades en Estados Unidos.

Fuente: (Fabien, 2012)

Perú no tiene mucho control sobre este problema social ya que no se cuenta con el recurso tecnológico ni apoyo de las autoridades competentes, 1.154 mujeres han sido víctimas de violencia sexual en los 2 primeros meses del año 2018 según el Ministerio de la Mujer y Poblaciones Vulnerables (MIMP).

Los casos de violación sexual tienen mayor incidencia en las siguientes regiones: Lima 446, Junín 91, Cusco 89, Arequipa 73, Huánuco 58, reportado de enero a marzo (MIMP, 2018)

En Apurímac los índices de violencia sexual son bajos, pero podría ser este el punto de partida para disminuir estos índices de violencia a la mujer (INEI, 2016).

Existe un portal web lanzado el 16 febrero del 2017 por el Ministerio del Interior, donde se encuentran publicadas fotografías de todas las personas que son miembros de una organización criminal, personas vinculadas a ella o que actúen por encargo de la

misma, organizaciones terroristas, así como presuntos autores y partícipes de uno o más delitos, cualquier ciudadano puede llamar a su número publicado en dicha página para denunciar anónimamente luego de eso recibir una recompensa si se logra su captura (MIP, 2017).

Este programa de recompensas lo primero que se debe de hacer es ingresar a la plataforma y memorizar los rostros de las personas que están siendo buscadas y para eso tener previo conocimiento del uso de internet.

En el Perú esto aún es un sueño de muchos aficionados a la tecnología ya que solo se han creado prototipos en algunas universidades para dar a conocer la funcionalidad de estas tecnologías de inteligencia artificial o tan solo dar a conocer el concepto.

1.1.1. Formulación del problema

¿Existe alguna aplicación móvil de reconocimiento facial para identificar a personas con antecedentes por abuso sexual?

1.1.2. Objetivos

1.1.2.1. Objetivo General

Desarrollar una aplicación móvil de reconocimiento facial para identificar a personas con antecedentes de abuso sexual.

1.1.2.2. Objetivos Específicos

- Realizar pruebas de eficacia, tiempo de respuesta a los algoritmos pre entrenados para detección de rostros.
- Desarrollar un prototipo de reconocimiento facial para personas con antecedentes de abuso sexual.
- Desarrollar un entrenador del reconocimiento facial con imágenes almacenados en una base de datos en tiempo real.

1.2. Justificación

El aprendizaje automático (del inglés, "Machine Learning") ha sido el favorito de muchos investigadores y desarrolladores y empresas privadas como google, Microsoft, IBM entre otros quienes desarrollan APIS cada vez más interesantes para desarrolladores ya sea en reconocimiento facial, objetos entre otros. Y además ofrecen computación en la nube (del inglés cloud computing) por medio de este captan la atención de empresas y desarrolladores ofreciéndoles servicios gratuitos como se hizo mención líneas arriba (reconocimiento facial, reconocimiento de objetos entre otros servicios) algunos servicios de manera gratuita y otros pagando según al volumen de datos utilizados.

Por ejemplo, las empresas que dan servicio de reconocimiento facial mediante sus APIS o SDK son: IBM con su cloud computing Bluemix y mediante este se puede hacer uso de su API Watson, el de Microsoft esta Azure mediante este se puede utilizar a su API Face, y de Google esta Google cloud este ofrece su API Visión. Si se hace uso de este servicio (reconocimiento facial) estas empresas cobran según la cantidad de imagen procesada, por ejemplo, IBM permite crear un modelo personalizado gratuito utilizando hasta 5000 imágenes de entrenamiento que permite clasificar hasta 7.500 imágenes por mes utilizando cualquier modelo preformado o personalizado con un límite de 250 imágenes por día (IBM, 2018).

La investigación sobre el reconocimiento de rostros nace en los años 60 (Narambuena, Vaiman, & Pereno, 2016) desde ahí que ha crecido mucho respecto a las investigaciones que hacen a esta materia de reconocimiento facial y las grandes potencias lo están estudiando para su implementación ya sea en seguridad, salud, educación, militar (Bravo, Ramírez, & Arenas, 2018).

Investigadores y científicos se encuentran optimizando los algoritmos de reconocimiento facial (Gómez & Mayumi, 2016) que va de la mano con las tecnologías de inteligencia artificial con redes neuronales, donde se está entrenando a las máquinas para poder reconocer objetos en tiempo real, ya sea a personas, cosas, etc. con tiempos de respuesta muy bajas.

Estas y más herramientas existen para poder implementar un prototipo que pueda ser utilizado para el público en general fácil de utilizar, más aún si se orienta a la seguridad ciudadana y posteriormente se pueda escalar a más, no necesariamente con los servicios de las empresas privadas que se mencionó líneas arriba si no que hay algoritmos implementados en librerías de código abierto como es Tensorflow y OpenCV las cuales serán descritas más adelante de acuerdo a su funcionalidad.

El sistema será novedoso porque hasta la fecha no hay ningún aplicativo similar, además en el futuro este podría ser conectado a la base de datos de la policía y un ciudadano común y corriente podrá con el uso de la cámara del teléfono inteligente identificar a sujetos con antecedentes de violación y tal vez se ofrezca alguna recompensa por la denuncia de este sujeto y así llevarse alguna gratificación.

Es por ello, que este proyecto busca empezar a identificar mecanismos con los cuales se puedan identificar aquellas personas reportadas como violadores o personas que son un peligro para la sociedad implementando soluciones tecnológicas que sean más eficaces y que permitan complementar a los actuales sistemas de búsqueda que existen en el país.

Según medios internacionales online como Elcolombiano, Pijamasurf, Lavanguardia, Telesurtv, destacan a ciudades inteligentes como Japón, Chima, Londres, Nueva York y Suiza, en utilizar las nuevas tecnologías en su transporte público, seguridad ciudadana y salud.

China es el primer país en exponer al mundo la implementación del uso de reconocimiento facial con sus 140 millones de cámaras de vigilancia con inteligencia artificial y son parte de una revolución tecnológica en materia de seguridad ciudadana que servirá para mantener bajo control a una población de no más de 1.400 millones de habitantes. Pero esta novedosa tecnología de reconocimiento facial solo esta implementada en sus cámaras ubicadas estratégicamente en las calles y en lentes de policías que pueden reconocen facialmente con la ayuda de un software con inteligencia artificial conectado a un Hardware especial que pueda soportar esta tecnología. Y no

solo eso, sino que además vigilan y rastrean en cada paso que se da, además, se le asigna al ciudadano una puntuación según su comportamiento (diario la república, 2018). En el Perú solo se puede apreciar esta tecnología en teléfonos de gama alta que usan reconocimiento facial como mediada de seguridad de acceso al equipo.

1.3. Viabilidad de la investigación

1.3.1. Viabilidad técnica.

Es viable técnicamente ya que existe suficiente documentación y soporte al respecto asimismo servicios gratuitos como librerías de código abierto para reconocimiento facial con OpenCV o Face API de Microsoft Azure que ya tiene implementado y mejorado los algoritmos de detección facial y reconocimiento facial, y para implementar el código nativo se está pretendiendo realizar en Android Studio, Visual Studio 2017, Anaconda cloud y para administrar la base de datos se tiene en mente contar con PHP y MySQL. Esto debido que hay más soporte para implantar algoritmos pre entrenados, debido al corto tiempo del taller. Además del hardware propio que cumple con los requisitos mínimos para desarrollar este tipo de proyecto.

1.3.2. Viabilidad económica.

Es viable económicamente ya que se cuenta con muchos sitios de cloud computing para desarrollar aplicaciones móviles, algunos gratuitos como anaconda cloud, otros con costo como Microsoft Azure, Watson cognitive de IBM, estos cobran según la cantidad de datos o imágenes procesadas. para este proyecto se pretende utilizar Microsoft Azure, y Anaconda Cloud para dispositivos móviles con sistema operativo Android. Con todas estas consideraciones hacemos accesible económicamente el proyecto con un coste necesario de S/. 38,780.00 el cual puede ser solventado.

1.3.3. Viabilidad social.

Es viable socialmente porque la mayoría de ciudadanas y ciudadanos cuentan con un celular inteligente con sistema operativo Android, el 80.8% del total de personas de una familia tiene teléfono celular en Apurímac (INEI, 2016).

Además, se debe tener en cuenta que el proyecto no causa efectos negativos a la sociedad por el contrario este proyecto brindara seguridad y tranquilidad para jóvenes y adultos de todas las condiciones socioeconómicas.

1.3.4. Viabilidad operativa.

El proyecto a desarrollarse será operativo ya que cumplirá con la necesidad de la población policial para mejorar en la eficacia de sus funciones en la captura de personas con antecedentes policiales por abuso sexual. Siendo estos de bajo costo y eficiente en su objetivo. Cabe mencionar también que este proyecto servirá de base de estudio a proyectos futuros en la utilización de nuevas tecnologías de inteligencia artificial.

1.4. Limitación del estudio

- Existen limitaciones respecto a la desactualización de datos proporcionados por el INEI.
- El no tener acceso a la información detallada de los procesos de la Policía Nacional del Perú. debido a que constituye información confidencial.
- Esta investigación tendrá una duración de cuatro meses.
- Existen muchas investigaciones sobre algoritmos eficientes en tiempo de respuesta de reconocimiento facial pero no se dispone de un código fuente.
- Existen librerías implementadas sobre reconocimiento facial en equipos de gama alta mas no para móviles.
- El prototipo será ejecutado en la plataforma Android, el cual soporta desde la versión 6.0 en adelante.

CAPITULO 2: MARCO TEÓRICO

2.1. Antecedentes

2.1.1. Antecedentes Internacionales

Rodríguez (2017), en su trabajo de investigación “Análisis e implementación del algoritmo de detección facial de Viola-Jones”, se muestra los métodos más utilizados para el reconocimiento facial y en efecto porque es el más utilizado el algoritmo de viola jones para lo cual se plantea el objetivo general de Analizar el algoritmo de VIOLA-JONES para detección de rostros. No solo hace referencia al algoritmo de viola jones sino también a diferentes métodos para la detección de rostros, para esto se realiza un cuadro comparativo sobre la eficacia de cada algoritmo. Además demuestra que no es necesario de un costoso o sofisticado hardware para su funcionamiento ni de mucho tiempo de cómputo, plantea que este método se ha convertido en las últimas décadas en uno de los algoritmos más importantes y ampliamente utilizados para la detección de rostros en aplicaciones en tiempo real y precisión de detección relativamente alta, además extrae un cuadro comparativo sobre las ventajas y desventajas de utilizar este algoritmo que son las siguientes: Las ventajas para utilizar este algoritmo son la selección eficiente de características, invariante a escala y localización, utilización de filtros haar, en lugar de puntos de concordancia, el tipo de entrenamiento se puede usar para detectar diferentes objetos. Las desventajas que se mencionan son: Más eficiente en imágenes de rostros frontales, sensible a cambios de iluminación, la cascada deslizante puede obtener varias detecciones del mismo rostro; además hace mención a cuatro métodos importantes que son los más utilizados por investigadores y desarrolladores al igual que viola jones para reconocer rostros estos son: Patrones binarios locales (Local Binary Patterns, LBP), detección de rostros usando la distancia de Hausdorff, detección de rostros con segmentación de tono de piel, detección de rostros usando redes neuronales. Rodriguez deja claro que su trabajo es sobre la detección de rostros

mas no de entrenamiento profundo para detectar rostros sobre unas bases de datos u otro similar. El algoritmo de viola jones luego de implementar y hacer comparativos con el algoritmo de Patrones binarios locales (en adelante LBP), utilizó la librería de código abierto OpenCV y el lenguaje de programación Python en un sistema operativo Ubuntu 14.04 y para obtener las gráficas utilizó Matlab. Tras sus pruebas pudo extraer las características que componen el algoritmo de Viola-Jones estas son: Imagen integral, características haar, Adaboost y cascada de clasificadores, cada una de ellas hace que este algoritmo sea más robusto y requiera de menor capacidad computacional para su procesamiento además realizo pruebas en el tiempo de ejecución, el algoritmo de Viola-Jones demostró ser la más eficiente en el momento de detección y de requerir menor tiempo de ejecución (Rodríguez Baque, 2017).

Andrade & Guillermo (2015) realizo su trabajo de investigación para obtener una especialización en seguridad informática que titula “Autenticación por reconocimiento facial para aplicaciones web, utilizando software libre”. En su proyecto detalla que el reconocimiento facial nace en el año 1960 con sistemas semiautomáticos capaz de reconocer características principales de la cara como la nariz, boca, cejas y ojos. En este trabajo resalta dos factores importantes que se deben tener en cuenta para realizar un reconocimiento facial exitoso los cuales son; Pose: las imágenes de una cara varían debido a la pose de la cara con respecto a la cámara (frontal, 45 grados, arriba, abajo, ect.), y algunas características faciales como los ojos o la nariz pueden obstruir la imagen parcial o total. Presencia o ausencia de componentes estructurales: Características faciales como barba, bigote, y lentes pueden o no estar presentes y hay una gran variabilidad entre estos componentes incluyendo formas, color y tamaño. Expresiones fáciles: La apariencia de las caras están directamente afectadas por la expresión facial de la persona. Obstrucción: Las caras pueden estar parcialmente obstruidas por otros objetos. En una imagen con un grupo de personas, algunas caras podrían estar parcialmente obstruidas por otras. Orientación de la

imagen: Las imágenes de las caras varían directamente por las diferentes rotaciones del eje óptico de la cámara. Condiciones de imágenes: Cuando las imágenes son capturadas, factores como la iluminación (espectro, fuente de distribución e intensidad) y las características de la cámara (sensor, respuesta, lentes) afectan la apariencia de una cara. Lo otro que se extrae de esta investigación es que da a conocer diferentes métodos para el reconocimiento de rostros, además menciona que el cerebro humano es capaz de reconocer personas pero no siempre aciertan en reconocer un rostro, lo mismo sucede con las máquinas primero se tiene que entrenar a la computadora para que pueda reconocer a una persona, para esto la computadora debe ser capaz de extraer las características más relevantes de las imágenes faciales de esta manera poder reconocer un rostro humano. Los métodos que hace mención y define detalladamente son: Entre los métodos más básicos se encuentran Eigenfaces o Principal Component Analysis (PCA), Fisherfaces o Linear Discriminant Analysis (LDA) y otros métodos más avanzados como Kernel Methods, 3D face recognition methods, Gabor Wavelets method, Hidden Markov Models y Active Appearance Models. Para lograr sus objetivos específicos se utilizó la librería OpenCV en su proyecto, con esta librería logro autenticar a usuarios legítimos al sistema web y para guardar información de sus usuarios se usó la base de datos MySQL. Luego de realizar pruebas a varios métodos para el reconocimiento facial se optó por elegir a los métodos Eigenfaces y Fisherfaces. Además su sistema tenía la funcionalidad de modificar o agregar usuarios con su imagen en la base de datos (Andrade & Guillermo, 2015).

Marín Posadas(2012) en su proyecto titulado “Prototipo de sistema de seguridad con reconocimiento facial”, se formuló los siguientes objetivos específicos: Desarrollar un método bajo técnicas de procesamiento de imágenes que permita reconocer el rostro de las personas que están autorizadas para llevar a cabo un objetivo, por medio de un registro previo en una base de datos y Analizar con la ayuda de una cámara digital

el rostro de la persona que se presente al sistema para saber si está autorizada, sino lo está, su rostro será registrado en la base de datos como no autorizado y se impedirá el acceso. Para lograr esto detalla los conceptos de visión por computadora, biometría, realidad aumentada, entre otros conceptos. Y se hace mención que para lograr un mejor resultado se debe tener en cuenta al software y hardware, este último debe de tener sensores para extraer las características deseadas. Se hace mención que la utilización del reconocimiento facial es ventajosa porque el nivel de eficacia es elevado, puede ser usado a distancia, es muy aceptado por usuarios, ocupa poco espacio, no requiere acciones por parte del usuario. las desventajas que menciona son que necesita actualizarse constantemente por los efectos de la edad y es sensible de iluminación. Para realizar la comparación entre dos imágenes utiliza la ecuación del coeficiente de correlación que es una covarianza normalizada, es decir, este método proporciona un criterio de comparación entre dos imágenes. El valor del coeficiente de correlación es invariable entre +1 y -1. El valor 1 indica la correlación más fuerte entre dos señales, es decir, el más similar. Lo que significa que corresponden a objetos homólogos con una gran certeza. Los que están cerca del 0 indica que no hay correspondencia es decir no existe correlación entre ambos valores. Debido al ruido de las imágenes la correlación perfecta 1, es muy rara. Por lo general se elige un valor como por ejemplo 0.75 y si el coeficiente de correlación excede dicho valor existe correspondencia entre los objetos analizados. Sus pruebas fueron implementadas en el software Matlab y los algoritmos en lenguaje c++, logrando completar sus objetivos planteados (Marín Posadas, 2012).

2.1.2. Antecedentes Nacionales.

López (2014) desarrolló su investigación titulado "Diseño e implementación de un sistema de reconocimiento facial usando Matlab" realizando un análisis a nivel de seguridad en Perú utilizado para empresas y sistemas que requieren alto grado de seguridad pero además manifiesta que casi todas las personas utilizamos tarjetas o

claves de seguridad que podríamos perder o podrían ser robados, cada ser humano tienen diferentes características faciales como los ojos, nariz, forma de la barbilla, pómulos, forma de la boca, etc. Estos rasgos podrían ser utilizados como seguridad de acceso. Estas características podrían utilizarse para reconocer a una persona por medio del software Matlab y una cámara web. Para cumplir con sus objetivos específicos y probar su hipótesis desarrollo su trabajo siguiendo 5 pasos: PRE procesamiento de las imágenes, detección de los rasgos biométricos del rostro (ojos, nariz y boca), extracción de dichos rasgos biométricos, comparación de similitud y toma de decisiones. Para el reconocimiento utilizó el método PCA, en Matlab y las imágenes procesadas se almacenaron con la extensión tipo JPG y sus parámetros de comparación fueron almacenados en un archivo XLS. Para una mejor interacción gráfica, utilizo la herramienta GUIDE del Matlab (López Ramírez, 2014).

Calli (2015) en su trabajo de “Reconocimiento facial basado en el algoritmo Eigenface” manifiesta que todas las personas hoy en día están sujetas a identificarse para lo cual hacen uso de su DNI, huellas dactilares, entre otros. En el presente trabajo de investigación plantea la hipótesis: “El conocimiento de herramientas de desarrollo para el procesamiento de imágenes de Matlab y los algoritmos de reconocimiento biométrico posibilita el diseño e implementación de una aplicación para el reconocimiento de rostros humanos”, para poder probar su hipótesis lo que hace es trabajar en imágenes bidimensionales estáticas y desarrollar un sistema que capture una fotografía, luego con el algoritmo de Eigenface, encontrar características importantes del rostro, la posición dentro de la imagen, marcar puntos estratégicos como el ojo la nariz, los labios luego por medio de otro algoritmo de procesamiento de imágenes PCA, comparar si la persona está dentro de las plantillas de imágenes para ser reconocido (Calli Olvea, 2015).

Collantes, et al. (2013). En este trabajo de investigación “Reconocimiento facial de personas basado en el análisis de componentes principales PCA” menciona que el

reconocimiento facial funciona mucho mejor en ambientes controlados es decir solo en los laboratorios donde se experimenta su funcionamiento y prueba de los algoritmos para este fin, realizándose con el software Matlab, llegando a las siguientes conclusiones: Mediante la aplicación de distintos filtros de Gabor, basados en determinadas orientaciones, frecuencias y tamaños de mascara se logró resaltar las principales características de las imágenes faciales utilizadas en el trabajo. A través de la aplicación del método del análisis de componentes principales PCA se logró la identificación facial de una persona minimizando los porcentajes de FAR y FRR de un conjunto de pruebas realizadas. La aplicación de los filtros de Gabor, resultado muy beneficioso debido a que permitió resaltar de mejor manera las características principales del rostro en las imágenes faciales utilizadas, con lo cual facilito el desempeño del método del análisis de componentes principales PCA, debido a que se redujo características innecesarias de las imágenes, mejorando de esta manera el reconocimiento. Para probar la mejor funcionalidad del algoritmo PCA se trabajó con imágenes transformadas previamente en escala de grises (Collantes Arana, Bustos, & Richard, 2013).

Villa (2015) planteó su investigación titulado “Sistema automático de reconocimiento de rostros” para cumplir con sus objetivos planteados, utilizo el software de Matlab y la Transformada Wavelet que es una herramienta matemática y el algoritmo de componentes principales PCA, este trabajo fue realizado en un ambiente controlado realizando un prototipo de madera con una cámara web además de contener su propia iluminación para la captura estándar de las fotos y así no tener problemas para el reconocimiento. sus conclusiones fueron: El uso de un fondo blanco para la captura de imágenes es bastante útil ya que se eliminan problemas de iluminación. El uso de la Transformada Wavelet y el Análisis de Componentes Principales ayudan a reducir los datos obtenidos perdiendo la menor cantidad posible de información, reduciendo de esta manera la carga computacional. El uso de la segmentación de imágenes es

de gran ayuda, ya que permite ubicar mejor el rostro al descartar aquello en la imagen que no es facial, como por ejemplo el fondo, y esto hace que el margen de error disminuya en gran medida. A mayor número de imágenes ingresadas en la base de datos, mayor tiempo de procesamiento tomará el programa para hacer los cálculos y poder obtener los resultados (Villa Palacios, 2015).

2.2. Marco conceptual de la Biometría

2.2.1. Seguridad Biométrica

Desde hace muchos años atrás las personas utilizan seguridad biométrica que son nada menos que el reconocimiento de la cara, la voz, los ojos, cicatrices, huellas dactilares, etc. (Serrano, Ortega, Araiza, & Araiza, 2018). Con estas características las personas se diferencian, los unos de los otros. La seguridad biométrica es utilizada por muchas organizaciones ya que este es un problema crítico para poder identificar personas en un mundo globalizado, es por esto que se pueden ver sus usos desde un cajero automático hasta el permiso de entrada a un país. El reconocimiento biométrico, definido como la técnica que posibilita la identificación automática de individuos basándose en sus características físicas o de comportamiento, está ganando gran aceptación como método para determinar la identidad de cada persona y se está utilizando en muchas aplicaciones tanto comerciales como públicos gubernamentales, en ámbitos tanto civiles como forenses (es decir, relacionadas con lo policial y lo legal). Mediante el uso del reconocimiento biométrico, es posible establecer la identidad de una persona mediante “algo que se es”, a diferencia de los tradicionales sistemas basados en “algo que se posee” (como un DNI, una tarjeta de identificación o una llave), que podría perderse o podría sufrir un robo, o en “algo que se sabe” (como una contraseña), que puede ser olvidado (J. O. García, Fernández, & Belmonte, 2008).

También indica García, que a partir de los años 90 se hace de la informática y la microelectrónica para hacer uso de la seguridad biométrica aplicado solo por los policías y forenses desde ese entonces las empresas prestan más atención en desarrollar aplicaciones para civiles en sus móviles como autenticación de acceso al mismo o para acceso a instalaciones que requieren de dicha tecnología para la identificación.

2.2.2. Estructura general de un sistema biométrico

Un sistema biométrico esencialmente es un reconocedor de patrones que captura datos biométricos de un individuo, extrae un conjunto de características a partir de dichos datos y las compara con otros patrones previamente almacenados en el sistema. Todos los sistemas de reconocimiento de patrones poseen una estructura funcional común formada por varias fases cuya forma de proceder depende de la naturaleza del patrón o señal a reconocer (J. O. García et al., 2008). La Figura 3 muestra esta estructura. En general el usuario únicamente tiene acceso al sensor, el cual captura el rasgo biométrico.

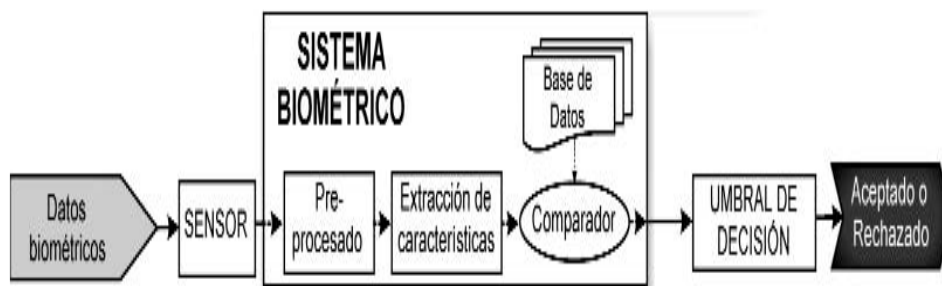


Figura 3: Arquitectura de un sistema de reconocimiento biométrico.
Fuente: García, Javier Ortega Biometría y seguridad.

2.2.3. Aplicaciones de la biometría en seguridad

La biometría es la disciplina que estudia las características de una persona para poder ser reconocido (Serrano et al., 2018) tiene muchas aplicaciones con respecto a la seguridad, pero solo las grandes empresas y países desarrollados lo utilizan en seguridad ciudadana para la identificación de delincuentes, controles de acceso

automático, para dar seguridad a los teléfonos inteligentes entre otros y los principales autenticadores que destacan son huellas dactilares, la geometría de la mano, la cara, el termograma facial, el iris, la retina, la voz, el estilo de la escritura, etc. Siendo todos estos como herramientas utilizadas por los policías para capturar a delincuentes a nivel mundial además de investigación criminal, también se hace uso en otras aplicaciones específicas como acceso a datos protegidos entre otros usos que podrían utilizar las empresas ya que su costo de implementación es alto para una persona común (Duró, 2001).

También podríamos mencionar su aplicación en los siguientes apartados: en cajeros automáticos, acceso a edificios, Mejora de la interacción hombre-máquina por ejemplo: un ordenador que reconoce quien lo está usando, Identificación de delincuentes en archivos policiales, Vigilancia, seguridad y seguimiento de personas (Moreano, Pulloquina, Lagla, Chisag, & Pico, 2017)

2.3. Visión por computadora

La visión por computadora o visión artificial es una rama de la inteligencia artificial que tiene como objetivo convertir en modelos matemáticos todo lo que podemos observar los humanos para luego ser tratados por una computadora, esto es representado en 3 dimensiones a partir de una o varias imágenes bidimensionales del mundo, estas imágenes pueden ser tratados en escala de grises o a color. Obteniéndose por una cámara ya sea desde un móvil o estacionaria (Estarita, Jiménez, Brochero, Escobar, & Moreno, 2018).

La visión por computadora es un término que viene siendo utilizado hasta la actualidad (Mery, 2004) pero muchos lo confunden con otros conceptos es por eso que Mery lo separa en términos muy sencillos.

2.3.1. Procesamiento Digital de Imágenes

Es el proceso donde se crea una versión de la imagen tratada.

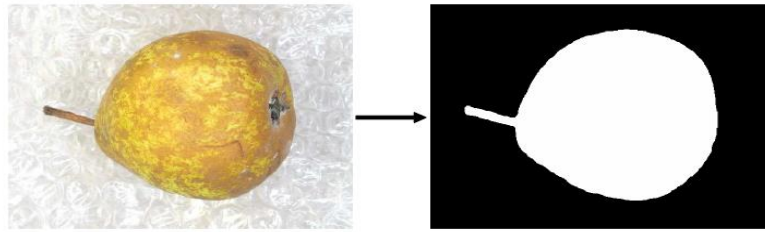


Figura 4: Separar el objeto de estudio del fondo de la imagen

Fuente: (Mery, 2004).

2.3.2. Análisis de Imágenes

Es el proceso mediante el cual una imagen después de ser procesada y obtenida su medición, es interpretada y posteriormente se toma una decisión.

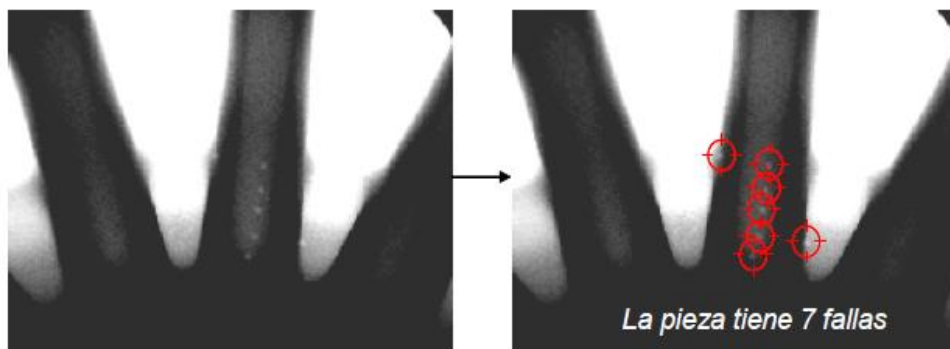


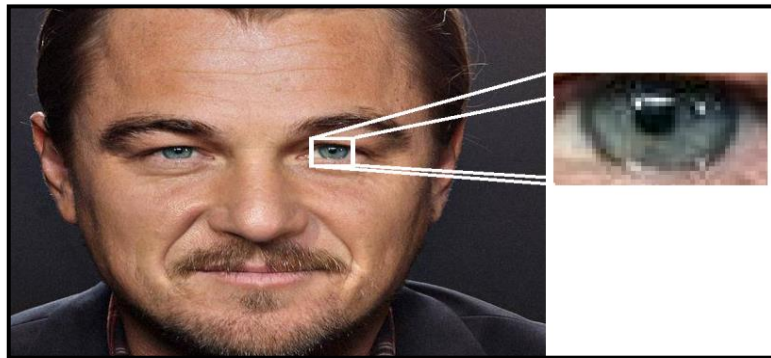
Figura 5: Análisis de la imagen radioscópica

Fuente: (Mery, 2004).

En el análisis de imágenes se pretende obtener cierta información a partir de una imagen. A través del análisis de la imagen radioscópica de la pieza mostrada se determina que la pieza no pasa el control de calidad por encontrarse defectuosa.

2.3.3. Pixel

Pixel es la abreviatura de Picture Element (Elemento de Imagen). Es la unidad mínima de una imagen, como se puede apreciar en la figura 4 al realizarle zoom en dicha imagen se puede observar los pixeles diminutos cuadrados donde se encuentran almacenado los colores(Sánchez & Elizabeth, 2014).



*Figura 6: Extracción de los píxeles
Fuente: elaboración propia*

2.3.4. Adquisición de imágenes

La adquisición de imagen se puede realizar desde distintos dispositivos digitales como teléfono celular, cámara fotográfica, cámara de video vigilancia, cámaras propias de tarjetas embebidas(Sánchez & Elizabeth, 2014).

2.4. Reconocimiento facial

En los últimos años la utilización del reconocimiento facial se ha extendido por todo el mundo y ha sido ampliamente utilizada en numerosas aplicaciones como sistemas de seguridad, sistemas de control de acceso a equipos o ambientes, identificación de conductas, entre otras. Más aún, debido a las mejoras notables en la tecnología, es posible para una computadora reconocer un rostro rápidamente, de manera que resulta práctico utilizar un sistema de este tipo para desbloquear un dispositivo móvil o identificarse en el acceso a una parte de un edificio con mayor nivel de seguridad que el resto.(Barraza, Thuillier, Will, & Rodriguez, 2013)

Es en el comienzo de 1970 donde el reconocimiento facial toma validez para los investigadores y desarrolladores (Parmar & Mehta, 2014) analizándose para reconocer a una persona mediante una fotografía por la computadora mediante la medición de la distancia entre los ojos u otros puntos importantes o medir diferentes ángulos de componentes faciales, (Parmar & Mehta, 2014) menciona en su trabajo que el reconocimiento facial es un problema pero a la vez interesante y desafiante que conlleva al análisis de diferentes puntos desde la psicología, reconocimiento de patrones, redes

neuronales, visión por ordenador, y gráficos por ordenador tras analizar todos es tos puntos llega a la conclusión de que la “investigación en el reconocimiento de caras es un área excitante para muchos años y mantendrán muchos científicos e ingenieros ocupados. En el futuro, 2D y 3D de reconocimiento de caras y aplicaciones a gran escala, tales como el comercio electrónico, identificación de estudiante, licencias de conducir digitales, o incluso de identificación nacional es la difícil tarea de reconocimiento de rostros y el tema está abierto a futuras investigaciones”.

2.4.1. Algoritmos para la detección facial

Existen diferentes algoritmos detección facial, estos tienen el objetivo de reconocer rostros en una imagen o video y si es encontrado este devuelve la posición y dimensión para cada rostro encontrado entre los más citados y comprobados por los científicos y desarrolladores tenemos a los algoritmos de viola jones y redes neuronales

2.4.2. Algoritmo de Viola Jones

El algoritmo de viola jones fue creado en el año 2001 por los ingenieros Paul viola y Michael janes (Viola & Jones, 2001), muchos desarrolladores y científicos realizaron pruebas con este algoritmo y con otros, como resultado este es el más eficiente y tiene bajo costo computacional, en la figura 5 se puede observar el recorrido de una imagen al ser aplicado por el algoritmo de viola jones, como se observa en dicha imagen se aprecia tres contribuciones un algoritmo que usa para la detección de objetos que utiliza la clasificación en cascada (HAAR) y un entrenador de clasificador basado en el algoritmo de aprendizaje AdaBoost, realizado esto una imagen integral (Viola & Jones, 2001) .

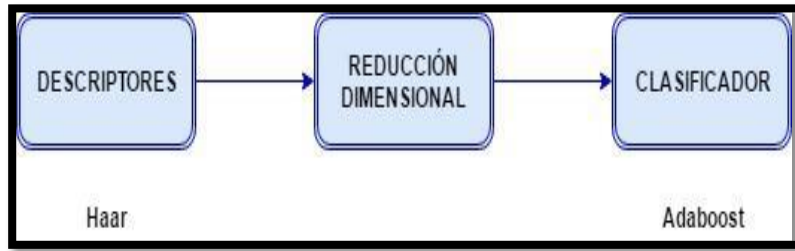


Figura 7: Esquema algoritmo Viola-Jones
Fuente: (González, 2017)

Este algoritmo tiene un funcionamiento en cascada mediante la concatenación de varios clasificadores débiles. Primero como en cualquier imagen se extraen los descriptores característicos, en este caso son de tipo Haar, y posteriormente se clasifican esos descriptores con un clasificador llamado AdaBoost o “Boosting”, que divide todas las imágenes procesadas en dos tipos de imagen, clase 1 “Cara” y clase 2 “No cara”. (González, 2017).

El algoritmo de Viola Jones supone un gran avance dentro de los algoritmos de detección por su gran rapidez, y porque la clasificación se realiza mediante características en vez de píxel a píxel, lo que permite una cierta abstracción del algoritmo respecto el resultado (Wang, 2014).

En definitiva, podemos concluir que Viola Jones no es solo una técnica de detección de caras o no caras, sino que es un algoritmo el cual goza de dos procesos bien diferenciados: la detección y la clasificación, en la que podemos ajustar su umbral de decisión para que cometa una menor tasa de error. (González, 2017).

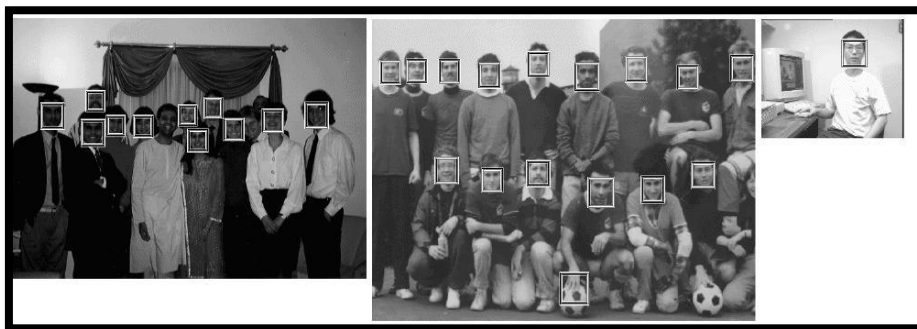


Figura 8: Salida luego de usar el sistema de detector de rostros de viola jones.
Fuente: (Viola & Jones, 2001)

2.4.3. Redes neuronales

Las redes neuronales artificiales son un paradigma de aprendizaje ampliamente tratado en la literatura de la inteligencia. Las redes neuronales artificiales intentan imitar la naturaleza de las neuronas del cerebro humano (P. P. G. García, 2013).

Algunos investigadores lo mencionan como un método importante para el reconocimiento de rostros, tienen casi los mismos objetivos que PCA o PLA para detectar rostros tales como la posición de los ojos u otras simetrías de rasgos faciales de las personas (Moreano et al., 2017) pero por su difícil implementación no es muy utilizado para dicho fin ya que se encontró poco material sobre la implementación del uso de este método.

Según (Sánchez & Elizabeth, 2014) las redes neuronales se han convertido en una técnica popular para el reconocimiento de patrones. Se usan generalmente para la clasificación de imágenes según patrones establecidos. La red se entrena usando un conjunto de imágenes que representan rostros de todo tipo y otro conjunto de imágenes que no representan rostros, de forma que la red neuronal pueda establecer el criterio adecuada acerca de lo que es y no es un rostro. Además las redes neuronales tiene un acierto igual mayor a 95% de encontrar un rostro habitualmente se usa en la categorización de imágenes según reglas establecidas, por ejemplo, saber si una imagen está borrosa o no (Gallardo Calvopiña & Sánchez Quevedo, 2016).

2.5. Algoritmo para reconocimiento facial

2.5.1. Análisis de componentes principales (PCA)

El análisis de componentes principales o Eigenface es un método propuesto por (Turk & Pentland, 1991), este método fue comprobado que es robusto, rápido y eficiente para llevar a cabo el reconocimiento facial (Eugenio, Manuel, Edgar, Alejandro, & Martin, 2015).

Mediante PCA uno puede transformar cada imagen original de la formación que figura en una Eigenface correspondiente. Una característica importante de la PCA es que uno puede reconstruir cualquier imagen original desde el conjunto de entrenamiento mediante la combinación de los Eigenface. Recuerde que Eigenface son nada menos que rasgos característicos de las caras. Por lo tanto, se podría decir que la imagen de la cara original puede ser reconstruida a partir Eigenface si uno suma todos los Eigenface (características) en la proporción adecuada. Cada Eigenface representa solo ciertas características de la cara, que puede o no puede estar presente en la imagen original. Si la característica está presente en la imagen original a un grado más alto. (Calli Olvea, 2015)

Una ventaja del algoritmo PCA es la reducción de los datos necesarios en la identificación para una relación 1 a 1000, este algoritmo utiliza Eigenface, las imágenes deben estar normalizadas y del mismo tamaño, así permiten el lineamiento de la boca y los ojos, el PCA resulta ser una poderosa herramienta para el análisis de los datos teniendo como objetivo general lo antes mencionado (Collantes Arana et al., 2013; González, 2017)

“Este método es un algoritmo de reducción dimensional, extrae las características principales de la imagen que mejor la define y las agrupa en diferentes vectores, cada uno de los cuales pertenece a una elipse” (González, 2017).

En la figura 7 se muestra el procedimiento que realiza este algoritmo para analizar un rostro los autores siguientes detallan más a detalle sobre el significado (Calli Olvea, 2015; Collantes Arana et al., 2013; González, 2017; Parmar & Mehta, 2014; Serrano et al., 2018; Villa Palacios, 2015)

1. Expresar cada imagen como vector columna I_k
2. Calcular el vector promedio $\mu = \frac{1}{M} \sum_{k=1}^M I_k$
3. Restar a cada imagen el promedio $x_k = I_k - \mu$
4. Obtener la matriz $A = [x_1 x_2 \dots x_M]$
5. Calcular los vectores y valores propios de la matriz $A^T A$
6. Seleccionar los n vectores con los valores propios asociados de mayor valor.
7. Obtener los vectores propios de la matriz de covarianza por medio de la relación $U_i = AV_i$. Donde V_i son los vectores obtenidos en el paso 6.
8. Normalizar los vectores obtenidos de tal forma que $\|U_i\| = 1$
9. Proyectar en el sub-espacio todos los vectores columnas x_k correspondientes a las imágenes de referencia para obtener los respectivos pesos en cada dirección $w_i = U_i^T x_k$

Figura 9: Procedimiento de implementación de PCA.

Fuente: (Martínez, Loaiza, & Caicedo, 2011)

2.5.2. Algoritmo análisis de discriminación lineal (LDA)

LDA se desempeña cuando las mediciones realizadas en variables independientes para cada observación son cantidades continuas. El objetivo es proyectar un conjunto de datos en un espacio de menor dimensión con una separabilidad de clases para evitar la sobre equipacion y también reducir los costos computacionales. Cuando se trata de variables categóricas independientes, la técnica equivalente es el análisis de correspondencia discriminante (Moreano et al., 2017).

Tanto el análisis discriminante lineal (LDA) como el análisis de componentes principales (PCA) son técnicas de transformación lineal que se utilizan comúnmente para reducir la dimensionalidad. La PCA se puede describir como un algoritmo "no supervisado", ya que "ignora" las etiquetas de clase y su objetivo es encontrar las direcciones (los denominados componentes principales) que maximizan la varianza en un conjunto de datos. A diferencia de PCA, LDA se "supervisa" y calcula las direcciones ("discriminantes lineales") que representarán los ejes que maximizan la separación entre múltiples clases.

Aunque podría sonar intuitivo que LDA es superior al PCA para una tarea de clasificación de clase múltiple donde las etiquetas de clase son conocidas, este no siempre es el caso.

Por ejemplo, las comparaciones entre las precisiones de clasificación para el reconocimiento de imágenes después de usar PCA o LDA muestran que PCA tiende a superar a LDA si el número de muestras por clase es relativamente pequeño. En la práctica, tampoco es raro usar tanto LDA como PCA en combinación: por ejemplo, PCA para la reducción de dimensionalidad seguida de una LDA (Raschka, 2014; Serrano et al., 2018)

2.5.3. ORB (RÁPIDO ORIENTADO y ROTO GIRADO)

(Rublee, Rabaud, Konolige, & Bradski, 2011) fueron quienes publicaron este algoritmo que es nada menos que la mejora de los algoritmos surf y sift, también se podría decir que es mucho mejor que los 2 algoritmos antes mencionados. Además, es mucho más ligero para ser implementado a un móvil porque no requiere de muchos recursos para su implementación.

(Ilbay, Fabricio, Córdova, & Gonzalo, 2013) presentan el siguiente cuadro:

Tabla 1: Comparativo de SIFT, SURF y ORB

Item	VENTAJAS	DESVENTAJAS
SIFT	Invariancia respecto de rotaciones, translaciones, Escala y cambios de Iluminación, repetibilidad.	Alto Coste computacional. Tamaño del descriptor mucho mayor que los anteriores. Discretización de los filtros gaussianos. Depende del tamaño
SURF	Mayor robustez y velocidad de cálculo respecto del descriptor SIFT.	Menor tamaño que el descriptor SIFT pero todavía incomparable respecto a de los anteriores. Dependiente del tamaño de la imagen
ORB	Mejor rendimiento en tiempo real, Dependiente del tamaño ideal para dispositivos de baja potencia sin aceleradores de CPU	Dependiente del tamaño de la calidad de la imagen, demasiado sensible a la rotación de imágenes.

Fuente: (Ilbay et al., 2013)

2.5.4. Aprendizaje automático, Máquinas de vectores de soporte SVM

(Sánchez & Elizabeth, 2014) resumen a SVM “es un método genérico para resolver problemas de reconocimiento de patrones. Dado un conjunto de puntos en un determinado espacio que pertenecen a dos clases distintas, SVM encuentra el hiperplano que separa la mayor cantidad de puntos de la misma clase del mismo lado. Esto se realiza maximizando la distancia de cada clase al hiperplano de decisión, denominado OSH (Optimum Separating Hyperplane). Los puntos más cercanos al hiperplano, son los llamados vectores soporte (Support Vectors)”.

(" OpenCV," 2018): define con un ejemplo su explicación se tiene la figura 10 la pregunta sería: ¿En qué sentido es el hiperplano obtenido óptimo? Consideremos el siguiente problema simple:

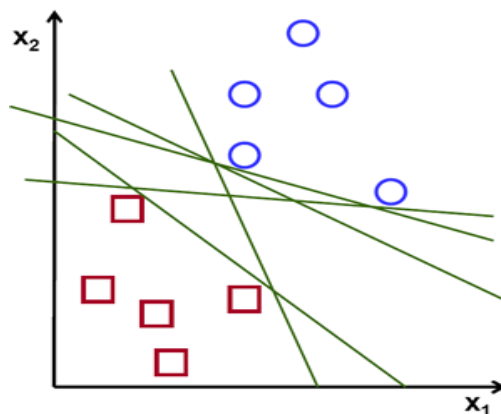


Figura 10: múltiples líneas que ofrecen una solución al problema
Fuente: (" OpenCV," 2018).

Para un conjunto separable linealmente de puntos 2D que pertenecen a una de dos clases, encuentre una línea recta de separación.

En la figura 10 de arriba se aprecia múltiples líneas que ofrecen una solución al problema. ¿Alguno de ellos es mejor que los demás? Podemos definir intuitivamente un criterio para estimar el valor de las líneas:

Una línea es mala si pasa demasiada cerca de los puntos porque será sensible al ruido y no se generalizará correctamente. Por lo tanto, nuestro objetivo debe ser encontrar la línea que pase lo más lejos posible de todos los puntos.

Entonces, la operación del algoritmo SVM se basa en encontrar el hiperplano que da la distancia mínima más grande a los ejemplos de entrenamiento. En dos ocasiones, esta distancia recibe el importante nombre de margen dentro de la teoría de SVM. Por lo tanto, el hiperplano de separación óptimo maximiza el margen de los datos de entrenamiento.

Después del análisis en la figura 11 se aprecia la solución al problema que se muestra en la figura 10.

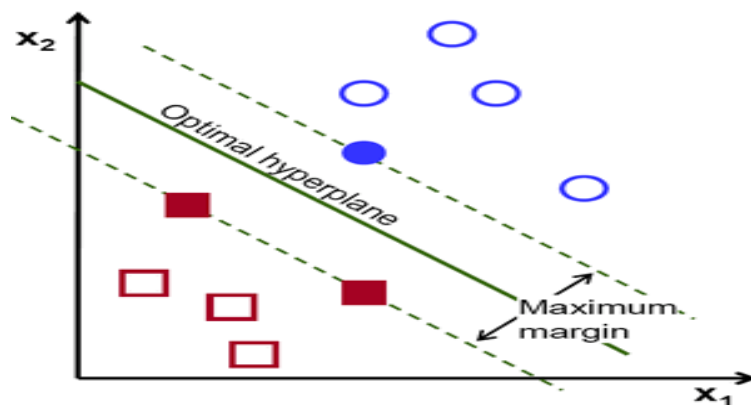


Figura 11: Solución al problema de la figura 10 con SVM.

Fuente: ("OpenCV," 2018)

2.6. Dispositivos móviles

Según (Ortega, González Ispuerto, & Pérez Peláez, 2015) en su trabajo de investigación concluyen diciendo "Nos encontramos ante una sociedad siempre conectada y plenamente digitalizada, que se encuentra en pleno proceso de redefinición del cómo asigna su tiempo a la hora de consumir productos y servicios de los medios de comunicación. Los nuevos dispositivos más utilizados para estas tareas son el teléfono inteligente y la tableta, pequeños aparatos táctiles que están cambiando el mundo de las

comunicaciones y las costumbres de los navegantes digitales de los medios, los digital-digito-nautas”

2.6.1. Aplicación móvil

El compañero (Cárdenas Taype, 2017) cita la siguiente frase del autor Cuello y Vittone la siguiente frase: “En esencia, una aplicación no deja de ser un software. Para entender un poco mejor el concepto, podemos decir que las aplicaciones son para los móviles lo que los programas son para los ordenadores de escritorio”.

2.7. Sistemas operativos móviles

Un sistema operativo móvil es el software básico de un teléfono inteligente básicamente provee al teléfono una interfaz entre el usuario, las aplicaciones y los dispositivos de hardware, además coordina la conectividad inalámbrica que sería uno de los factores más importantes en estos dispositivos (Ilbay et al., 2013).

2.7.1. Sistema operativo Android

Este sistema operativo está basado en Linux y orientado a dispositivos móviles, como teléfonos inteligentes y tabletas. Fue desarrollado inicialmente por Android Inc. Pero fue comprado por Google en el año 2015 (Ilbay et al., 2013).

Para desarrollar aplicaciones para este sistema operativo es necesario tener conocimiento al menos de código java utilizando las librerías propias de Android, por lo tanto un programador que tenga conocimientos de java no tendría dificultad para comenzar a programar en Android (Cárdenas Taype, 2017).

2.7.2. Sistema operativo iOS

Anteriormente llamado iPhone OS, este sistema operativo pertenece a la compañía Apple, no soporta java ni adobe flash pero si html5, se recomienda desarrollar aplicaciones para este sistema operativo desde una MAC y el IDE oficial de Apple (Ilbay et al., 2013).

2.7.3. Sistema operativo Windows Phone

Anteriormente llamado Windows Mobile, fue desarrollado por Microsoft, está basado en el núcleo del sistema operativo Windows CE y cuenta con un conjunto de aplicaciones básicas utilizando las API de Microsoft Windows (Ilbay et al., 2013) para desarrollar aplicaciones es necesario tener conocimiento de programación en C# el software donde fue desarrollado fue en visual Studio (Cárdenas Taype, 2017).

2.8. Cloud computing

Según (Armbrust et al., 2010) define a la computación en la nube, que es llevar todo lo que se realiza en una computadora local a servidores que ofrecen servicios de alojamiento o realizar todo lo que hacemos en servicios públicos que se encuentran en internet como crear documentos, hojas de cálculo, diseño gráfico, guardar archivos, etc. También podría decirse que La computación en la nube se refiere tanto a las aplicaciones entregadas como servicios a través de Internet como al software de hardware y sistemas en los centros de datos que brindan esos servicios.

Los servicios en sí mismos se conocen desde hace mucho tiempo como Software as a Service (SaaS). Algunos proveedores usan términos como IaaS (Infraestructura como Servicio) y PaaS (Plataforma como Servicio) para describir sus productos, pero evitamos estos porque las definiciones aceptadas para ellos todavía varían ampliamente. La línea entre la infraestructura de "bajo nivel" y una "plataforma" de mayor nivel no es nítida. Creemos que los dos son más parecidos que diferentes, y los consideramos juntos. Del mismo modo, el término relacionado "Grid computing", de la comunidad informática de alto rendimiento, sugiere protocolos para ofrecer computación compartida y almacenamiento a largas distancias, pero esos protocolos no condujeron a un entorno de software que creció más allá de su comunidad.

El hardware y el software del centro de datos es lo que llamaremos una nube. Cuando una nube se pone a disposición del público en general en una modalidad de pago por uso, la llamamos nube pública; el servicio que se vende es informática de servicios

públicos. Usamos el término nube privada para referirse a los centros de datos internos de una empresa u otra organización, que no están disponibles para el público en general, cuando son lo suficientemente grandes como para beneficiarse de las ventajas de la computación en la nube que discutimos aquí. Por lo tanto, la computación en la nube es la suma de SaaS y la informática de servicios públicos, pero no incluye centros de datos pequeños o medianos, incluso si estos se basan en la virtualización para la administración. Las personas pueden ser usuarios o proveedores de SaaS, o usuarios o proveedores de informática de servicios públicos. Nos enfocamos en los proveedores de SaaS (usuarios de la nube) y en los proveedores de la nube, que han recibido menos atención que los usuarios de SaaS. La figura 1 hace que las relaciones proveedor-usuario sean claras. En algunos casos, el mismo actor puede jugar múltiples roles. Por ejemplo, un proveedor de la nube también podría alojar sus propios servicios orientados al cliente en la infraestructura de la nube.

En la figura hace que las relaciones proveedor-usuario sean claras. En algunos casos, el mismo actor puede jugar múltiples roles. Por ejemplo, un proveedor de la nube también podría alojar sus propios servicios orientados al cliente en la infraestructura de la nube

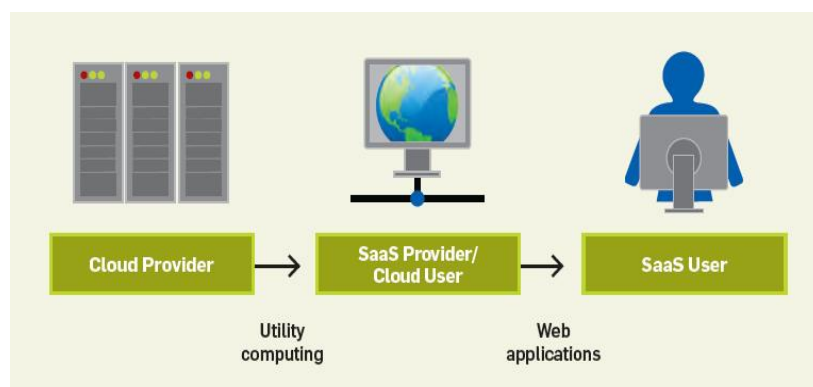


Figura 12: Proveedores computación en la nube, proveedores SaaS / usuarios de la nube.

Fuente: (Armbrust et al., 2010)

2.8.1. Cloud computing Azure

Esta plataforma fue lanzado por la corporación de Microsoft en el año del 2010 (Microsoft, 2010) hace 8 años atrás pero hasta la actualidad a lanzado muchas mejoras, y servicios gratuitos muy interesantes como el de reconocimiento facial.

(ggailey777, 18/10/2017) crea una cuenta en GitHub con ayuda de casi 12 colaboradores, estos realizan una documentación para desarrolladores en este se detalla más a profundo sobre los servicios y como iniciarse en Azure, para lo cual lo definen de la siguiente manera: Azure es una plataforma en la nube completa que puede alojar sus aplicaciones existentes, agilizar el desarrollo de nuevas aplicaciones e incluso mejorar las aplicaciones locales. Azure integra los servicios en la nube que necesita para desarrollar, probar, implementar y administrar sus aplicaciones, al tiempo que aprovecha las eficiencias de la computación en la nube. Al hospedar sus aplicaciones en Azure, puede comenzar desde cero y escalar fácilmente su aplicación a medida que crece la demanda de sus clientes. Azure también ofrece la confiabilidad que se necesita para aplicaciones de alta disponibilidad, incluso la conmutación por error entre diferentes regiones. El portal de Azure le permite administrar fácilmente todos sus servicios de Azure. También puede administrar sus servicios mediante programación mediante el uso de API y plantillas específicas del servicio.

2.8.2. Microsoft Face API

Face API tiene dos funciones principales: detección de rostros con atributos y reconocimiento facial. Detecta hasta 64 rostros humanos con una ubicación de la cara de alta precisión en una imagen. Que indica la ubicación de la cara en la imagen que se devuelve junto con cada cara detectada. Opcionalmente, la detección de rostros extrae una serie de atributos relacionados con la cara, como pose, sexo, edad, pose de cabeza, vello facial y anteojos.



Figura 13: Verificación facial en Face Api
Fuente: <https://azure.microsoft.com/en-us/services/cognitive-services/face/?v=18.05>

2.8.3. Cloud computing Bluemix

La innovadora plataforma de computación en la nube de IBM combina plataforma como servicio (PaaS) con infraestructura como servicio (IaaS) e incluye un amplio catálogo de servicios en la nube que se pueden integrar fácilmente con PaaS y IaaS para crear aplicaciones de negocios rápidamente. IBM Cloud (anteriormente Bluemix) tiene implementaciones que se adaptan a sus necesidades, ya sea que sea una empresa pequeña que planea escalar, o una empresa grande que requiera aislamiento adicional. Puede desarrollarse en una nube sin fronteras, donde puede conectar sus servicios privados a los servicios públicos de IBM Cloud disponibles en IBM. Usted y su equipo pueden acceder a las aplicaciones, servicios e infraestructura en IBM Cloud y usar datos, sistemas, procesos, herramientas PaaS y herramientas IaaS existentes. Los desarrolladores pueden aprovechar el ecosistema en rápido crecimiento de los servicios disponibles y los marcos de tiempo de ejecución para crear aplicaciones utilizando enfoques de programación polígota.

Con IBM Cloud, ya no tiene que hacer grandes inversiones en hardware para probar o ejecutar una nueva aplicación. En cambio, lo gestionamos todo por usted y solo le cobramos por lo que usa. IBM Cloud proporciona modelos de implementación integrada pública, dedicada y local.

Puede llevar una idea desde el inicio, al sandbox de desarrollo, a un entorno de producción distribuido globalmente con infraestructura de cómputo y almacenamiento, servicios y contenedores de plataforma de código abierto, y servicios de software y herramientas de IBM, Watson y más. Más allá de las capacidades de la plataforma en sí, IBM® Cloud también ofrece una implementación flexible. Aprovechone los recursos de la nube de IBM® localmente, en entornos de nube privada dedicada o en la nube pública.

Todos los recursos de la nube de IBM que se implementan en entornos públicos y dedicados se alojan desde ubicaciones de centros de datos en la nube de IBM® disponibles en todo el mundo. Los centros de datos en la nube de IBM brindan redundancia regional, una red troncal de red global que conecta todos los centros de datos y puntos de presencia, y estrictos controles de seguridad e informes. A través de IBM Cloud Data Centers, IBM puede satisfacer sus necesidades más exigentes de expansión, seguridad, cumplimiento y residencia de datos.

IBM le permite: Implemente infraestructura de cómputo y almacenamiento de alto rendimiento en centros seguros de datos en la nube de IBM en todo el mundo.

Pruebe y adopte una amplia gama de servicios y capacidades en la nube de IBM, comunidades de código abierto y desarrolladores de terceros (Bluemix, 2016).

2.8.4. Watson Visual Recognition

Este es un API que permite reconocer previo a un entrenamiento cualquier objeto a la vez analizar imágenes con algoritmos de aprendizaje profundo para identificar escenas, objetos y caras en las imágenes que carga en el servicio. Puede crear y entrenar un clasificador personalizado para identificar temas que se adapten a sus necesidades (Bello de Haro; Bluemix, 2016).



Figura 14: Demo de detección de rostro con Watson

Fuente: <https://console.bluemix.net/docs/services/visual-recognition>

2.8.5. Cloud computing Anaconda

Anaconda Cloud es un servicio de administración de paquetes. Hace que sea más fácil encontrar, acceder, almacenar y compartir portátiles, entornos, paquetes públicos y privados. Cloud también facilita mantenerse al día con las actualizaciones realizadas en los paquetes y entornos que está utilizando. La nube alberga cientos de paquetes, cuadernos y entornos de Python útiles para una amplia variedad de aplicaciones. Cloud es el Anaconda Repository de Anaconda Inc. ® puesto a disposición del público. El repositorio también está disponible para la compra por parte de compañías que desean impulsar su propia nube interna y privada. Para obtener más información, consulte Suscripciones a Anaconda.

Puede crear nuevos paquetes utilizando la interfaz de línea de comando (CLI) del Cliente Anaconda, luego cargarlos manualmente o automáticamente a la nube para compartirlos rápidamente con otros o acceder a usted desde cualquier lugar.

Para los desarrolladores, Cloud está diseñado para facilitar el desarrollo, lanzamiento y mantenimiento de software al proporcionar un amplio soporte de gestión de paquetes. La nube permite repositorios privados y públicos de paquetes pagos, así

como canales de paquetes, proporcionando un servicio flexible y escalable para grupos y organizaciones de todos los tamaños.

El alojamiento de paquetes de libre acceso siempre es gratis para individuos y organizaciones que alojan hasta 3 GB de paquetes. Hay una pequeña tarifa por el alojamiento de paquetes privados o más de 3 GB (Analytics).

2.9. OpenCV

OpenCV (Open Source Computer Vision Library) se lanza bajo licencia BSD y, por lo tanto, es gratuito tanto para uso académico como comercial. Tiene interfaces C ++, Python y Java y es compatible con Windows, Linux, Mac OS, iOS y Android. OpenCV fue diseñado para la eficiencia computacional y con un fuerte enfoque en aplicaciones en tiempo real. Escrito en C / C ++ optimizado, la biblioteca puede aprovechar el procesamiento de múltiples núcleos. Habilitado con OpenCL, puede aprovechar la aceleración de hardware de la plataforma informática heterogénea subyacente.

Adoptado en todo el mundo, OpenCV tiene más de 47 mil personas de usuarios y una cantidad estimada de descargas que supera los 14 millones. El uso va desde el arte interactivo hasta la inspección de minas, la costura de mapas en la web o la robótica avanzada. (" OpenCV," 2018).

La biblioteca cuenta con más de 2500 algoritmos optimizados, que incluyen un conjunto completo de algoritmos de visión artificial y de aprendizaje automático tanto clásico como avanzado. Estos algoritmos se pueden usar para detectar y reconocer rostros, identificar objetos, clasificar acciones humanas en videos, rastrear movimientos de la cámara, rastrear objetos en movimiento, extraer modelos 3D de objetos, producir nubes de puntos 3D desde cámaras estéreo, unir imágenes para producir una alta resolución imagen de una escena completa, encuentre imágenes similares de una base de datos de imágenes, elimine los ojos rojos de las imágenes tomadas con flash, siga los movimientos oculares, reconozca el escenario y establezca marcadores para superponerlo con realidad aumentada, etc. OpenCV tiene más de 47 mil personas de usuarios comunidad y número

estimado de descargas que exceden los 14 millones. La biblioteca se usa ampliamente en compañías, grupos de investigación y por organismos gubernamentales (García Sevilla, 2014; " OpenCV," 2018).

2.10. Luxand, Inc.

Luxand, Inc. es una empresa privada de alta tecnología formada en 2005. Las actividades de investigación Luxand comenzaron con la inteligencia artificial y las tecnologías de identificación biométrica, permitiendo a la compañía desarrollar un conjunto completo de herramientas y bibliotecas para realizar reconocimiento totalmente automático de rostros humanos y rasgos faciales. Hoy, la compañía ofrece una amplia gama de soluciones de reconocimiento facial. Los productos y las tecnologías de la compañía son utilizados por empresas de seguridad e identificación biométrica, bancos, industria del entretenimiento, industria médica y cosmética, en portales de entretenimiento en línea, salas de chat y sitios web de películas en todo el mundo. Las organizaciones policiales y militares están utilizando las tecnologías Luxand para identificar a las personas y localizar registros personales mediante fotografías faciales. Gobierno, Las entidades municipales y comerciales están utilizando nuestros productos en su flujo de trabajo de documentos para automatizar el proceso de ingreso de imágenes faciales en una base de datos. Las universidades y las instalaciones de investigación están utilizando nuestras herramientas y tecnologías para realizar investigaciones científicas.

2.10.1. Luxand FaceSDK

FaceSDK permite a los desarrolladores de Microsoft Visual C ++, C #, Objective C, VB, Java y Delphi crear aplicaciones de 32 y 64 bits para Web, Windows, Linux, MacOS, iOS y Android con funciones de reconocimiento facial e identificación biométrica basada en la cara (Luxand, 2010).

FaceSDK se utiliza en cientos de aplicaciones para identificar y autenticar usuarios con cámaras web, buscar caras coincidentes en bases de datos de fotos, detectar automáticamente las características faciales en editores gráficos y detectar caras en imágenes fijas y secuencias de video en tiempo real. FaceSDK se ha utilizado para crear sistemas seguros de control de identificación, vigilancia y asistencia (Luxand, 2010).

Es una excelente herramienta pero tiene un costo para su implementación (Luxand, 2010).

2.11. Base de datos MYSQL

MySQL es un sistema de administración de base de datos relacionales (SGBDR) robusto, rápido y fácil de usar que se adapta a casi la mayoría de lenguajes de programación como PHP entre otros, además es compatible con el servidor web apache y paginas dinámicas. MySQL es open Source que se adapta bien a la administración de datos en un entorno de red especialmente en arquitecturas cliente – servidor (Thibaud, 2006).

2.11.1. Orígenes de MySQL

MySQL fue diseñada por la empresa TCX debido a que necesitaban una base de datos que sea muy rápida para procesar mucha información en esos tiempos como no encontraron una que cumpla este requisito decidieron crear la base de datos MySQL. Actualmente es esta dirigido por la empresa sueca MySQL AB.

Respecto al origen del nombre no se está muy seguro ni por la empresa que la desarrollo, podría ser porque el nombre de la hija de uno de los principales desarrolladores (Monty) que se llama “My” (Thibaud, 2006).

2.11.2. Lenguaje de programación PHP.

Hypertext Preprocessor antes era personal home page, es un lenguaje interpretado del lado del servidor que surge dentro de la corriente denominada código abierto (open source). Se caracteriza por su potencia, versatilidad, robustez y modularidad. Al igual que ocurre con tecnologías similares, los programas son integrados directamente dentro del código HTML (Cobo, 2005).

2.12. MADAMDM

MANDAMDM (metodología ágil para el diseño de aplicaciones multimedia de dispositivos móviles) esta metodología es más cercana para dispositivos pequeños al igual que Mobile – D, consta de 5 etapas:

- Requerimientos
- Planificación
- Diseño
- Codificación
- Pruebas

Lo que hace interesante a esta metodología es que incluye tecnologías como XP (Programación Extrema), MOOMH (Metodologías Orientada a Objetos para la producción de software Multimedia e Hipermedia), UWE (Based Web Engineering) y OOHDM (Método de Diseño Hipermedia Objeto Orientado), tomando las características más importantes para concretar los objetivos y así facilitar la labor del desarrollo y diseño de las aplicaciones móviles, en los diferente sistemas operativos (FUZI, 2013).



Figura 15: Metodología ágil para el diseño de aplicaciones multimedia-dispositivos móviles
Fuente: elaboración propia

2.12.1. Conclusión de las Metodologías Analizadas

Para aclarar solo se propone metodologías ágiles debido al corto tiempo del taller de tesis, al analizar cada una de ellas se concluye que todas cumplen los requisitos y son eficaces para culminar un proyecto en corto tiempo además se identifican los métodos ágiles como la solución potencial para el desarrollo de software en móviles y debido a que cada una tiene características propias se opta por la metodología MANDAMDM (metodología ágil para el diseño de aplicaciones multimedia de dispositivos móviles) porque esta metodología extrae las características más relevantes que ayudan a concretar los objetivos y a facilitar el trabajo de desarrollo y diseño de la aplicación además de contar con la documentación bien elaborada hecha por (FUZI, 2013).

2.12.2. Estructura de la Metodología MANDAMDM

Como se observa en la figura de arriba existen 5 tipos de colores las cuales son resumidas por FUZI en la siguiente imagen:

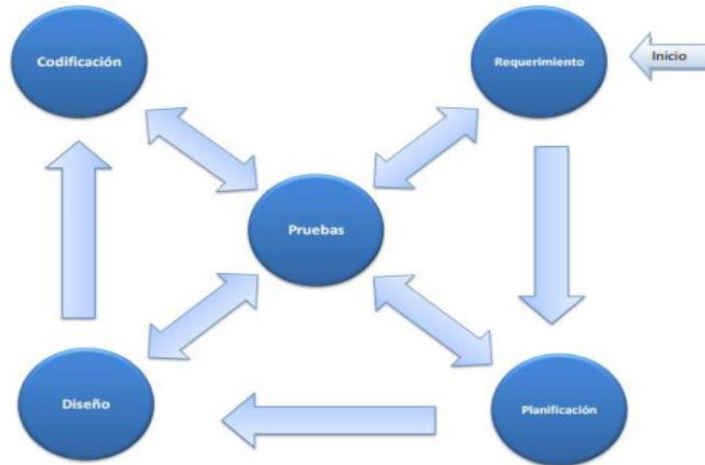


Figura 16: Estructura MANDAMDM

Fuente: (FUZI, 2013).

A. Fase de Requerimientos

Es la primera fase donde se definen las características de lo que el usuario quiere para el desarrollo de la aplicación. En esta fase el desarrollador debe de hacer uso sus destrezas de análisis y objetividad para establecer cuáles serán las características principales que predominan en la aplicación móvil.

B. Fase de Planificación

En esta etapa se debe de encontrar el problema, una vez encontrado se interpreta para una posible solución. Seguidamente se definirá las ventajas y desventajas de la solución, seguidamente el desarrollador tiene que organizar y planificar las actividades.

C. Identificación de Roles o Eventos

En esta etapa el desarrollador empieza la organización de los eventos que tendrá el usuario con la aplicación, extrayendo y modelando los contenidos, definiendo las características que poseerá los terminales, en otras palabras, las limitaciones del diseño y codificación del software, y estos a su vez deben ser hechos en diagramas de actividades para satisfacer los tiempos de desarrollo de la aplicación y así cumplir con los objetivos.

D. Modelo de Iteración

Es un esquema basada en las relaciones que tienen las diferentes herramientas de un dispositivo móvil, unos ejemplos de estos son: La cámara, el flash, Wifi, también deben plasmarse las posibles interfaces que tendrá la aplicación y todo esto es en forma general. Se recomienda utilizar el diagrama de casos de uso ya que mediante la cual se plasmarán las relaciones que se tendrá los usuarios con la aplicación.

Actividades del Proyecto

Esta etapa es la más importante ya que el desarrollador debe de hacer las actividades y los resultados respectivamente, estas actividades deben ser reales.

E. Fase de Diseño

En esta fase el desarrollador empieza con el diseño de la aplicación móvil, teniendo en consideración las diferentes herramientas que ofrecen los sistemas operativos móviles para la estructura y colores que deben tener las aplicaciones.

a) Diseño Conceptual

Esta etapa construye los esquemas conceptuales que poseerá la aplicación constituido por los objetos del dominio y sus relaciones. El diseño está formado por clases, relaciones y subsistemas. En consecuencia, se usará diagramas de clase, donde se verán los objetos y atributos que puedan ser diferentes tipos.

b) Diseño Navegacional

En esta etapa se deberá precisar los esquemas y estructuras de las aplicaciones anticipadamente concientizadas de las limitaciones de la aplicación y las herramientas que ofrecen los Sistemas Operativos.

c) Diseño Interfaz Abstracta

En esta etapa, el desarrollador debe acoplar los conceptos anteriores y establecer una interfaz abstracta, a eso añadiendo lo necesario para la aplicación móvil como

podrían ser: Audio, Video, Imágenes, También se puede plasmar cómo será la aplicación haciendo unos bocetos.

F. Fase de Codificación

Esta fase es la etapa más extensa del proyecto, en donde el desarrollador debe tener bien en claro cuál será el tipo de aplicación móvil que quiere desarrollar ya sea una aplicación nativa, bajo ambiente web o híbrida, ya que de ello dependerá el código y la sintaxis.

G. Fase de Pruebas

Esta es la última fase de la metodología MADAMDM es posible saltarse desde cualquier fase a la fase de pruebas, con el fin de crear un filtro donde el desarrollador puede analizar y visualizar un resultado rápido del producto que es la aplicación móvil desarrollada.

a) Etapa de Ejecución

En esta etapa del desarrollador comienza la ejecución de la aplicación móvil ya sea del diseño, del modelo o diagrama, con la finalidad de validar que se ejecute de manera coherente entre las interfaces o módulos y ofrezca estabilidad para un resultado esperado.

b) Etapa de Resultados

En esta etapa se tendrá un control para validar el producto final ya realizado para así garantizar que será un producto final de calidad y así cubrir las expectativas del mercado y de los usuarios finales.

2.13. Gestión de proyectos PMBOK

PMBOK es una guía concebida para identificar conocimientos, procesos, habilidades, herramientas y técnicas fundamentales sobre la dirección de proyectos. Estos elementos son generalmente reconocidos como buenas prácticas, pues son aplicables

a la mayoría de los proyectos en la mayoría de las ocasiones. La aplicación de conocimientos, habilidades, herramientas y técnicas puede contribuir a aumentar las posibilidades de éxito de una amplia variedad de proyectos. Los conocimientos descritos en el PMBOK no siempre se deben aplicar de la misma manera en todos los proyectos, el cómo aplicarse es algo que determina la organización, empresa y/o dirección del proyecto en función de sus particularidades (Parra, Saroza, Martínez, & Bello, 2016).

Por la razón descrita líneas arriba se ha tomado como referencia los conocimientos y buenas prácticas para la gestión exitosa de proyectos reunidas por el PMI® (Project Management Institute) en el documento llamado PMBOK (Project Management Body of Knowledge) quinta edición.

2.13.1. Procesos de la Dirección del proyecto

Los procesos de PMBOK se agrupan en 5 grupos que se tomará en cuenta en el presente proyecto los cuales son:

- A. El grupo de procesos de iniciación: En este se encuentra los procesos que definen el proyecto.
- B. El grupo de procesos de planificación: Definen los procesos que establecen el alcance del proyecto, definir objetivos y acciones para alcanzar los objetivos.
- C. El grupo de procesos de ejecución: Se encuentra los procesos que se realizarán para completar el proyecto.
- D. El grupo de procesos de seguimiento y control: Se encuentra los procesos para dar seguimiento, analizar y regular el progreso del proyecto, además permitirá identificar áreas en las que el proyecto requiere cambios.
- E. El grupo de procesos de cierre: Se encuentra los procesos que permitirán finalizar las actividades a fin de cerrar formalmente el proyecto.

2.13.2. Áreas de conocimiento

La quinta edición de PMBOK cuenta con 10 áreas de conocimiento, con las cuales se gestionará el presente proyecto:

- A. Gestión de la integración del proyecto
- B. Gestión del alcance del proyecto
- C. Gestión del tiempo del proyecto
- D. Gestión de los costos del proyecto
- E. Gestión de la calidad del proyecto
- F. Gestión de los recursos humanos del proyecto
- G. Gestión de las comunicaciones del proyecto
- H. Gestión de los riesgos del proyecto
- I. Gestión de las adquisiciones del proyecto
- J. Gestión de los interesados del proyecto

CAPITULO 3: MATERIALES Y MÉTODOS

La propuesta tecnológica tiene como objetivo implantar un sistema de reconocimiento facial que ayude a los policías en la búsqueda de personas con delito de violación sexual, con este aplicativo se intenta resolver el problema descrito anteriormente.

Para la elección de materiales y métodos se tuvo que definir cada una de ellas que se describen a continuación, se plantearon diversas opciones para desarrollar el prototipo, y para tomar una mejor decisión se analizaron estos factores:

- ✓ Experiencia
- ✓ Facilidad de integración
- ✓ Documentación, material y comunidad disponible en la web
- ✓ Aspectos de calidad de software brindados por la tecnología. Es decir, si brinda rendimiento, disponibilidad, usabilidad, entre otros.

A partir de estos factores analizados y dependiendo del costo y limitaciones que el proyecto implica, se decidió que tecnología era la más conveniente.

3.1. Tecnología de reconocimiento Facial.

Se ha optado por la herramienta Face API que ofrece diversas funcionalidades entre las que se encuentran la detección de caras, verificación facial, identificación facial, búsqueda de caras similares y la agrupación facial.

La detección de caras encuentra las caras que se encuentren en una imagen y devuelve los rectángulos donde se encuentran dichas caras en la fotografía, así como algunas características de cada cara basadas en aprendizaje automático de los rasgos faciales. Las características que esta aplicación proporciona son el género, el sexo, la edad, la sonrisa el vello facial entre otros.

La herramienta de verificación facial se utiliza para calcular la probabilidad de que dos caras pertenezcan a la misma persona. Esta herramienta utiliza la anterior para detectar las dos caras y poder hacer la comparación, además Microsoft recomienda que para

utilizar esta funcionalidad solo haya una cara en las imágenes que se envíen para comparar.

La aplicación que se usa para identificar a personas por la cara funciona teniendo en cuenta que, la cara que esté intentando identificar, se encuentren registradas en una base de datos, la herramienta define como un grupo de personas que tiene que haber sido creado antes de intentar identificar a alguien y que se puede ir modificando con el tiempo. Cada grupo puede contener hasta 1000 personas distintas con las que comparar después, así mismo cada persona que pertenezca a un grupo puede tener varias caras registradas. Cuando un grupo se crea se debe entrenar para que en el futuro pueda identificar nuevas caras. Cuando se envía una cara para la identificación, si esta es positiva la herramienta devuelve un identificador de la persona a la cual pertenece esa cara.

La búsqueda de caras similares se centra en la detección de una cara y la obtención de caras de características similares en la base de datos de Microsoft. Esta herramienta puede funcionar con dos modos. El primero "matchPerson" devuelve una serie de caras similares después de aplicar un umbral de coincidencia usando la aplicación explicada anteriormente de verificación. El segundo modo "matchFace" ignora el umbral que se aplica mediante la utilización de la verificación facial y devuelve caras similares pero que pueden pertenecer a distintas personas. Para una misma cara de búsqueda el primer método encontrará menos o, como mucho, el mismo número de coincidencias que el segundo y a su vez el segundo incluirá entre sus respuestas el resultado del primer método. Por último, la herramienta de agrupación facial, para un grupo dado de caras desconocidas, se encarga de dividir dichas caras en conjuntos basados en la similitud facial. Puede darse el caso de que las caras pertenecientes a un mismo grupo creado como resultado de esa asociación pertenezcan a la misma persona (ggailey777, 2017). Usar esta herramienta permite contar con una caja negra para el reconocimiento facial. Tanto el entrenamiento como la consulta solo es, el envío de fotografías y la respuesta recibida. No se conoce qué algoritmos de machine Learning o redes neuronales utiliza

Microsoft Face API. Esto tiene algunas ventajas como la reducción de la complejidad del desarrollo del prototipo y la abstracción del reconocimiento de imágenes a solo un servicio que puede ser cambiado por otro en el futuro.

La librería OpenCV también realiza las mismas funciones que el API de Microsoft ya que este fue creado para proporcionar una plataforma común para aplicaciones de visión por computadora y para acelerar el desarrollo de las mismas, cuenta con más de 2500 algoritmos entre la cuales se puede encontrar para realizar detección de rostros, objetos, rastrear movimientos entre otros, es compatible con diversas plataformas Linux, Windows, Android.

Muchos trabajos de investigación como (Collantes Arana et al., 2013), (Rojas-Cala, 2017; G. Santana & Adrián), concluyeron que al ser implementado usa mucho recurso de Hardware y memoria de almacenamiento en dispositivos pequeños, es por esto que es necesario implementar sus algoritmos en otra plataforma como PHP u otros, usando métodos de comunicación como servidores web REST para hacer que funcione correctamente y realice la función que se pretende lograr de reconocimiento facial, además de otras desventajas como, no reconocen cuando el sujeto a ser reconocido usa gafas u otros objetos en el rostro, además la iluminación también influye bastante ya que si hay poca iluminación no realiza la función que se espera solo trabaja bien en ambientes controlados, otra de las desventajas es que los algoritmos principales que tienen más aceptación en reconocer una persona son Eigenface, Fisherfaces y LBPH (Rojas-Cala, 2017), los métodos que usan son arboles de regresión o vectores, ejemplo toman como patrones las cejas, nariz entre otros rasgos faciales, los cuales no son precisos cuando una persona envejece. Como el prototipo se pretende utilizar en un ambiente no controlado, no se opta la utilización de esta librería.

3.2. Infraestructura y uso de Cloud Computing

Dado que el prototipo debe contar con aplicaciones móviles y web que se comuniquen con una misma lógica de negocio, se consideró necesario la utilización de un servidor de

tecnologías de Cloud Computing. Este último aspecto sería importante para proveer al cliente la posibilidad de escalar el sistema en el futuro, pues estos servicios en la nube son ideales para aplicaciones escalables, muchos proveen auto-escalamiento, y en su mayoría se pueden utilizar como una caja negra sin conocer en profundidad la configuración e infraestructura.

Se decidió por Microsoft Azure ya que este integra a todos sus API de inteligencia artificial, también incluye la API que se utilizara en este proyecto, es fácil de usar además que ya se cuenta con experiencia en su uso el cual facilita para lograr cumplir con los objetivos del desarrollo del proyecto.

3.3. Lenguajes y tecnologías para la lógica de negocio o Back-End

Se pretende desarrollar un aplicativo que sea capaz de instalarse en móviles con sistema operativo Android a partir de la versión 6.0 y para no sobrecargar la memoria del móvil se tomara la foto luego será enviado a un servidor web alojado en la nube donde estará alojado el algoritmo de reconocimiento facial, este no debe ser almacenado en la memoria del celular, esta es la complejidad del desarrollo ya que no existe documentación suficiente para realizarlo con Android, pero si hay mucha documentación y proyectos ya desarrollados instalados en cámaras de seguridad inteligentes o en computadoras de alta gama, al ser mi proyecto desarrollo de una aplicación móvil se hará uso del IDE Android Studio 3.1.4 con lenguaje de programación java, y para comunicar el AP con el cloud computing donde estará la API lo ideal para mejorar la modificabilidad e interoperabilidad de los módulos de este prototipo sería crear una aplicación web que consuma la API en formato JSON mediante PHP.

Al ser un prototipo se necesita otro programa para subir las fotografías, debido a que una persona para ser sentenciado o tener antecedentes penales por violación primero tiene que pasar por muchos procesos de investigación por parte del poder judicial y policía nacional, es por eso que no se integra a la aplicación el módulo de entrenar al sistema,

para esto se creara otro sistema en C# con visual Basic que servirá para entrenar a la API.

3.4. Base de datos.

Para la base de datos se decidió por MySQL debido a los requerimientos del prototipo y que ya se tiene experiencia en el uso de este gestor de base de datos el cual se instalara para las pruebas en una laptop Core i3 marca HP por medio de Xampp esta herramienta incluye PHP, servidor apache. Además, es libre como se describe en el marco teórico.

3.5. Aplicación móvil

Se planteó 2 opciones para desarrollar la aplicación, considerando que se pretende programar en código nativo usando java, estos son en Xamarin form o Android Studio para tomar una decisión se consideró el siguiente cuadro:

Tabla 2: Comparativo de ID's para desarrollo de aplicación móvil.

características	Android Studio	Xamarin Form
Lenguajes utilizados	Java	C#
Experiencia en los lenguajes usados	Si	Media
Multiplataforma	No	Si
Se genera código nativo	Si	Si
Rendimiento y usabilidad	Al ser nativo se tiene una buena performance y usabilidad	Baja ya que se requiere de una máquina de buen rendimiento(recomiendan usar con procesador core i5 en adelante)
Documentación y material disponible	Documentación extensa, buen material y comunidad online	Documentación extensa pero algunos con costo.
Facilidad de capacitación	Buena ya que solo se tendrá que capacitar en Android	No se tiene experiencia en el lenguaje, es necesario capacitarse en Xamarin.

Fuente: Elaboración propia, utilizando Microsoft Excel 2013.

Al analizar el cuadro se concluyó en usar Android Studio dado que aportaba ventajas de performance y usabilidad con respecto a Xamarin Form.

3.6. Dirección del proyecto según las áreas del conocimiento de la guía del PMBOK.

Para el desarrollo de esta tesis se utilizó esta guía PMBOK con el fin de alcanzar los objetivos trazados, además de ser una guía utilizada para estos tipos de trabajos o proyectos, en el marco teórico se define detalladamente.

3.6.1. Gestión de la integración del proyecto

En esta etapa se incluyeron los procesos y las actividades necesarias para el buen desarrollo de la presente tesis de investigación tecnológica.

Tabla 3. Actividades para el desarrollo de la investigación.

PROYECTO		“APLICACIÓN MOVIL DE RECONOCIMIENTO FACIAL EN PERSONAS CON ANTECEDENTES DE ABUSO SEXUAL EN LA PROVINCIA DE ANDAHUAYLAS, APURÌMAC – 2018”	
PATROCINADOR	Jefe del Proyecto: Ervin Lewis Caceres Mariño		
PREPARADO POR	Jefe del Proyecto: Ervin Lewis Caceres Mariño	FECHA:	20/07/2018
REVISADO POR	Jefe del Proyecto: Ervin Lewis Caceres Mariño	FECHA:	22/07/2018
APROBADO POR	Jefe del Proyecto: Ervin Lewis Caceres Mariño	FECHA:	23/07/2018
REVISION	DESCRIPCION		FECHA
1	Preparación de acta de constitución (ERVIN LEWIS CACERES MARIÑO)		25/07/2018
BREVE DESCRIPCION DE LA APLICACIÓN MOVIL DEL PROYECTO			
La aplicación móvil desarrollada para la plataforma Android, solucionara el gran problema que tienen los policías al momento de realizar operativos en las fronteras de las provincias, departamentos, ciudades con el fin de arrestar a personas que tienes requisitorias por abuso sexual contra la mujer, con una simple toma de fotografía con el aplicativo, los policías obtendrán un reporte completo de la persona para lo cual el aplicativo sacara un Fingerprint del rostro de la persona y lo compara con la Base de datos de la policía.			
ALINEAMIENTO DEL PROYECTO			
OBJETIVOS ESTRATÉGICOS DE LA ORGANIZACIÓN		PROPÓSITO DEL PROYECTO	
Dar talleres y capacitación sobre el uso del aplicativo en la comisaría provincial de Andahuaylas		Promover el funcionamiento y la utilización de la aplicación.	
OBJETIVOS DEL PROYECTO			
Realizar pruebas de eficacia, tiempo de respuesta a los algoritmo pre entrenados para detección de rostros.			
Desarrollar un prototipo de reconocimiento facial para personas con antecedentes de abuso sexual.			
Implementar un servidor web para alojar el sistema de reconocimiento facial.			

FACTORES CRÍTICOS DE ÉXITO DEL PROYECTO	
Disposición de las herramientas para el desarrollo de la aplicación móvil dentro de los plazos establecidos.	
Diseño de la aplicación móvil acorde a los requerimientos de los interesados.	
REQUERIMIENTOS DE ALTO NIVEL	
La aplicación debe ser desarrollada exactamente de acuerdo a los requerimientos funcionales planteados.	
La aplicación debe ser desarrollado exactamente para plataforma Android.	
EXTENSIÓN Y ALCANCE DEL PROYECTO	
FASES DEL PROYECTO	PRINCIPALES ACTIVIDADES
FACE I: ANALISIS	Acta de constitución del proyecto.
	Análisis de la información recolectada.
	Análisis de los requerimientos funcionales.
	Análisis de los requerimientos no funciones.
	Diagrama de casos de uso.
	Diagrama de secuencia.
FACE II: DISEÑO	Diseño de la base de datos.
	Diagrama de clases.
	Diseño navegaciones
	Definición de las interfaces abstractas
FACE III: CIDIFICACION	Adquisición de la laptop.
	Adquisición de dispositivo móvil.
	Adquisición de las herramientas de desarrollo.
	Codificación.
FASE IV: PRUEBAS	Pruebas duran el desarrollo
	Pruebas unitarias
	Pruebas integradas
INTERESADOS CON EL PROYECTO	
Policía nacional de Andahuaylas	
Población de la provincia de Andahuaylas	
Programador o desarrollador	
RIESGOS	

Demora en la adquisición de la laptop, dispositivo móvil con Android 6.0 y las herramientas de desarrollo
Poca información por parte de la policía sede Andahuaylas.
La disponibilidad de recursos económicos en el tiempo establecido.
Retraso en la programación
HITOS PRINCIPALES DEL PROYECTO
Aprobación del acta de constitución del proyecto
Alcanzar todos los objetivos trazados
Pruebas y correcciones de la aplicación.
PRESUPUESTO DEL PROYECTO
El costo del proyecto que asciende a la suma de S/. 10, 505.25 será asumido en un 100% por el jefe del proyecto.

Fuente: Elaboración propia, utilizando Microsoft Excel 2013.

3.6.2. Gestión del alcance del proyecto

Sobre la base de resolver el problema planteado en la sección anterior, surge el planteo de un prototipo funcional que implica el desarrollo e implementación de una aplicación móvil. La misma permite a un policía realizar consultas si una persona cuenta con antecedentes penales por violación sexual, utilizando para ello un modelo de reconocimiento de patrones del rostro de cada persona, almacenada su fotografía en la base de datos de la policía.

Dicho modelo se trata de un sistema que a través de algoritmos de machine Learning aprende a reconocer rostros, lo primero que realizara es sacar el Fingerprint de cada rostro para luego compararlo con rostros almacenados en una base de datos. Luego devolverá un reporte si la persona reconocida se encuentra en la base de datos, para esto primero tendrá que encontrar el rostro de una persona en una fotografía si es que lo hay, buscara en la base de datos.

La aplicación a desarrollar permitirá a los policías realizar su labor de buscar personas con requisitoria mucho más rápidas y fáciles.

3.6.3. Gestión del tiempo del proyecto

“La gestión del tiempo para proyectos de software está manejada por el riesgo, la disponibilidad de recursos, el valor de negocio, y los métodos de planificación utilizados, la mayor parte del costo de desarrollo de un proyecto es el esfuerzo humano, y el esfuerzo es el producto de las personas y el tiempo” (Cárdenas Taype, 2017).

Esta sección tiene por objetivo detallar los procesos necesarios para lograr la conclusión del proyecto en tiempo y forma. En este caso fue fundamental crear un cronograma de trabajo, Dicho cronograma sirvió para controlar la ejecución del proyecto, más en este caso que se tiene poco tiempo de trabajo.

A. Definición de la duración de las actividades.

La duración de las actividades se definió utilizando la técnica de juicio de expertos. El juicio de expertos se realizó en una reunión, así mismo se utilizó las plantillas de Project a fin esquematizar las secuencias y duración de las actividades.

B. Desarrollo del cronograma

Para el desarrollo del cronograma de las actividades se analizó el orden, duración, recursos y restricciones. La creación del cronograma se realizó usando el software Project, este utiliza el método de La ruta crítica, calculando las fechas teóricas de inicio y finalización.

C. Control del cronograma

El proceso de control de cronograma, se realizó en cada actividad estrictamente utilizando la técnica de revisión del desempeño, semanal y mensualmente para cada una de las actividades del cronograma, esto permitió saber si el desarrollo estaba adelantado o atrasado respecto a la línea base del cronograma.

D. Roles

Posterior a la definición de tareas y cronograma se procedió a definir los roles necesarios para ejecutar las mismas. En el cuadro 3 se describen las principales tareas y roles.

Tabla 4. Roles y responsabilidades de la investigación

Rol	Responsabilidades
Arquitecto	Diseñar las estructuras necesarias para cumplir con los atributos de calidad especificados en los requerimientos. Definir los elementos de software, relaciones entre los mismos y propiedades de éstos
Desarrollador	Codificar los requerimientos establecidos por el ingeniero de requerimientos cumpliendo a la vez con las decisiones de arquitectura definidas
Documentador	Documentar las justificaciones de decisiones, definiciones y resultados obtenidos en el curso del proyecto
Gerente	Definir y asignar recursos al proyecto. Dar soporte a las tareas de estimación y definición de las actividades contenidas en los planes y realizar la revisión y aprobación de los mismos
Gestor de la calidad	Elaborar y hacer el seguimiento del plan de SQA definiendo los requisitos y estándares necesarios para asegurar la calidad del producto y del proyecto
Gestor de la comunicación	Elaborar un plan que exprese las vías de comunicación entre los miembros del equipo y sus interesados

Gestor de la configuración	Elaborar y hacer el seguimiento del plan de SCM definiendo los elementos de configuración necesarios para asegurar el correcto desarrollo del proyecto
Gestor de riesgos	Elaborar y hacer el seguimiento de un plan que identifique, evalúe y gestione los riesgos que pueden impactar en el proyecto
Ingeniero de Requerimientos	Recolectar, analizar y documentar las funcionalidades y restricciones que debe cumplir el sistema
Tester	Diseñar, ejecutar y registrar resultados de las pruebas del desarrollo de software

Fuente: Elaboración propia, utilizando Microsoft Excel 2013.

Para dejar en claro que todos los roles lo asumieron el gerente del proyecto, se detalla para dar una idea más clara sobre lo que conlleva desarrollar una aplicación móvil.

3.6.4. Gestión de los costos del proyecto

Estimación de Costos: La estimación de costos se realizó utilizando la técnica de juicio experto y cotización en tiendas de la zona. Se realizó el análisis de precios unitarios, de cada una de las actividades que se desglosan en los siguientes rubros:

- ✓ Bienes
- ✓ Servicios
- ✓ Software
- ✓ Recursos humanos

A. Determinación del presupuesto

El presupuesto se determinó de acuerdo a la estimación de costos anteriormente mencionado.

B. Control de los Costos

Los costos del proyecto no sufrieron mayores variaciones positivas o negativas debido a que los costos de los equipos y materiales fueron estables durante el avance del proyecto.

3.6.5. Gestión de la calidad del proyecto

La gestión de la calidad del proyecto incluye los procesos y actividades de la organización ejecutante que determinan responsabilidades, objetivos y políticas de calidad a fin de que el proyecto satisfaga las necesidades por la cuales fue emprendido. Implementa el sistema de gestión de calidad por medio de políticas y procedimientos, con actividades de mejora continua de los procesos llevados a cabo durante todo el proyecto, según corresponda.

Tabla 5. Gestión de calidad del proyecto

ÍTEM	FASE	ENTREGABLES CLAVES	METRICAS UTILIZADAS
Actividad 1	análisis	Análisis de la información recolectada.	Internas
		Análisis de los requerimientos funcionales	
		Análisis de requerimientos no funcionales	
		Diagrama de casos de uso	
		Diagrama de secuencia	
Actividad 2	Diseño	Diseño de la base de datos	Internas
		Diagrama de clases	
		Diseño navegacional	
		Diseño de las interfaces abstractas.	
Actividad 3	Codificación	Funcionamiento: logueo	Internas y externas
		Funcionamiento: modulo activar cámara	
		Funcionamiento: modulo detectar rostro	

		Funcionamiento: Modulo reconocer persona.	
		Funcionamiento: modulo reporte de persona reconocida	
Actividad 4	Pruebas	Resultados de pruebas durante el desarrollo	Internas y externas
		Resultado de pruebas unitarias	
		Resultados de pruebas integradas	

Fuente: Elaboración propia, utilizando Microsoft Excel 2013.

Tabla 6. Descripción de calidad de proyecto

CARACTERISTICAS	SUBCARACTERISTICAS
Usabilidad	Facilidad.
	atractivo
Funcionalidad	Exactitud
	seguridad
Transportabilidad	Instabilidad
	Compatibilidad

Fuente: Elaboración propia, utilizando Microsoft Excel 2013.

3.6.6. Gestión de los Riesgos

La gestión de riesgos del software tiene como objetivo mejorar la probabilidad de lograr los objetivos del proyecto; la gestión de oportunidades de software tiene como objetivo superar los objetivos del proyecto. La gestión de oportunidades es una práctica común en la gestión de proyectos de software, especialmente en proyectos de adaptación que tienen la oportunidad de responder rápidamente a los cambios solicitados por el cliente, aplicar nuevas tecnologías, o aceptar recursos adicionales. El proceso de gestión de riesgos es "un proceso continuo para la identificación sistemática, análisis, tratamiento y seguimiento del riesgo a lo largo del ciclo de vida de un producto o servicio" (Cárdenas Taype, 2017).

Tabla 7. Plan de gestión de riesgos

PLAN DE GESTIÓN DE RIESGOS

objetivos de la Gestión de Riesgos del Proyecto de Software:
Los objetivos de la gestión de riesgo son identificar, controlar y eliminar las fuentes de riesgo antes de que empiecen a afectar el cumplimiento de los objetivos del proyecto.
Por otra parte, identificar, evaluar la probabilidad de que ocurra, estimar el impacto y establecer un plan de contingencia en caso de que el problema se presente.
Alcances
La identificación, análisis, priorización y seguimiento de riesgos más críticos será realizado por el jefe de Proyectos asignado.
El proceso de Gestión de Riesgo del software debe ser definido e implantado por los involucrados en el proceso de desarrollo del software.
Herramientas
Opinión de la alta gerencia
Tormenta de ideas
Juicio de expertos
Check list riesgos potenciales
Análisis de los supuestos identificados
Fuentes de Datos
La identificación de todos los riesgos fue por parte de todos los involucrados en el proceso de desarrollo del software, según su experiencia y juicio.
Estrategias de riesgo reactivas y proactivas
Reactivo
El equipo de proyecto no hace nada hasta que sucede algo mal.
Tratar de remediar el problema al apuro (Aparece la Gestión de Crisis para tomar el control, y es cuando el proyecto está en verdadero peligro).
ser proactivo:
Inicia antes del trabajo técnico.
Identificar los riesgos potenciales, valorando su probabilidad e impacto.
Se clasifican según su importancia.
Establecer un plan.
Los riesgos involucran dos características:
Incertidumbre: Se desconoce si puede suceder.
Pérdida: El riesgo se convierte en realidad.
Tipos de riesgos que se encuentran:
Riesgos del Proyecto (Amenazan el plan del proyecto).

Riesgos Técnicos (Amenaza la calidad y actualidad del software que se producirá).		
Riesgos de Negocios (Amenaza la viabilidad del software que se construirá).		
POSIBLES RIESGOS DEL PROYECTO DE SOFTWARE GENÉRICOS		
RIESGO	TIPO	DESCRIPCIÓN
rotación del personal	proyecto	Personal con experiencia abandona el proyecto sin antes terminar o ser entregado.
Cambio de gestión	proyecto	Habrà un cambio de gestión organizacional con diferentes prioridades.
Cambio de requerimientos	Proyecto y producto	Habrà más cambios en los requerimientos de lo esperado.
Retrasos en la especificación	Proyecto y producto	Las especificaciones de las interfaces esenciales no estarán a tiempo.
Subestimación del tamaño	Proyecto y producto	El tamaño del proyecto se ha subestimado
Cambio de tecnología	Negocio	Un producto competitivo se pone en venta antes de que el sistema se complete.
Competencia del producto	Negocio	La tecnología fundamental sobre la que se construirà el sistema se sustituye por nueva tecnología.
Etapas del Proceso de la Gestión del Riesgos:		
Identificación de riesgos: Identificar los posibles riesgos para el proyecto, el producto y los negocios.		
Análisis de riesgos: Valorar las probabilidades y consecuencias de estos riesgos.		
Planeación de riesgos: Crear planes para abordar los riesgos, ya sea para evitarlos y minimizar sus efectos en el proyecto.		
Supervisión de riesgos: Valorar los riesgos de forma constante y revisar los planes para la mitigación de riesgos tan pronto como la información de los riesgos esté disponible.		

Fuente: Elaboración propia, utilizando Microsoft Excel 2013.

3.6.7. Gestión de los Interesados

Los interesados con el proyecto son:

- ✓ Policía nacional del Perú.
- ✓ Mujeres que realizaron su denuncia a sujetos que abusaron de ellas.
- ✓ Población en general de la provincia de Andahuaylas.

A. Planificación de los interesados

Se realizó constante comunicación, reunión y trabajo en conjunto con los miembros de investigación de la policía, producto de esto el resultado se obtuvo una aplicación móvil de calidad que cumple con los objetivos del proyecto.

3.7. Puesta en marcha de la metodología MADAMDM.

3.7.1. Fase de Requerimientos

En esta fase el cliente cumple un rol importante donde dará a conocer las características necesarias para el desarrollo de la aplicación, en consecuencia, pretender satisfacer las necesidades del mercado y de los usuarios.

Esta tarea es responsabilidad del cliente debido a que se prioriza en orden del valor que aporta el requerimiento a la solución final teniendo en cuenta el contexto de las limitaciones del proyecto. Con esto el equipo se refiere a que pueden existir requerimientos catalogados con una prioridad baja y que en un futuro puedan dar gran valor al producto, pero, al tratarse de un proyecto a corto plazo, tal vez no sea posible desarrollarlos dentro de este prototipo funcional de testeo. Para entender mejor este punto se puede considerar que para la aplicación móvil pueden surgir varias funcionalidades extras interesantes, pero dada la limitante de tiempo solo se van a abarcar las principales funcionalidades de la misma, lo que va a indicar si es viable seguir con el proyecto o no.

3.7.1.1. Requerimientos funcionales.

Tabla 8. Requerimientos funcionales

ID	REQUERIMIENTOS FUNCIONALES
R1	Se debe crear una base de datos.
R2	La DB debe permitir almacenar datos de los sujetos con requisitoria por violación sexual
R3	La BD debe permitir almacenar fotografías de los sujetos. con requisitoria por violación sexual
R4	Debe permitir cargar la fotografía a la BD para la búsqueda de la persona.
R5	Debe detectar rostros dentro de la imagen.
R6	Debe permitir comparar con la base de datos de imágenes de personas con antecedente por violación sexual.
R7	Debe permitir reconocer a la persona en tiempo real.
R8	Al ser reconocido debe permitir al efectivo policial actualizar el estado de la persona a “capturado” en tiempo real sobre la BD.
R9	Debe permitir visualizar el reporte si cuenta con requisitoria o no.
R10	Debe borrar automáticamente la fotografía de la memoria para no sobrecargarlo del dispositivo.

Fuente: Elaboración propia, utilizando Microsoft Excel 2013.

3.7.1.2. Requerimientos no funcionales

Tabla 9. Requerimientos no funcionales

ID	REQUERIMIENTOS NO FUNCIONALES
R1	La aplicación se instalará solamente en teléfonos Smartphone
R2	La aplicación se instalará solamente en plataforma o sistema operativo Android.
R3	La aplicación solamente ejecutará desde la versión Android 4.0 en adelante.
R4	El envío de mensajes de textos de la aplicación debe ser de forma segura.
R5	La recepción de mensajes de textos enviados de la aplicación será en cualquier teléfono móvil (no es necesario teléfono Smartphone)

Fuente: Elaboración propia, utilizando Microsoft Excel 2013.

3.7.2. Fase de Planificación

En esta fase se realizar pruebas y analizaran los resultados de eficacia de los algoritmos de reconocimiento facial, Los métodos que se encuentran en OpenCV basados en características tienen un problema cuando se trata de detectar características faciales cuando estas no se encuentran en condiciones normales, es decir, los ojos cerrados, la boca abierta, o con el uso de accesorios. Los métodos basados en imágenes 2D demuestran que son sensibles principalmente a los cambios en la iluminación y la postura. Bajo condiciones controladas, la mayoría de los métodos basados en la imagen 2D alcanzan una precisión superior al 90% pero caen cuando se introducen las condiciones reales.

El método de Cascada se ha vuelto una base natural para comenzar el estudio de reconocimiento facial, una vez que se conocen e identifican las deficiencias de este modelo puede pasarse a uno más complejo o buscar las soluciones a sus desventajas (M. A. G. Santana, Díaz-Sánchez, Paz, & Huertas).

Al observar las pruebas analizadas y definidas por (M. A. G. Santana et al.) No se opta para realizar el prototipo de reconocimiento de personas con antecedentes por violación sexual ya que se trata de un tema delicado por tratarse de la seguridad.

Para plasmar lo mencionado anteriormente se harán en diagramas de actividades, Diagrama de casos de uso, y se usará el software StarUML 3.0.2

3.7.2.1. Diagramas de casos de uso

Los casos de uso de la aplicación, son los que definen los requerimientos de los usuarios y representan justamente la funcionalidad de la aplicación cuando el actor lleva un proceso una actividad (Morccolla Ancco, 2017).

A. Caso de uso general

Los casos de uso de la aplicación en forma general han sido catalogados de acuerdo a los roles que cumple la aplicación.

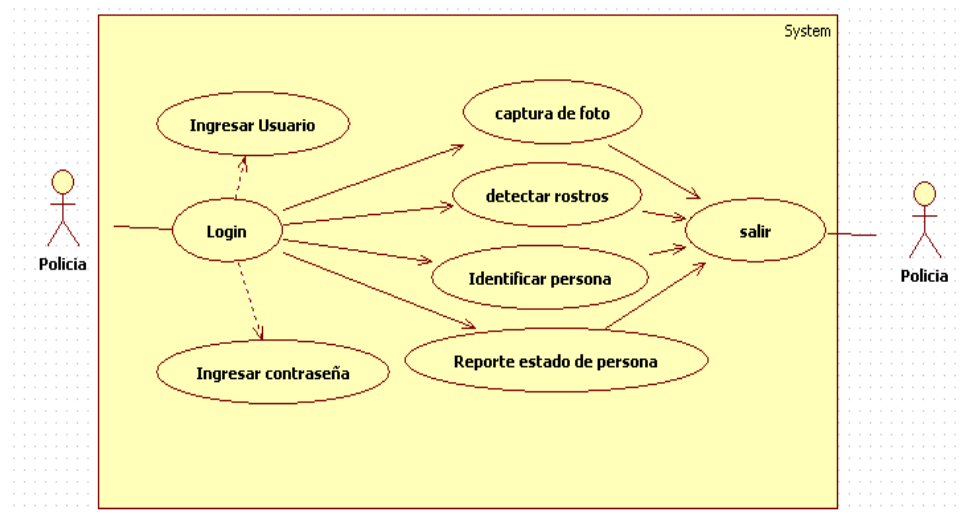


Figura 17. Catálogos de acuerdo a sus roles

Fuente: software StarUML 3.0.2

B. Caso de uso captura de foto.

En este se encuentran el caso de uso necesario para capturar mediante una captura de imagen del rostro de la persona.

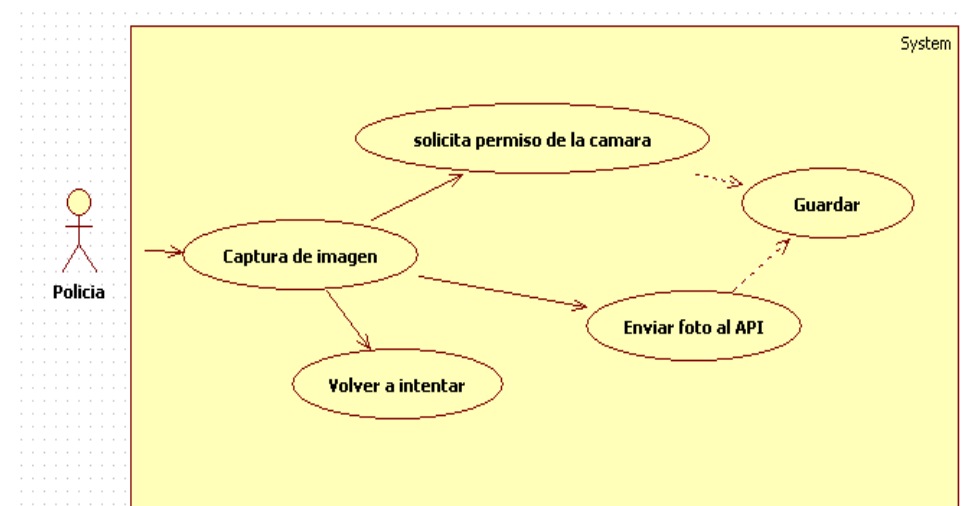


Figura 18. Catálogos de captura de rostro

Fuente: software StarUML 3.0.2

C. Caso de uso detectar rostro de personas.

En este se encuentran los casos de uso necesarios para el envío de la fotografía al API y este valide si hay rostros dentro de la fotografía, la respuesta del API es en formato JSON.

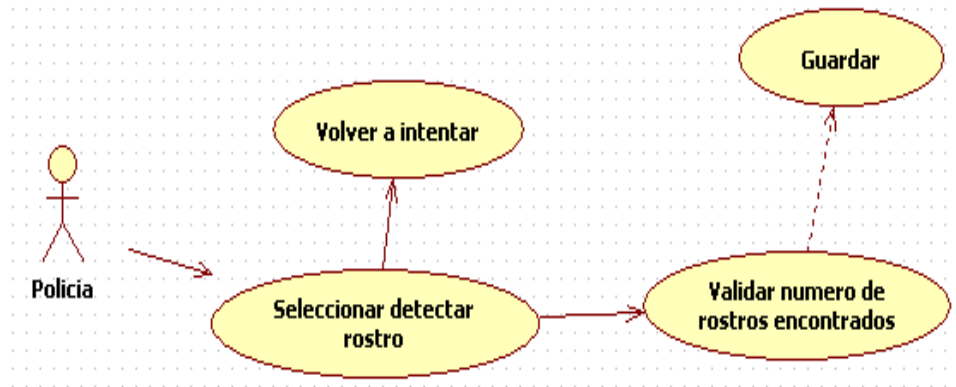


Figura 19. Catálogos de detectar rostro de personas

Fuente: software StarUML 3.0.2

D. Caso de uso identificación de la persona.

En este se encuentran los casos de uso necesarios para la identificación de la persona.

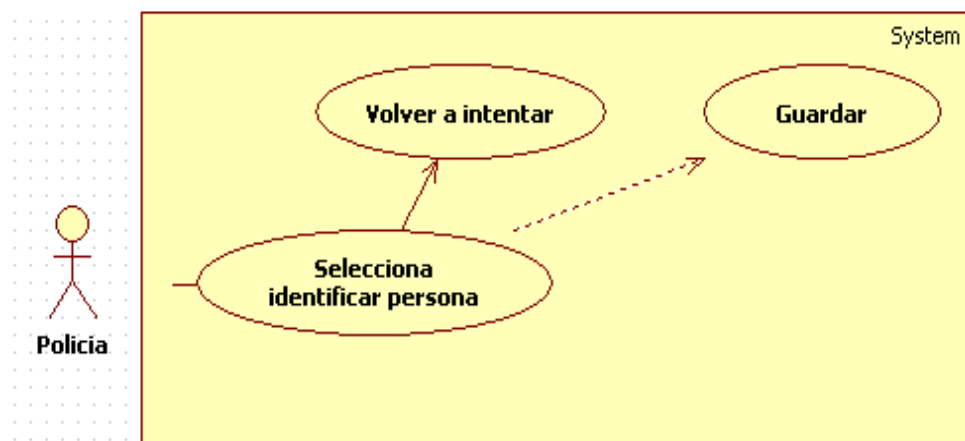


Figura 20. Catálogos de identificación de persona

Fuente: software StarUML 3.0.2

E. Caso de uso reporte de estado de persona.

En este se encuentran los casos de uso necesarios para visualizar el reporte de la persona registrada con el aplicativo.

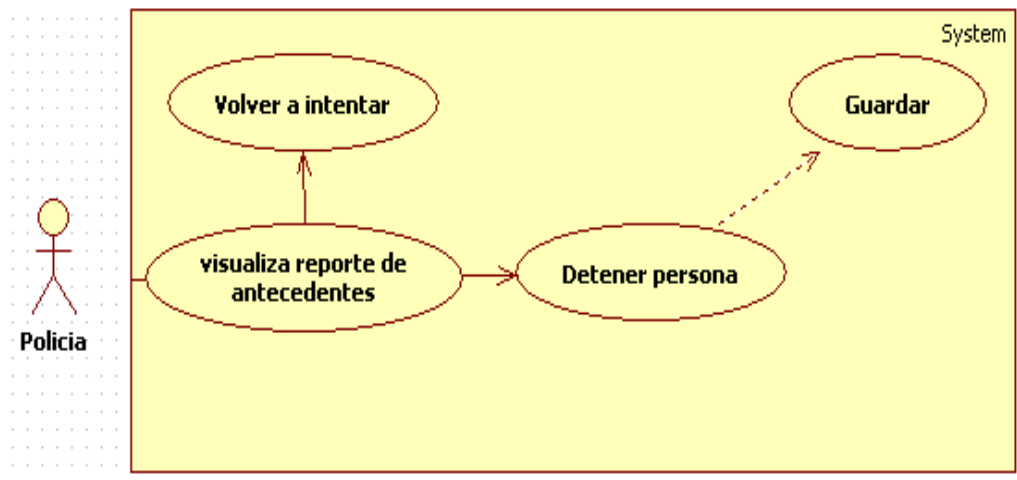


Figura 21. Catálogos de reporte de estado de persona

Fuente: software StarUML 3.0.2

F. Especificación de casos de usos

Para un mejor entendimiento se realiza la especificación de caso de uso que es la descripción de las partes definidas con el fin de detallar la información completa. La especificación se realizó bajo un cuadro que muestra las partes y las indicaciones básicas para que sea más sencilla y fácil de escribir y leer.

Tabla 10. Especificaciones de captura de foto

CASO DE USO:	CAPTURA DE FOTO
Tipo:	Primario
Propósito:	Capturar la foto para luego ser comprimido en un formato que pueda ser reconocido por la API para su envío.
Precondición:	Autenticación ha sido realizado previamente. Aceptar uso de la cámara ha sido realizado previamente.
Final exitoso:	Cámara activa se muestra en pantalla y foto para su envío.
Actores:	Policía
Evento inicio:	El policía inicia la cámara o selecciona la imagen.

Flujo principal	El policía selecciona capturar imagen
	El policía selecciona en aceptar los términos del uso de la cámara Android.
	El policía acepta la imagen y lo envía.

Fuente: elaboración propia

En la siguiente tabla se detalla textualmente la especificación de caso de uso “**Detectar rostro de personas**”.

Tabla 11. Especificaciones de detector de rostro

CASO DE USO: DETECTAR ROSTRO DE PERSONAS	
Tipo:	Primario
Propósito:	Detectar si en la fotografía existe algún rostro humano
Precondiciones:	Capturar fotografía ha sido realizado previamente.
Final exitoso:	El api responde en formato JSON si hay rostros y el número de rostros humanos.
Final Fallido:	No se detectaron rostros humanos o personas.
Actores:	Policía
Evento inicio:	El policía debe tener en pantalla la fotografía capturada.
Flujo principal:	El policía selecciona detectar persona.
	El policía pide al sistema realizar la consulta si existe algún rostro dentro de la fotografía.
	El policía pide al sistema cuantos rostros fueron detectados en la fotografía.

Fuente: elaboración propia

En la siguiente tabla se detalla textualmente la especificación de caso de uso “**Identificación de la persona**”.

Tabla 12. Especificaciones de identificación de la persona

CASO DE USO: IDENTIFICACION DE LA PERSONA	
Tipo:	Primario

Propósito:	Comparar la fotografía con personas de la base de datos de la policía. Y reconocer a quien pertenece el rostro.
Precondición:	Autenticación ha sido realizado previamente. El aplicativo haya reconocido si hay rostros en la fotografía.
Final exitoso:	La imagen haya sido comparada satisfactoriamente mediante el Fingerprint. Y se haya reconocido a la persona
Actores:	Policía
Evento inicio:	Procesar la imagen.
Flujo principal	El policía selecciona identificar persona.
	El policía pide al sistema si se logró reconocer a la persona

Fuente: elaboración propia

En la siguiente tabla se detalla textualmente la Especificación de caso de uso “Reporte estado de persona”.

Tabla 13. Especificaciones de reporte de persona

CASO DE USO:	REPORTE ESTADO DE PERSONA
Tipo:	Primario
Propósito:	Reconocer a la persona, y mostrar los motivos por el cual presenta requisitoria por parte de las autoridades y posterior a su detención.
Precondición:	Autenticación ha sido realizado previamente. El aplicativo haya reconocido a quien pertenecería el rostro.
Final exitoso:	Detención de la persona por los delitos cometidos de violación sexual
Final fallido:	No se obtuvo los datos de la persona a identificar por el API

Actores:	Policía
Evento inicio:	Procesar la imagen y comprar con la base de datos.
Flujo principal	El policía visualiza el reporte.
	El policía selecciona capturar requisitoriado por violación sexual contra la mujer.

Fuente: elaboración propia

3.7.2.2. Diagrama de actividades.

Los diagramas de actividades muestran la serie de actividades que se deben realizar en un caso de uso. Es decir, muestra el flujo de actividad desde el punto de inicio hasta el punto final mostrando las rutas de decisiones que existen en el progreso de eventos que contiene cada actividad.

En la siguiente figura vemos Diagrama de actividad "Login" se muestra la secuencia de actividades para acceder a la ventana principal de opciones de la aplicación.

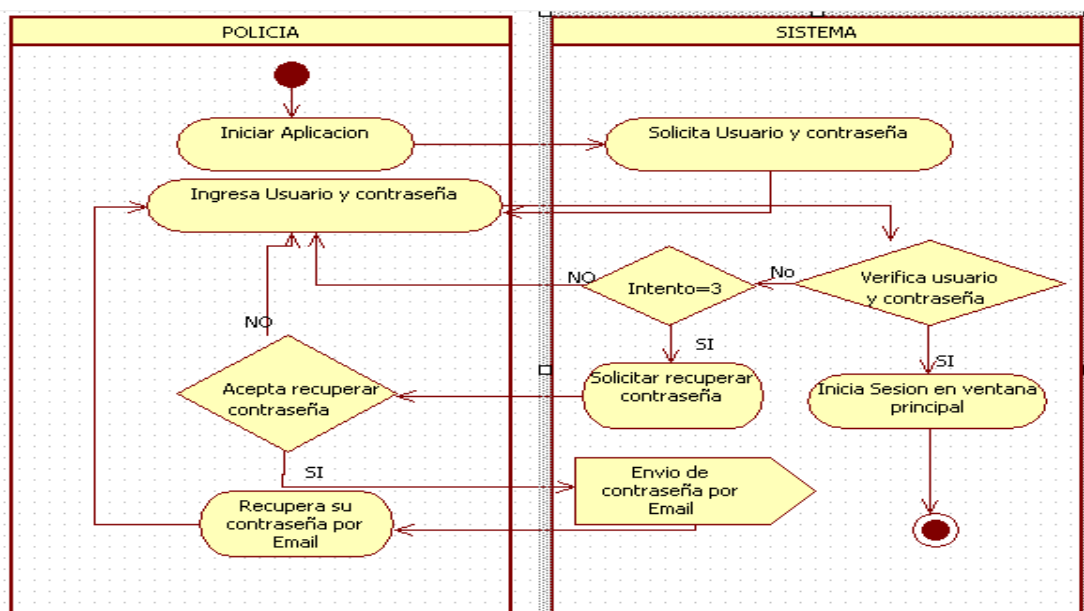


Figura 22. Diagrama de actividades

Fuente: Elaboración propia

En la siguiente figura vemos **Diagrama de actividad “Captura de fotos”** se muestra la secuencia de actividades para capturar la fotografía adecuada para luego enviar al API.

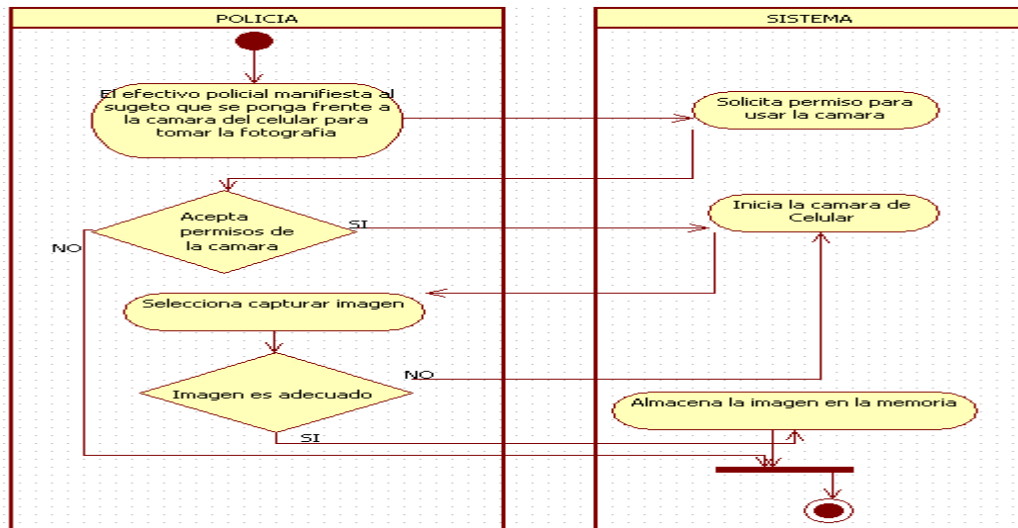


Figura 23. Diagrama de actividades – captura de fotos

Fuente: Elaboración propia

En la siguiente figura vemos **Diagrama de actividad “Detectar rostro”** se muestra la secuencia de actividades para verificar si existe un rostro dentro de la imagen o fotografía capturada.

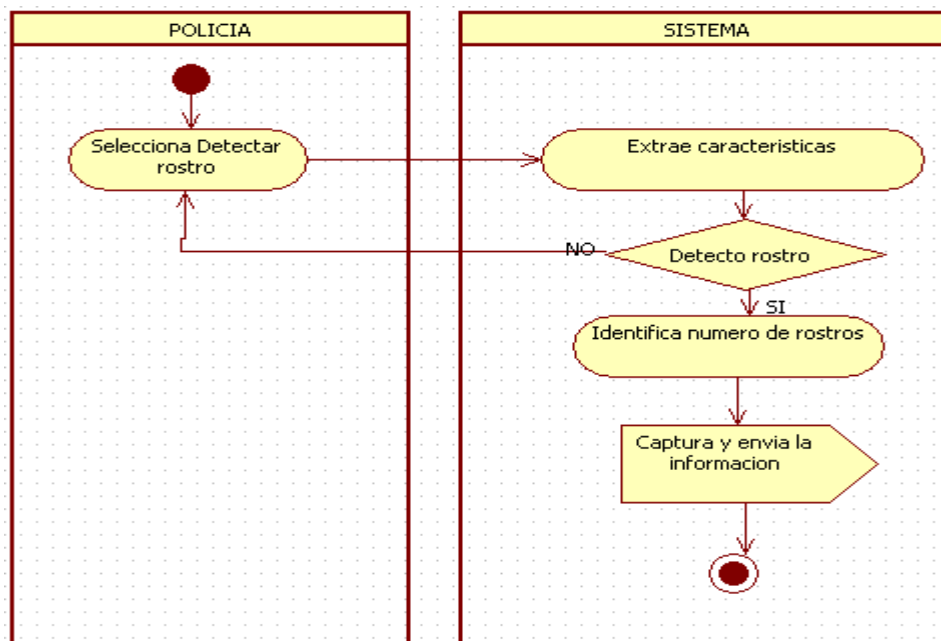


Figura 24. Diagrama de actividades – detectar rostro

Fuente: software StarUML 3.0.2

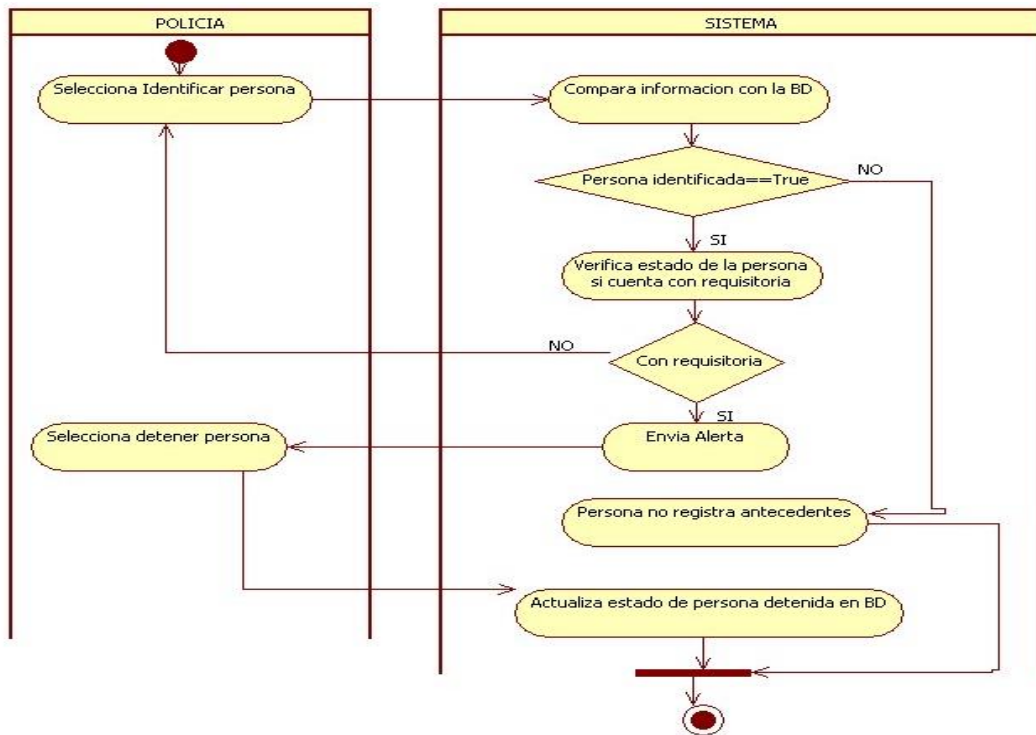


Figura 25. Diagrama de actividades – Identificar rostro

Fuente: software StarUML 3.0.2

3.7.3. Fase de Diseño

El diseño de la aplicación móvil de reconocimiento facial en personas con antecedentes de abuso sexual en la provincia de Andahuaylas se diseñó de la siguiente manera:

A. Diseño de la base de datos

La base de datos que será utilizado en la aplicación está constituida por las siguientes tablas.

Tabla 14. Base de datos del diseño

Ítem	Nombre	Item	Nombre
1	Tabla_Login	4	Tabla_Persona_Antecedentes
2	Tabla_Persona	5	Tabla_Antesedentes
3	Tabla_Sexo		

Fuente: elaboración propia

B. Modelo Lógico.

En el modelo lógico se muestra el significado de las entidades de información de la aplicación y las relaciones que hay entre ellas. A continuación, se muestra las entidades que conforman la aplicación para su funcionamiento ya que solo se diseñó para ver el funcionamiento del sistema este puede ser conectado o modificado de acuerdo a cualquier base de datos es por eso no se presenta una entidad bien desarrollada.

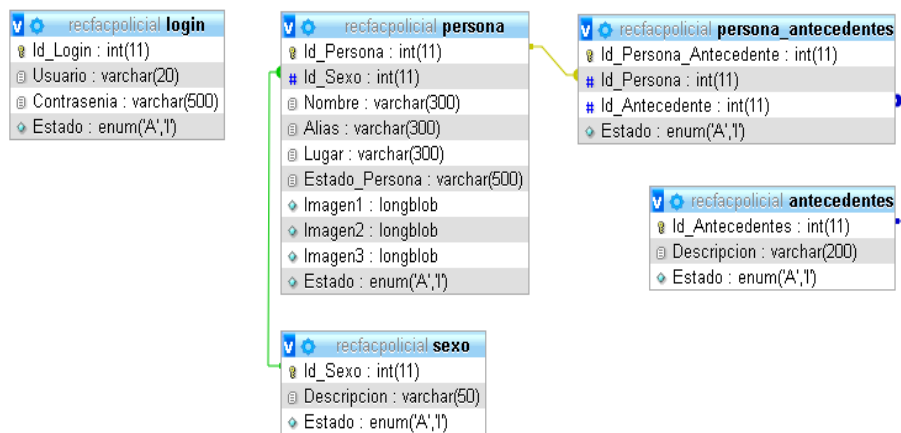


Figura 26. Modelo lógico
Fuente: elaboración propia

3.7.3.1. Diseño del sistema.

El sistema tiene como propósito realizar el reconocimiento facial de personas buscadas y brindar una alerta en tiempo real, así para luego ser capturado en ese instante. Para lo cual en el desarrollo del sistema se tomó en cuenta algunos requerimientos importantes como lo son: velocidad de procesamiento, eficacia, tamaño, además del costo de elementos empleados en el desarrollo del prototipo de sistema de reconocimiento facial, En la figura 15 se muestra el modelo conceptual de la solución: primero el policía tomara una fotografía al individuo intervenido para luego realizar la consulta a la base de datos de la policía nacional del Perú, si el sujeto presenta antecedentes de captura por abuso sexual contra la mujer si los tuviese este será procesado de acuerdo a la ley y procedimientos que los policías manejan, como se observa si se logra este podría ampliarse a mas trabajos de investigación como encontrar personas desaparecidas entre otros.

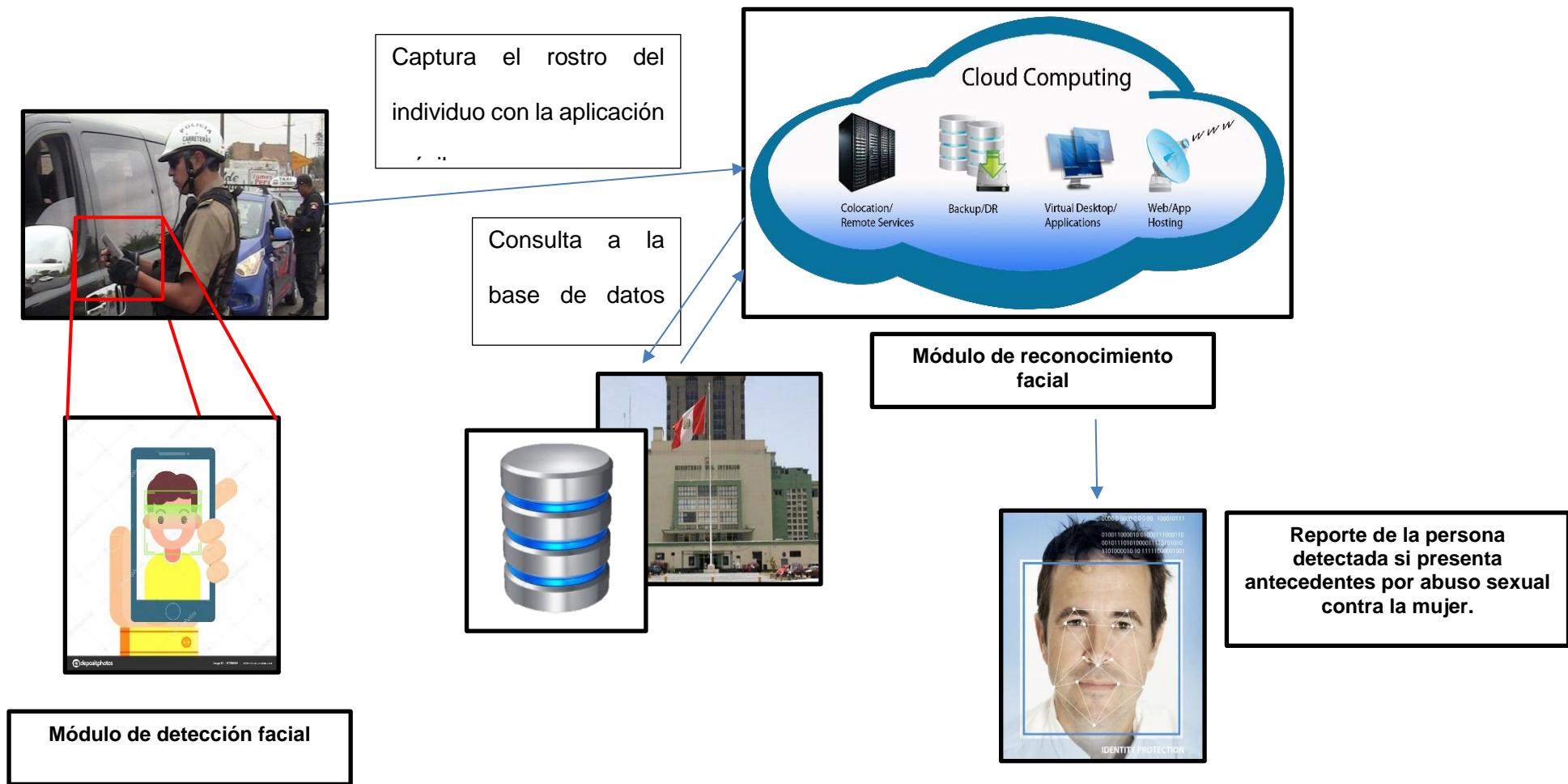


Figura 27. Modelo conceptual de la solución

Fuente: elaboración propia.

A. Adquisición y pre procesamiento

La correcta identificación de las personas en este caso hombres se requiere de un correcto proceso en el cual primeramente se adquiere la imagen mediante la cámara del dispositivo móvil y posteriormente se realiza un pre procesamiento para que la imagen mejore en su resolución, con la finalidad de tener una correcta extracción de características faciales, como se muestra en el diagrama de flujo en la figura Consiste en la captura de la imagen facial de una persona para lo cual el sistema captura una imagen mediante el uso de la cámara para la adquisición de la misma, y es indispensable la utilización de la librería adecuada para el uso de la cámara conectada al sistema.

Los pasos a seguir para la detección del rostro son los siguientes:

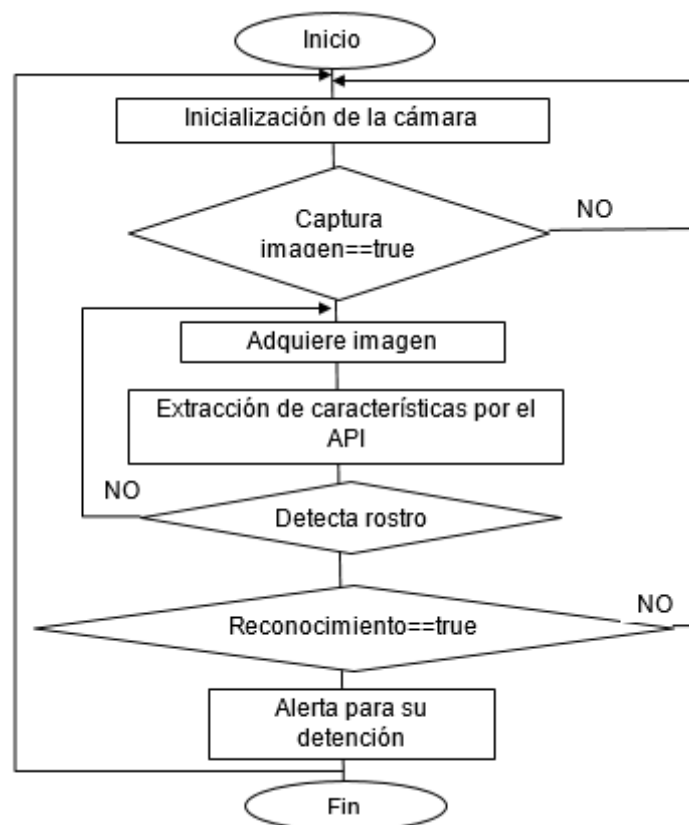


Figura 28. Pasos de detección de rostro

Fuente: elaboración propia.

El primer paso es establecer una conexión entre la cámara del celular y la aplicación de reconocimiento facial, de forma correcta, permitiendo extraer la imagen por medio de un dispositivo móvil no muy sofisticado si no uno promedio que pueda contar exclusivamente con Android 6.0 en adelante, mediante este se extraerán las características faciales por medio del uso del **Face API de Microsoft** para el reconocimiento del rostro de las personas que han pasado por el sistema de reconocimiento.

B. Envío de la información.

Una vez adquirido la imagen este es enviado por la nube, para lo cual será necesario que el dispositivo móvil esté conectado a internet mediante plan de datos o de preferible un **modem LTE**.

La información es receptada en la nube por Cloud computing Azure y por la BD de la policía. En este caso se utilizará un servidor local con Xampp, que es la encargada de realizar el procesamiento de manera centralizada gracias a que se crea un puente virtual por medio de una dirección IP que es establecida para el ordenador y el celular, haciendo uso del software seleccionado para la extracción de imágenes faciales de las personas que pasen por este sistema de identificación y reconocimiento.

C. Extracción de características (Procesamiento principal)

Con la información disponible en el computador central con relación a los rostros almacenados por carpetas se procede con un análisis de los mismos haciendo uso de las funciones de Face API proporcionadas por Microsoft, para poder extraer las características de la imagen para su reconocimiento, internamente realiza procesos importantes como lo son: Lo primero será utilizar el servicio Face a través de su biblioteca de clientes .NET. La aplicación detecta caras en una imagen, dibuja un marco alrededor de cada cara y muestra una descripción de la cara en la barra de estado.

Para lo cual se necesitó una clave de suscripción para ejecutar el algoritmo de inteligencia artificial. Las claves de suscripción de prueba gratuitas se encuentran de **Try Cognitive Services**.

Las características principales son enmarcadas por puntos para hacer referencia de la cara que son una serie de puntos detallados en una cara; Por lo general, los puntos de los componentes faciales como las pupilas, el canto o la nariz. Los puntos de referencia de caras son atributos opcionales que pueden analizarse durante la detección de caras. Puede pasar 'verdadero' como un valor booleano al parámetro de consulta returnFaceLandmarks al llamar a Face - Detect , o usar el parámetro opcional returnFaceLandmarks para el método DetectAsync de la clase FaceServiceClient para incluir los puntos de referencia de la cara en los resultados de la detección.

Por defecto, hay 27 puntos de referencia predefinidos. La siguiente figura muestra cómo se definen los 27 puntos:

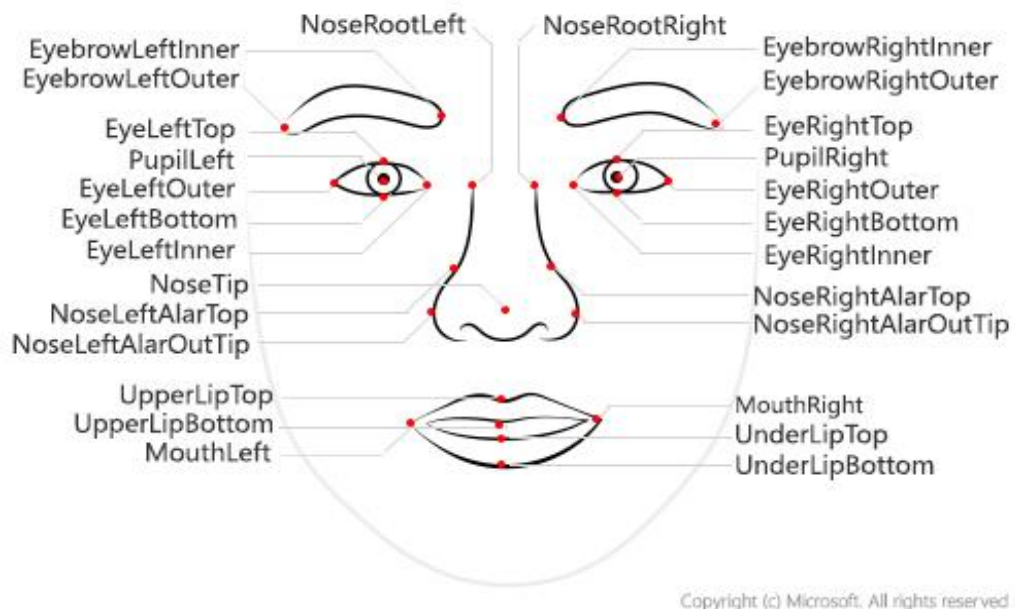


Figura 29. Puntos de referencia

Fuente: (Ggaily, 2017).

3.7.3.2. Entrenamiento del sistema.

Lo primero fue crear un PersonGroup llamado "recFacPolicial": este PersonGroup contiene una lista de personas conocidas para lo cual es necesario subir la foto de una persona en varias poses. Estas caras se utilizan como una línea de base para identificar a las personas. Se recomienda usar caras frontales transparentes, al igual que su identificación con foto. Un buen conjunto de fotos debe contener caras de la misma persona en diferentes poses, colores de ropa o peinados.

A. Crear el grupo de personas llamado "bddelincuentes"

Una persona es una unidad básica de identificación. Una persona puede tener una o más caras conocidas registradas. Sin embargo, un PersonGroup es una colección de personas, y cada persona está definida dentro de un PersonGroup particular. La identificación se realiza contra un PersonGroup. Entonces, la tarea es crear un PersonGroup, y luego crear personas en él.

Esto se ejecuta utilizando el PersonGroup - Crear API. La API de la biblioteca del cliente correspondiente es el método CreatePersonGroupAsync para la clase FaceServiceClient. La ID de grupo especificada para crear el grupo es única para cada suscripción; también puede obtener, actualizar o eliminar los Grupos de Personas utilizando otras API de Grupo de Personas. Una vez que se define un grupo, las personas pueden definirse dentro de él mediante el uso de la persona PersonGroup Person - Create API. El método de la biblioteca del cliente es CreatePersonAsync. Puede agregar cara a cada persona después de que se crean.

Lo primero que se hará es crear el grupo para lo cual el sistema ya define el nombre del grupo para eso se tendrá que seleccionar la opción 1 como se muestra en la figura 30.

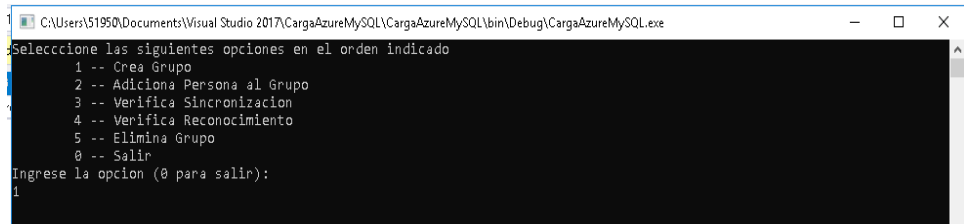


Figura 30. Software Entrenar sistema.

Fuente: elaboración propia.

B. Adición de persona al sistema.

De esta forma se procede con el ingreso de un nuevo individuo al sistema con su fotografía y nombre con el cual será identificado por el sistema y todos sus datos a lo que concierne a un fugitivo de la justicia. En este caso lo que hará este sistema será conectarse a la base de datos de la policía para obtener todas las fotografías de los delincuentes agrupándolos según su DNI ya que en la base de datos será necesario de que, por delincuentes, este alojado por lo menos 3 fotografías de cada sujeto para ser entrenado y pueda reconocer a la persona.

Para adicionar se elegirá la opción 2 como se muestra en la figura 31.

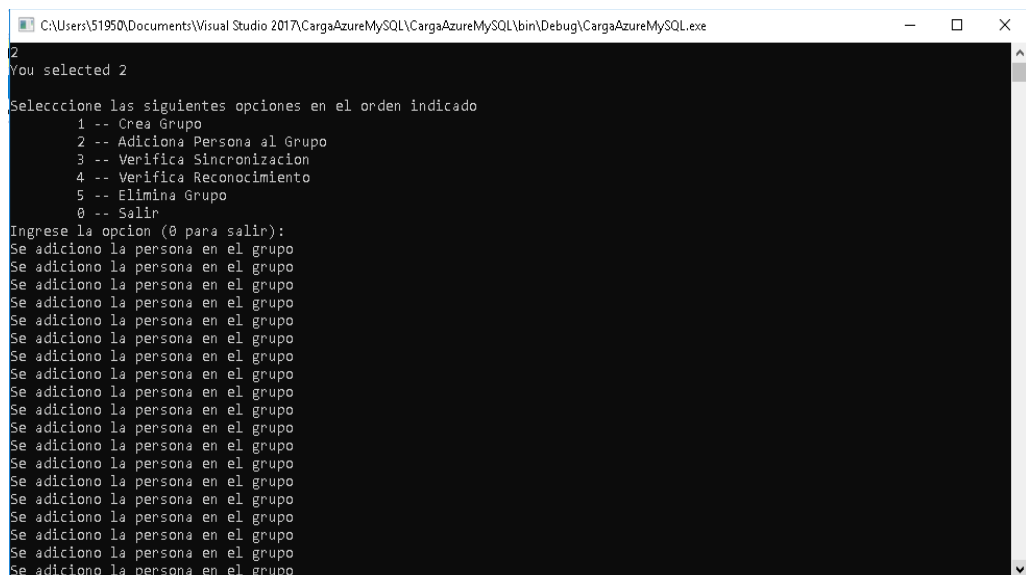


Figura 31. Adicionar persona de la Base de datos BDdelincuentes.

Fuente: elaboración propia.

Como se ve en la figura 31 lo que hace es agregar todas las fotografías de los delincuentes al sistema de reconocimiento de Face API.

Para probar la eficiencia del software se hace uso de las fotografías almacenadas en Yale Facedatabase A, también conocida como Yalefaces. La base de datos de AT&T es buena para las pruebas iniciales, pero es una base de datos bastante sencilla. Es un conjunto de datos más apropiado para los experimentos iniciales, ya que el problema de reconocimiento es más difícil. La base de datos consta de 15 personas (14 hombres, 1 mujer) cada una con 11 320 \ veces 243 píxeles de tamaño de imagen en escala de grises. Hay cambios en las condiciones de luz (luz central, luz izquierda, luz derecha), expresiones faciales (feliz, normal, triste, somnoliento, sorprendido, guiño) y gafas (gafas, sin gafas).

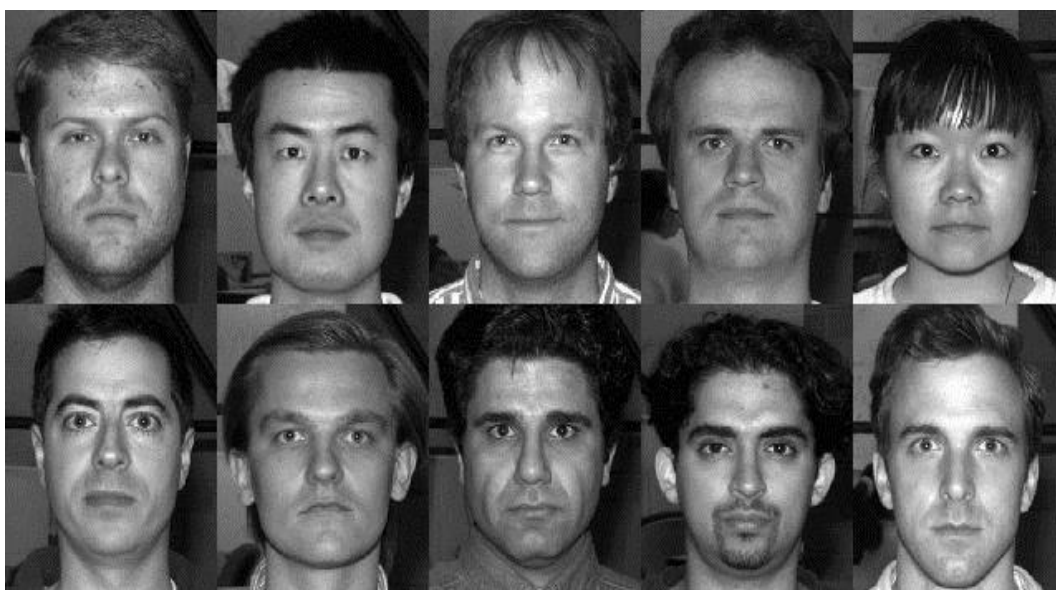


Figura 32. Imágenes de prueba.

Fuente: Yale Facedatabase A

3.7.4. Diseños de la Interfaces Abstractas

Los diseños nos permiten visualizar, ingresar, modificar y guardar los datos o la información en la aplicación.

El Diseño de la interfaz gráfica “Login” es para ingresar el usuario y la contraseña para poder acceder a la ampliación.

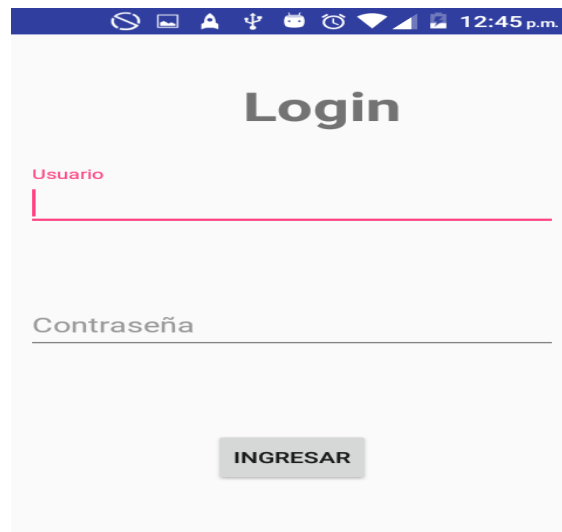


Figura 33. Sistema login

Fuente: Elaboración propia

El Diseño de la interfaz gráfica “Tomar Foto” es para visualizar la cámara del celular.



Figura 34. Interfaz gráfica Tomar Foto.

Fuente: Elaboración propia

El Diseño de la interfaz gráfica “Detectar rostro” es para poder detectar si en la fotografía existen rostros humanos.

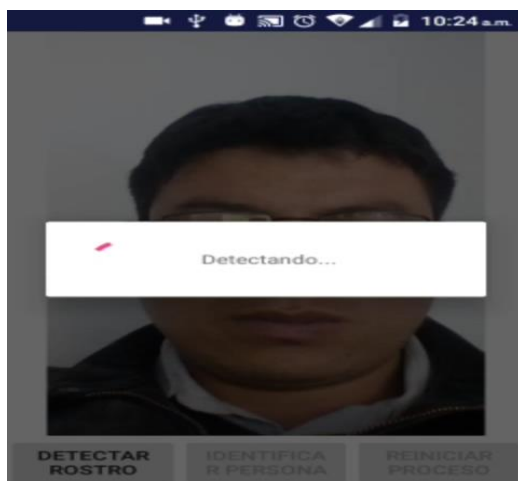


Figura 35. Interfaz gráfica “Detectar rostro

Fuente: Elaboración propia

El Diseño de la interfaz gráfica “Identificar Persona” es para poder identificar a la persona.



Figura 36. Interfaz gráfica Identificar Persona.

Fuente: Elaboración propia

El Diseño de la interfaz gráfica “Alerta de persona” se visualiza el reporte de la persona que será detenida.



Figura 37. Interfaz gráfica Alerta de persona.

Fuente: Elaboración propia

3.7.5. Funcionamiento

En el desarrollo del sistema de reconocimiento facial, fueron necesarias pruebas de Funcionamiento, con la finalidad de determinar los diferentes porcentajes de confiabilidad de dicho sistema, tomando en cuenta que el mismo realiza un procesamiento de la imagen receptada por la cámara del celular una vez tomado por parte del policía, de esta forma la captura del rostro es procesada mediante un análisis de características faciales del individuo en comparación con los rostros almacenados dentro del sistema.

3.7.5.1. Pruebas de usabilidad

La prueba de usabilidad permite conocer si la interfaz de usuario es amigable, intuitiva y si funciona correctamente. Para esta prueba se instaló la aplicación en 3 smartphone y se analizó lo siguiente:

- Número total de actividades realizadas exitosamente.

- El tiempo empleado en realizar una actividad.
- Número total de errores encontrados en la aplicación.

Para analizar las pruebas se indicó al usuario que inicie sesión y realizar las actividades en la aplicación. Al finalizar las pruebas se tomaron capturas (Anexo: Pruebas de captura)

3.7.5.2. Pruebas funcionales

La prueba funcional se aplica para validar si el comportamiento de la aplicación cumple con las especificaciones. Esta prueba se realizó teniendo en cuenta los casos de uso definidos en la fase de análisis y diseño. Los resultados en todos los casos fueron favorables. La Aplicación funciona correctamente de acuerdo a lo especificado.

3.7.5.3. Pruebas de portabilidad

La prueba de portabilidad permite verificar la funcionalidad de la aplicación en diferentes tipos y marcas de teléfonos móviles con plataforma Android. Para ello se utilizaron dispositivos con diferentes versiones de Android 6.0 iguales o en adelante y diferentes resoluciones de pantalla obteniendo resultados satisfactorios, ya que la aplicación se adapta a las pantallas proporcionando los procesos de forma fácil.

CAPITULO 4: RESULTADOS Y DISCUSIÓN

4.1. Análisis de resultados

El sistema realizó pruebas de funcionamiento para determinar el nivel de eficacia en el reconocimiento de las personas que pasan por el sistema, para lo cual se obtiene los siguientes tipos de datos:

Persona Buscada – Identificada: Al momento que una persona buscada pasa por el sistema y es identificada mediante su nombre y posición, este es el caso exitoso y se denomina verdadero positivo.

Persona Buscada – No Identificada: Al momento que una persona buscada pasa por el sistema y no es identificada mediante su nombre y posición, este es el caso fallido y se denomina falso negativo.

Persona No Buscada – No Identificada: Al momento que una persona no buscada pasa por el sistema y no es identificada por el mismo y se denomina verdadero negativo

4.2. Efectividad y confiabilidad en el sistema

Una vez establecidas las pruebas respectivas durante el día y la noche se proceden con el cálculo del porcentaje total de efectividad brindado por el sistema. Para lo cual se realiza un análisis de cada uno de los casos que el sistema brinda, a continuación, se presentan los resultados efectuados durante el día y la noche haciendo uso de la ecuación.

$$x = \frac{\# \text{ de casos efectuados} * 100\%}{\# \text{ total de capturas efectuadas}}$$

Ecuación 1. Ecuación para determinar la efectividad del prototipo.

A. Resultados obtenidos durante el día

Se capturaron 20 fotografías de personas buscadas por el sistema y 10 fotografías a personas externas no registradas en el sistema. En general se obtuvieron 30 capturas entre personas buscadas y no buscadas. En la tabla 15 se muestra la cantidad de capturas realizadas durante el día, para encontrar el porcentaje de efectividad para el

resultado se utiliza la ecuación 1 y el porcentaje de efectividad se muestra en la figura 38.

Tabla 15. Resultados obtenidos durante el día

Resultados de día	Número de casos	porcentaje
Persona buscada - identificada	20	66.7%
Persona buscada - no identificada	0	0%
Persona no buscada-no identificada	10	33.33%

Fuente: elaboración propia

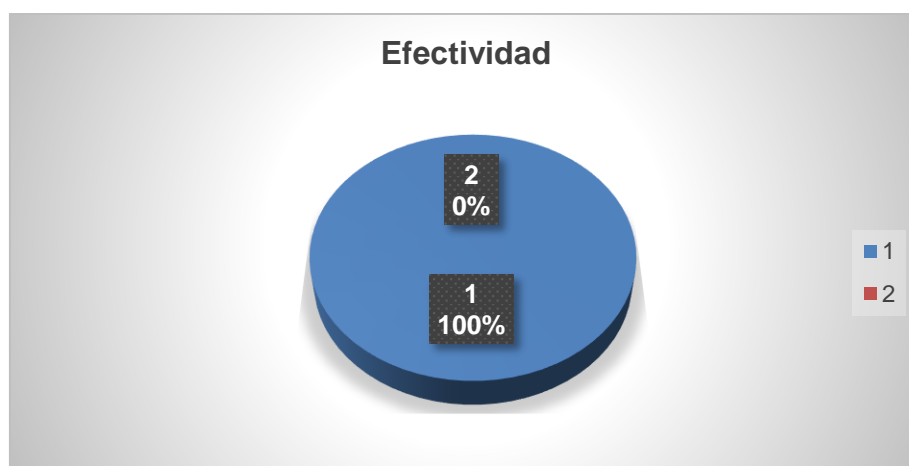


Figura 38. Representación grafica

Fuente: elaboración propia

B. Resultados obtenidos durante la noche

De igual forma que se realizó en el día, en la noche se capturaron en un total de 15 fotografías a individuos buscados y 10 fotos a individuos externos al registro de personas. En general se obtuvieron 25 capturas entre personas buscadas y no buscadas.

En la tabla 16 se muestra la cantidad de capturas realizadas durante la noche, para hallar el porcentaje de efectividad por resultado obtenido se utiliza la ecuación 1 y el porcentaje de efectividad se muestra en la figura 39.

Tabla 16. Resultados obtenidos durante la noche

Resultados de día	Número de casos	porcentaje
Persona buscada - identificada	14	66.7%
Persona buscada - no identificada	1	4%
Persona no buscada-no identificada	10	40%

Fuente: elaboración propia



Figura 39. Representación gráfica -Resultados obtenidos durante la noche

Fuente: elaboración propia

En la tabla 16 se muestran los resultados totales en relación a las pruebas de confiabilidad realizadas y los porcentajes de efectividad se muestran en la figura 40.

Tabla 17. Resultados totales

Descripción	Día	Noche
% de efectividad	100%	93%
% de inconsistencia	0%	7%
% total	100%	100%

Fuente: elaboración propia



Figura 40. Representación gráfica de resultado final.

Fuente: elaboración propia

CONCLUSIONES

Primero: Al realizar pruebas de eficacia, tiempo de respuesta al algoritmo pre entrenado para detección de rostros de OpenCV y Face API de Microsoft son bastante eficaces y tienen menor tiempo de respuesta solo con la diferencia de que Face Api es más eficaz con un menor número de rostros pre entrenados.

El prototipo de reconocimiento facial ofrece un elevado porcentaje de efectividad al momento de realizar la detección de un rostro humano dentro de una fotografía, y esto se debe a que realiza el procesamiento utilizando Inteligencia artificial de servicios cognitivos. Para este caso se utilizó Microsoft Azure, pero este valor en el porcentaje de efectividad no varía. Es casi del 100% de efectividad pero a comparación de los algoritmos de OpenCV al momento de la detección de rostro varían, pudiendo disminuir debido a diferentes factores como lo es la iluminación así como la posición del rostro al momento de realizar la extracción de la imagen facial previo análisis de características principales y homogéneas que tenemos todos los humanos como ojos, cejas, boca, entre otras características que pasa por el sistema de reconocimiento, la cuestión es usar el sistema en un ambiente no controlado para que los policías no tengan dificultades al momento de hacer uso del aplicativo.

Segundo: El prototipo de reconocimiento facial al ser usado en la mañana, tarde o noche es capaz de reconocer a la persona buscada, lo que realiza es mandar la fotografía en Stream a la nube y este es comparado con todas las fotografías entrenadas de los delincuentes para luego dar un reporte sobre a quién pertenece el rostro. Toda esta comunicación es gracias a la tecnología de redes 4g que cuentan los celulares hoy en día, no es necesario tener un modem costoso. No es conveniente usar OpenCV en dispositivos móviles ya que estos no cuentan con la cantidad de memoria para almacenar demasiada información de las fotografías de los delincuentes, al realizar los comparativos sobre la eficiencia en el reconocimiento facial de un delincuente se ve que a los algoritmos de OpenCV aún falta mejorar para extraer los patrones que nos hacen diferentes a todas las personas como

tamaño de ojos, labios, si tenemos alguna cicatriz, entre otros patrones que seguramente **Face API** los tiene bien definido en su algoritmo que es como una caja negra donde no se sabe cuáles son los procesos para reconocer a una persona de manera eficiente.

Tercero: Fue necesario desarrollar un aplicativo que esté conectado a la base de datos y este pueda extraer todas las fotografías de delincuentes en tiempo real y pueda subirlos al Persongrup de Face API para luego compararlos con las fotografías tomadas desde el prototipo de reconocimiento facial. para este proceso el aplicativo puede tomar fotos cuantas veces sea necesario ya que estas fotografías no son guardadas en su memoria interna.

RECOMENDACIONES

Primero: Es recomendable que por cada una de las personas buscadas se tenga un registro de por lo menos cinco imágenes almacenadas en el sistema para obtener el mayor porcentaje de efectividad al momento de la extracción de características por medio del algoritmo Face API , las que son utilizadas al realizar la comparativa de los rasgos faciales con las imágenes procesadas de los rostros de las personas que pasan por el sistema, de igual forma estas imágenes deben tener un formato de tamaño de 640x480 pixeles con el cual se llevó a cabo en el proyecto como estándar para las imágenes capturadas y almacenadas en la etapa de reconocimiento.

Segundo: Se recomienda una correcta manipulación del sistema, tomando como factor importante la ubicación de la cámara buscando evitar las variaciones luminosas excesivas que podrían ocasionar sombras en las imágenes faciales al momento de realizar la captura de las mismas, de esta forma se asegura el correcto funcionamiento en la etapa de procesamiento y extracción de datos de rostros a ser identificados.

Tercero: Se recomienda que este proyecto de investigación se utilice como antecedente para futuras investigaciones, logrando la integración de nuevas funciones y características, con una implementación entorno a las nuevas tecnologías existentes con el objetivo de contar con mejores herramientas para la identificación y reconocimiento de personas buscadas.

REFERENCIAS BIBLIOGRÁFICAS

- Analytics, C. Anaconda: *Completely Free Enterprise-Ready Python Distribution for Large-Scale Data Processing, Predictive Analytics, and Scientific Computing*. In.
- Andrade, N., & Guillermo, C. (2015). *Autenticación por reconocimiento facial para aplicaciones web, utilizando software libre*.
- Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A., . . . Zaharia, M. (2010). *A view of cloud computing*. *Commun. ACM*, 53(4), 50-58. doi:10.1145/1721654.1721672
- Barraza, S. L., Thuillier, E., Will, A., & Rodriguez, S. A. (2013). *Primeros pasos para una aplicación móvil offline de reconocimiento facial*. In: Recuperado de: www.conaiisi.unsl.edu.ar/2013/205-529-1-DR.pdf.
- Bello de Haro, J. M. *Inteligencia artificial para la realización de procesos comunes mediante Watson, NodeJS y Blue Prism*.
- Bluemix, I. (2016). *Next-Generation Cloud App Development Platform*. In.
- Bravo, C. J., Ramírez, P. E., & Arenas, J. (2018). *Aceptación del Reconocimiento Facial Como Medida de Vigilancia y Seguridad: Un Estudio Empírico en Chile*. *Información tecnológica*, 29, 115-122.
- Calli Olvea, J. (2015). *Reconocimiento facial basado en el Algoritmo Eigenface*.
- Cobo, Á. (2005). *PHP y MySQL: Tecnología para el desarrollo de aplicaciones web: Ediciones Díaz de Santos*.
- Collantes Arana, A. A., Bustos, S., & Richard, D. (2013). *Reconocimiento facial de personas basado en el análisis de componentes principales PCA*.
- Cárdenas Taype, B. C. (2017). *Desarrollo de una aplicación móvil con realidad aumentada para facilitar el uso de la guitarra acústica*.
- Duró, V. E. (2001). *Evaluación de sistemas de reconocimiento biométrico*. *Departamento de Electrónica y Automática*. *Escuela Universitaria Politécnica de Mataró*.
- Estarita, J., Jiménez, A., Brochero, J., Escobar, H., & Moreno, S. (2018). *Sistema de Reconocimiento de objetos en tiempo real*. *INVESTIGACIÓN Y DESARROLLO EN TIC*, 8(2).
- Eugenio, S. P., Manuel, T. A., Edgar, R. A., Alejandro, C. S. R. V., & Martin, S. Z. G. (2015). *Diseño y construcción de un sistema SCADA para el control de un campo de Helióstatos*. *La Mecatrónica en México*, 39.
- Fabien. (2012, 2018). *Seguridad*. Retrieved from <https://es.actualitix.com>
- FUZI, C. (2013). *Metodología Ágil para el Diseño de Aplicaciones Multimedia de Dispositivos Móviles*. Retrieved from <https://es.slideshare.net/cristopherf/metodologia-agil-para-el-diseo-de-aplicaciones-multimedias-moviles>

- Gallardo Calvopiña, E. S., & Sánchez Quevedo, E. X. (2016). Diseño y construcción de un sistema de autenticación con reconocimiento facial mediante procesamiento de imágenes con la utilización de software libre y tecnología Raspberry Pi. Universidad de las Fuerzas Armadas ESPE Extensión Latacunga. Carrera de Ingeniería en Electrónica e Instrumentación.,
- García, J. O., Fernández, F. A., & Belmonte, R. C. (2008). Biometría y seguridad: Fundación Rogelio Segovia para el Desarrollo de las Telecomunicaciones.
- García, P. P. G. (2013). Reconocimiento de imágenes utilizando redes neuronales artificiales. Facultad de Informática, Universidad Complutense de Madrid, 5.
- García Sevilla, M. (2014). Integración de un sistema de reconocimiento facial en dispositivos móviles basado en OpenCV.
- ggailey777. (18/10/2017). Guía de inicio para desarrolladores en Azure | Microsoft Docs. Retrieved from <https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/guides/developer/azure-developer-guide.md>
- ggailey777. (2017). Guía de inicio para desarrolladores en Azure | Microsoft Docs.
- González, R. B. (2017). Detección facial y reconocimiento anímico mediante las expresiones faciales.
- Gómez, T., & Mayumi, C. (2016). Análisis comparativo de los algoritmos Fisherfaces y LBPH para el reconocimiento facial en diferentes condiciones de iluminación y pose, Tacna–2015.
- Ilbay, Ñ., Fabricio, M., Córdova, T., & Gonzalo, L. (2013). Análisis de eficiencia en algoritmos de reconocimiento de imágenes digitales aplicables a dispositivos móviles bajo la plataforma Android. Universidad de las Fuerzas Armadas ESPE. Carrera de Ingeniería en Sistemas e Informática.,
- INEI. (2016). INEI. Retrieved from <https://www.inei.gob.pe/estadisticas/indice-tematico/tecnologias-de-la-informacion-y-telecomunicaciones/>
- Luxand, I. (2010). Luxand-detect human faces and recognize facial features with luxand FaceSDK. In.
- López Ramírez, C. O. (2014). Diseño e implementación de un sistema de reconocimiento facial usando Matlab.
- Martínez, D., Loaiza, H., & Caicedo, E. (2011). Una propuesta para incrementar por medio de Algoritmos Genéticos la capacidad discriminante de las técnicas PCA y LDA aplicadas al Reconocimiento de Rostros con Imágenes IR. Ingeniería y Ciencia, 7(13), 111-130.
- Marín Posadas, C. O. (2012). Prototipo de sistema de seguridad con reconocimiento facial.
- Mery, D. (2004). Visión por computador. Santiago de Chile. Universidad Católica de Chile.
- Microsoft, C. (2010). Plataforma y servicios de computación en la nube de Microsoft Azure. Retrieved from <https://docs.microsoft.com/en-us/azure/guides/developer/azure-developer-guide>

- Morccolla Ancco, G. (2017). Desarrollo de una aplicación móvil de alerta electrónica académica para padres de familia de la institución educativa Juan Espinoza Medrano.
- Moreano, J. A. C., Pulloquina, R. H. M., Lagla, G. A. F., Chisag, J. C. C., & Pico, O. A. G. (2017). Reconocimiento facial con base en imágenes. *Revista Boletín Redipe*, 6(5), 143-151.
- Narambuena, L., Vaiman, M., & Pereno, G. L. (2016). Reconocimiento de Emociones Faciales en Adultos Mayores de la Ciudad de Córdoba. *Psykhe*, 25(1), 1-13.
- Ortega, F., González Ispuerto, B., & Pérez Peláez, M. (2015). Audiencias en revolución, usos y consumos de las aplicaciones de los medios de comunicación en tabletas y teléfonos inteligentes. *Revista Latina de Comunicación Social*(70).
- Parmar, D. N., & Mehta, B. B. (2014). Face recognition methods & applications. arXiv preprint arXiv:1403.0485.
- Parra, K. F., Saroza, A. G., Martínez, Y. R., & Bello, I. P. (2016). PMBOK y PRINCE 2 similitudes y diferencias. *Revista Científica*, 3(23), 111-123.
- Raschka, S. (2014). Linear discriminant analysis. Sebastian Raschka's Website.
- Rodríguez Baque, K. E. (2017). Análisis e implementación del algoritmo de detección facial de violaciones Jones.
- Rojas-Cala, J. P. (2017). Autenticación en Windows a través de reconocimiento facial con dispositivo móvil.
- Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011). ORB: An efficient alternative to SIFT or SURF. Paper presented at the Computer Vision (ICCV), 2011 IEEE international conference on.
- Santana, G., & Adrián, M. Reconocimiento por Biometría Facial para Aplicaciones en Ciudades Inteligentes.
- Santana, M. A. G., Díaz-Sánchez, L., Paz, I. T., & Huertas, M. R. Estado del arte en reconocimiento facial.
- Serrano, F. E. A., Ortega, J. C. P., Araiza, E. A. R., & Araiza, J. E. R. (2018). Desarrollo de un proceso de autenticación facial en un sistema android utilizando el algoritmo Ida (análisis de discriminación lineal). *Pistas Educativas*, 39(128).
- Sánchez, G., & Elizabeth, S. (2014). Diseño de un sistema de autenticación biométrica basado en reconocimiento facial.
- Thibaud, C. (2006). MySQL 5: instalación, implementación, administración, programación: Ediciones ENI.
- Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1), 71-86.
- Villa Palacios, S. M. (2015). Sistema automático de reconocimiento de rostros.

- Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. Paper presented at the Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on.
- Wang, Y.-Q. (2014). An Analysis of the Viola-Jones face detection algorithm. Image Processing On Line, 4, 128-148.
- OpenCV. (2018). <https://docs.opencv.org> Retrieved from <https://docs.opencv.org>

Anexo A. Requisitos necesarios de pre-instalación

Para realizar una correcta ejecución del sistema de reconocimiento son necesarios algunos requerimientos entre los cuales se tienen:

Requisitos de hardware y software:

A continuación, detallo el catálogo de elementos de la vista de despliegue

Tabla 18. Requisitos de hardware y software

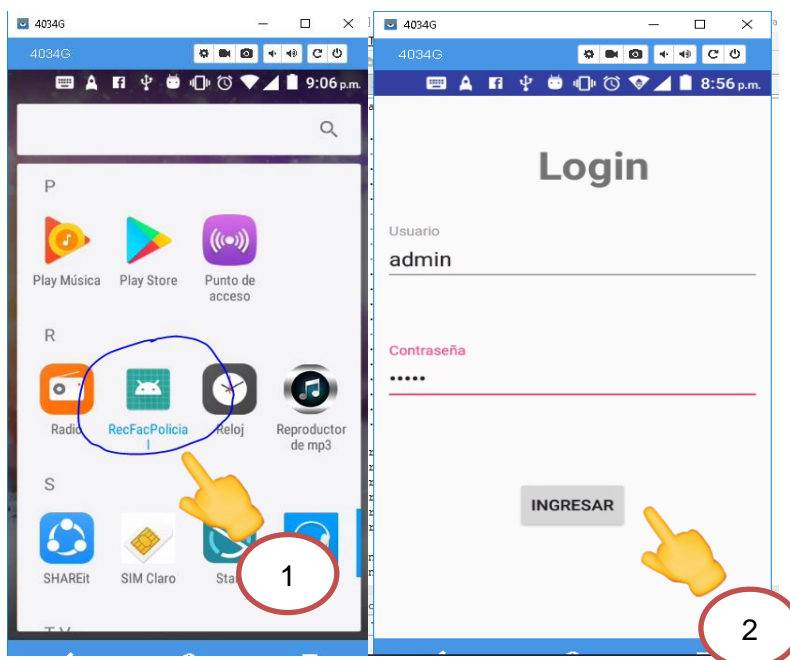
Elemento	Descripción.
Azure Server	Servidores físicos en donde se encuentran los servidores web de la plataforma de Azure. Este es ajeno al desarrollo y se desconoce sus características.
Celular o Tablet	Un dispositivo que corre Android. La aplicación corriendo dentro del sistema operativo de este dispositivo se comunica mediante HTTP con el servidor de la aplicación web, "Web Server con NodeJS Runtime", consumiendo su API en formato JSON.
PC	Dispositivo que tenga un navegador web. Se comunica con el servidor de la aplicación web, "Web Server con PHP", mediante HTTP, utilizando el servidor apache.
Android	El sistema operativo Android con versión igual o mayor a 6.0, corriendo en un dispositivo.
MySQL	Es la base de datos utilizada por la aplicación web de lógica del negocio, donde están guardados los usuarios y sus fotografías, la información de los diagnósticos y las enfermedades y todas las entidades del sistema. Es utilizada mediante SQL, configurando la dirección de la base al servidor externo.
Azure Face API	La API de Azure Face es un servicio cognitivo que proporciona algoritmos para detectar, reconocer y analizar rostros humanos en imágenes. La capacidad de procesar información de rostro humano.

Xampp	XAMPP es un paquete de software libre, que consiste principalmente en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script PHP y Perl. El nombre es en realidad un acrónimo: X, Apache, MariaDB/MySQL, PHP, Perl.
Android Studio 3.1.4	Android Studio es el entorno de desarrollo integrado oficial para la plataforma Android.
Visual studio 2017	Microsoft Visual Studio es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta múltiples lenguajes de programación, tales como C++, C#, Visual Basic.
Windows 10	Windows 10 es el vigente sistema operativo desarrollado por Microsoft como parte de la familia de sistemas operativos Windows NT.

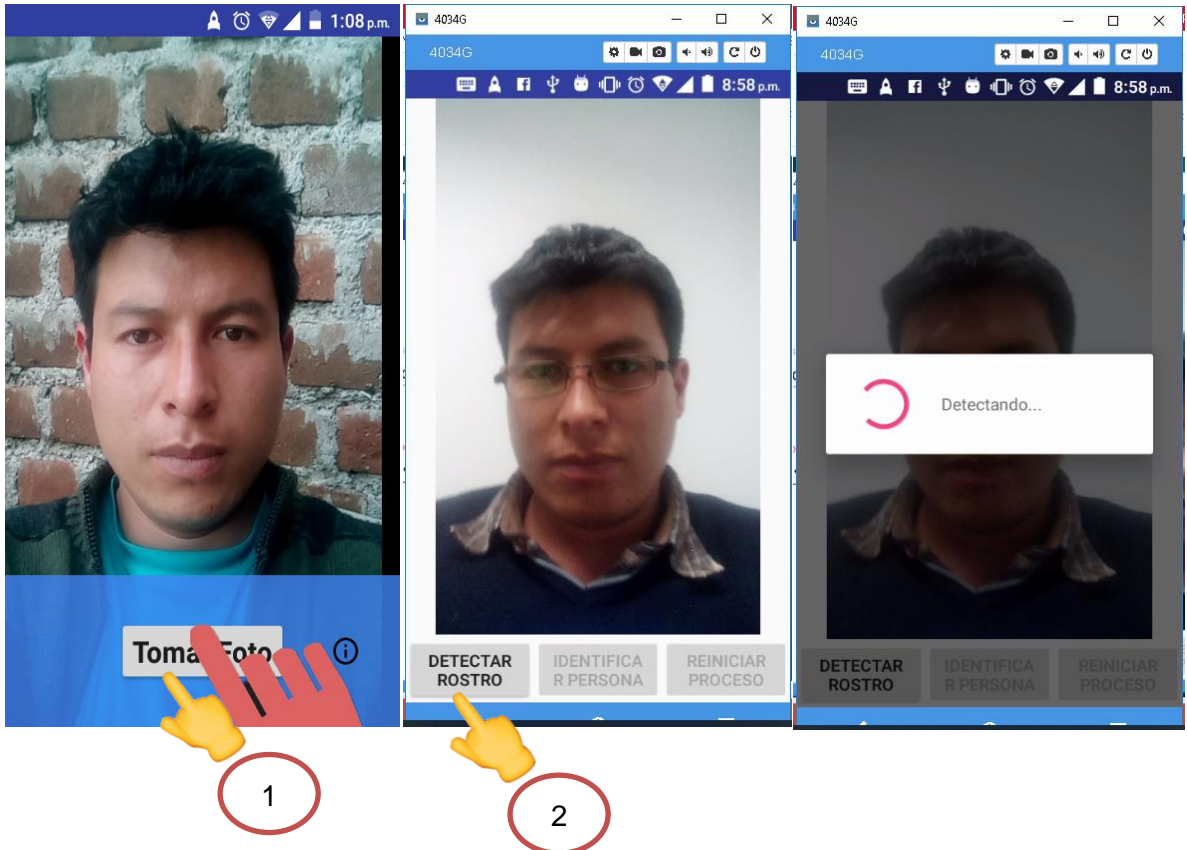
Fuente: elaboración propia

Anexo B: Manual de usuario.

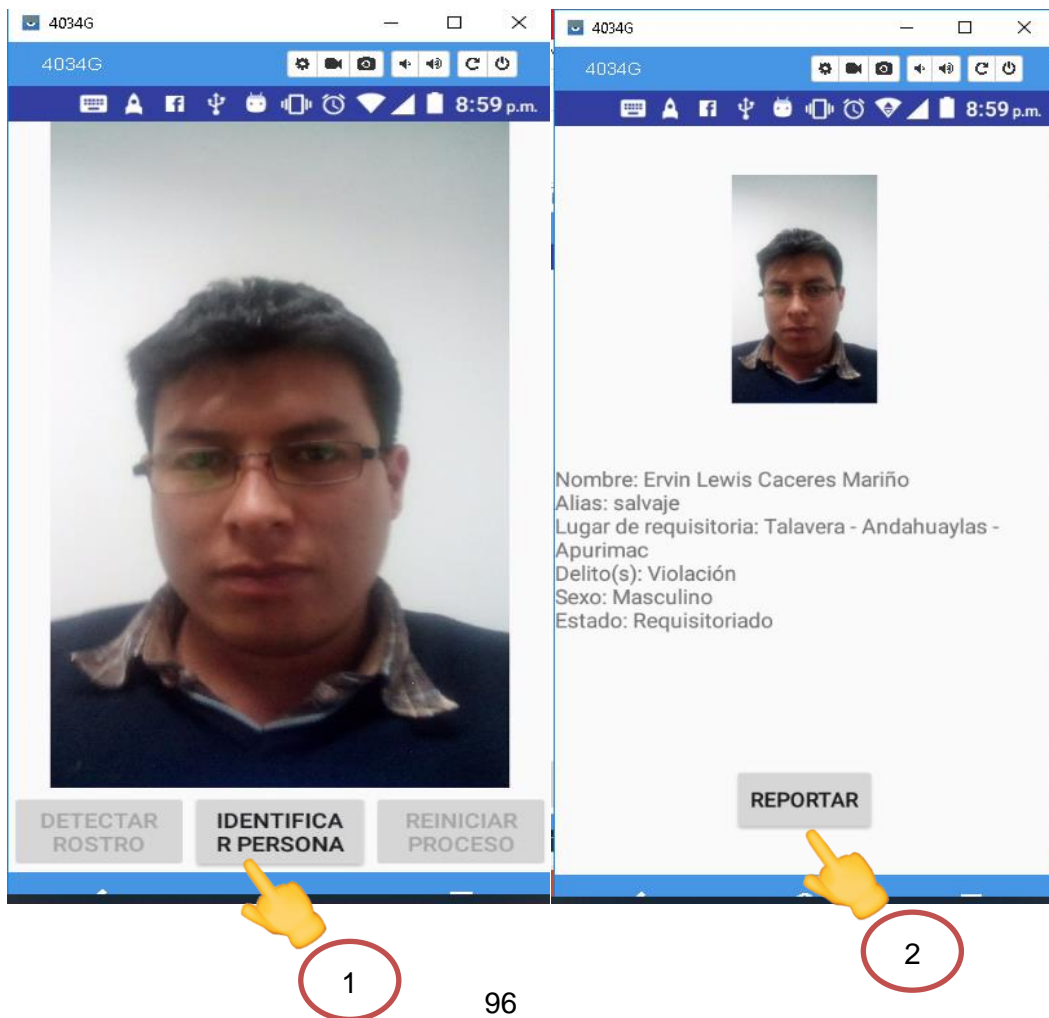
1. Ingresar al menú del dispositivo móvil, ejecutar la aplicación e iniciar sesión, el Policía para iniciar y confirmar sesión deberá ingresar su usuario y contraseña.

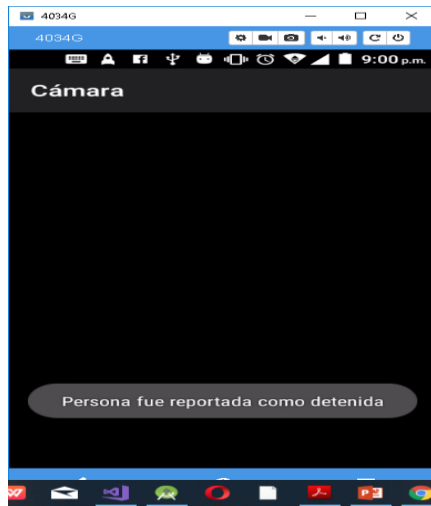


2. El Policía inicia la captura en el instante que se activa la cámara. Posterior selecciona detectar rostro.



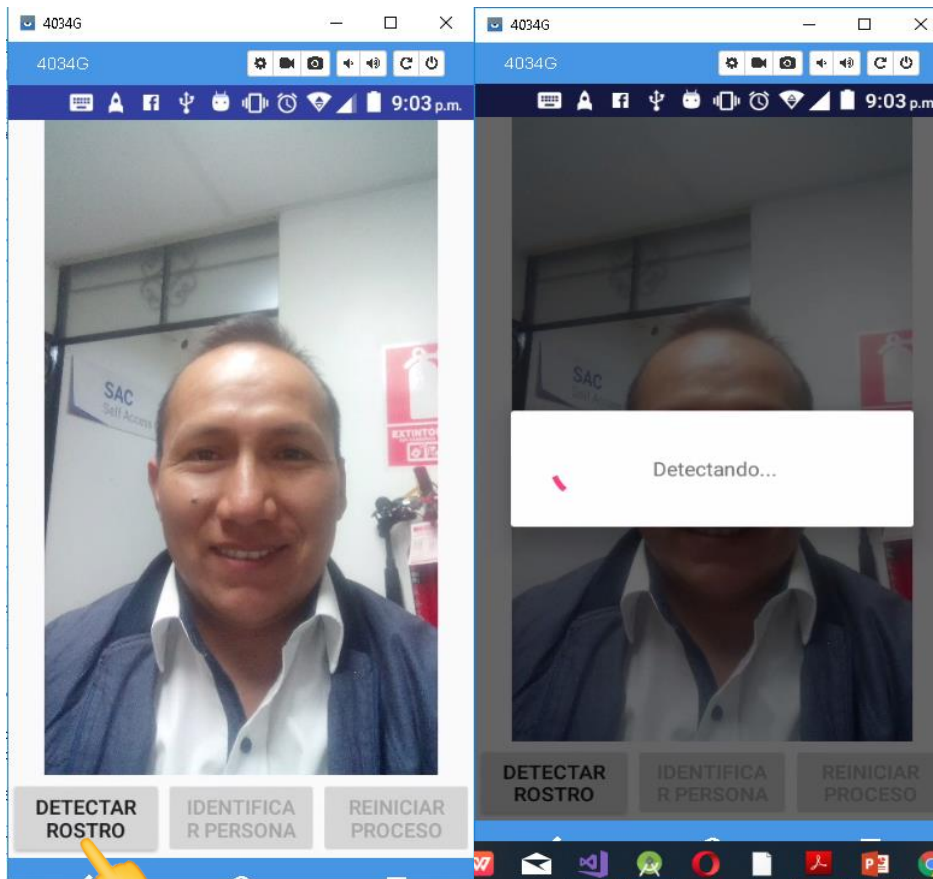
3. Si detecta rostro humano; seleccionar identificar persona.



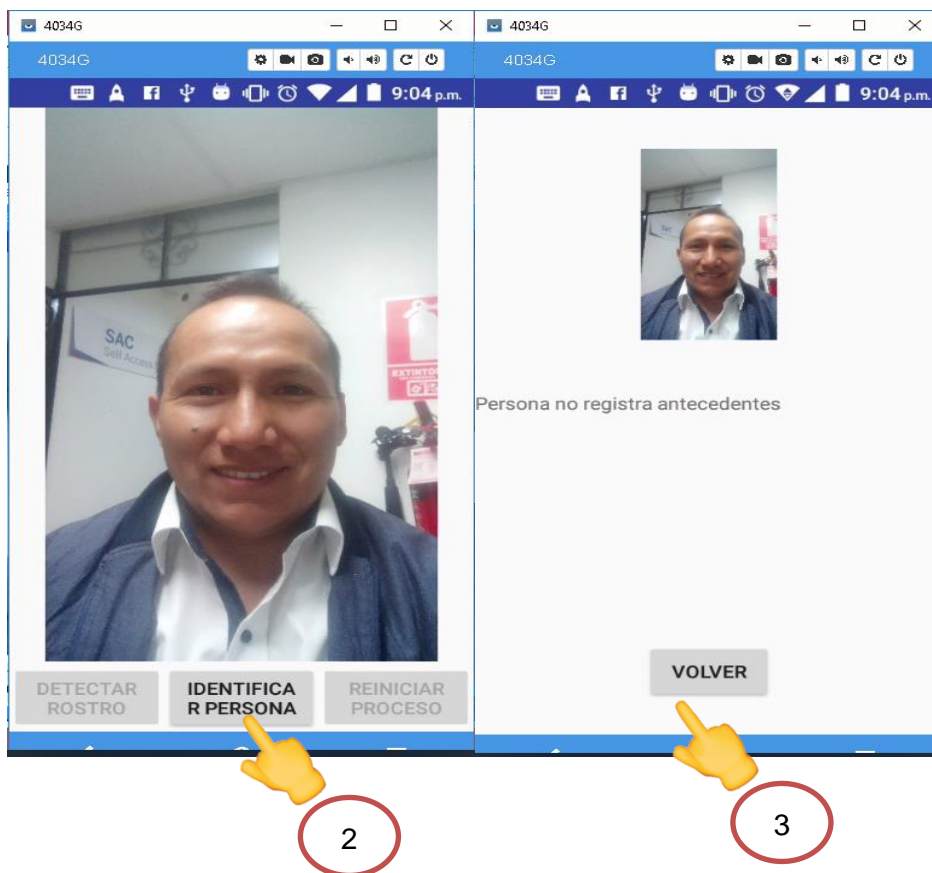


Al momento de reportar a la persona otra vez se activará la cámara de forma inmediata para escanear el siguiente rostro y nos saldrá un mensaje que la persona fue detenida.

4. En caso de que no sea la persona buscada saldrá de la siguiente manera: seleccionar volver para activar la cámara y tomar la fotografía otra vez.



1



Anexo C: Manual de programador.

El propósito de este manual del programador es dar a conocer al lector todos los listados del programa realizado.

Entorno de desarrollo.

Para trabajar con el proyecto se necesita tener instalados los siguientes programas y dependencias:

- ✓ Java JDK 7.
- ✓ Android Studio.
- ✓ Visual Studio 2017.
- ✓ OpenCV.
- ✓ Anaconda 1.9.2
- ✓ Xampp v3.2.2

A continuación, se indica como instalar y configurar correctamente cada uno de ellos.

Java JDK 7

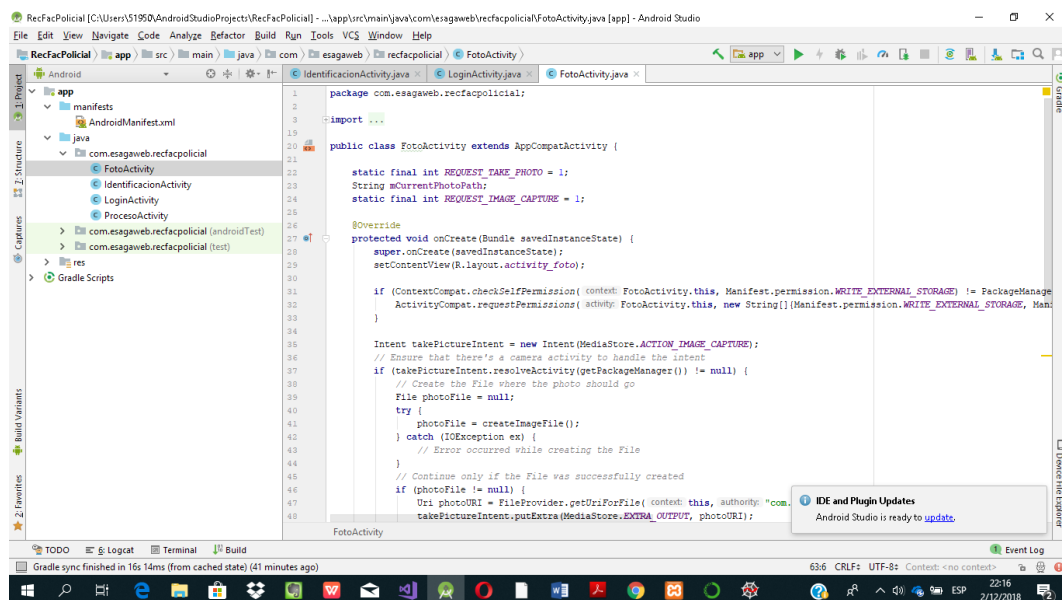
El lenguaje de programación más popular para realizar aplicaciones Android es Java.

Hoy en día, Android no soporta la versión 8 de Java, por lo que tenemos que trabajar con la versión 7. Podemos obtener esta versión desde la pagina de oracle. Se debe elegir correctamente el sistema operativo y la arquitectura del ordenador y posteriormente seguir el asistente de instalación.

Android Studio.

Android Studio es el IDE oficial para el desarrollo de aplicaciones Android. Está basado en IntelliJ IDEA de JetBrains. Proporciona soporte para Gradle, emulador, editor de layouts, refactorizaciones específicas de Android, herramientas Lint para detectar problemas de rendimiento, uso, compatibilidad de versión, etc.

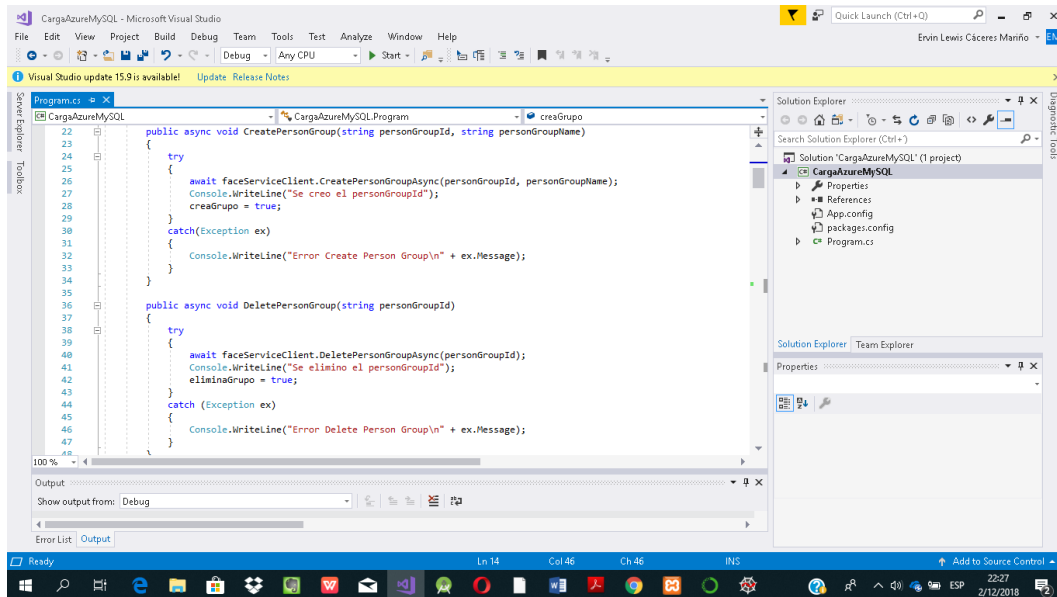
Se puede obtener desde developer.android.com. Junto con Android Studio se instala también el Android SDK y Android Virtual Device (AVD).



Visual Studio 2017

Para el desarrollo del entredador se realizo en visual studio community 2017 que es la version gratuita para desarrolladores que estan empezando al desarrollo de aplicaciones

mobiles, web, escritorio para obtener este programa se debe descargar de visualstudio.microsoft.com.



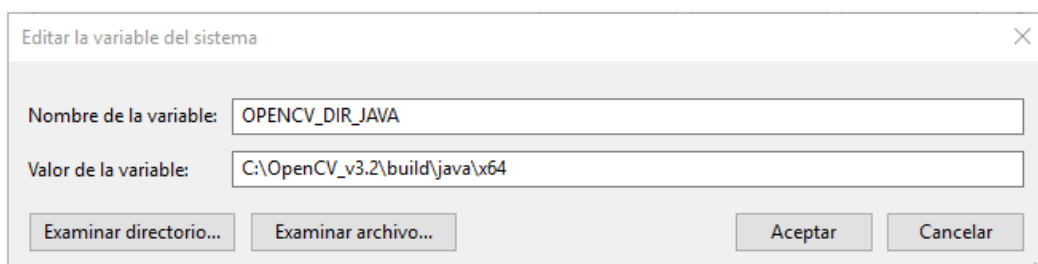
OpenCV

OpenCV es un paquete Open Source de visión artificial que contiene más de 2500 librerías de procesamiento de imágenes y visión artificial, escritas en C/C++ a bajo/medio nivel.

Para ejecutar OpenCV en un dispositivo Android se necesita tener instalado la aplicación OpenCV Manager. Sin embargo, para el desarrollo de la aplicación también debemos instalar la versión de escritorio de OpenCV para poder ejecutar los test de integración del algoritmo en local.

Podemos obtener OpenCV desde la página oficial opencv.org. En este proyecto hemos utilizado la versión 3.2.

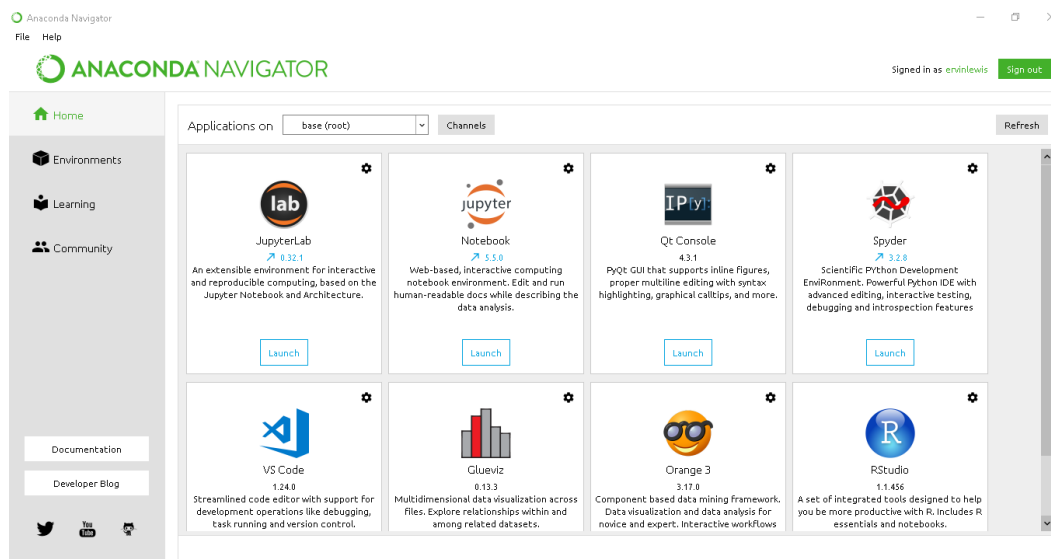
Una vez instalada, tenemos que añadir al path de Windows el directorio donde se encuentran los ejecutables (opencv/build/java/x64 o x86).



Anaconda 1.9.2

Anaconda Navigator es una interfaz gráfica de usuario (GUI) de escritorio incluida en la distribución de Anaconda® que le permite iniciar aplicaciones y administrar fácilmente paquetes, entornos y canales de Conda sin usar comandos de la línea de comandos. Navigator puede buscar paquetes en Anaconda Cloud o en un repositorio local de Anaconda. Está disponible para Windows, macOS y Linux.

Para obtener el navegador, desde anaconda.com, La sección de inicio de Navigator con Navigator muestra cómo iniciar Navigator desde los accesos directos o desde una ventana de terminal.



Actualizamos Python a 3.2

```
Anaconda Prompt - conda install python
(base) C:\Users\51950>conda install python
Solving environment: |
```

Instalamos la librería OpenCV

```
Anaconda Prompt - conda install opencv
¿Desea terminar el trabajo por lotes (S/N)? n
(base) C:\Users\51950>conda install opencv
Solving environment: \
```


httpd.conf: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
# If your host doesn't have a registered DNS name, enter its IP address here.
```

```
#
```

```
#ServerName localhost:8080
```

```
ServerName 192.168.8.100:8080
```

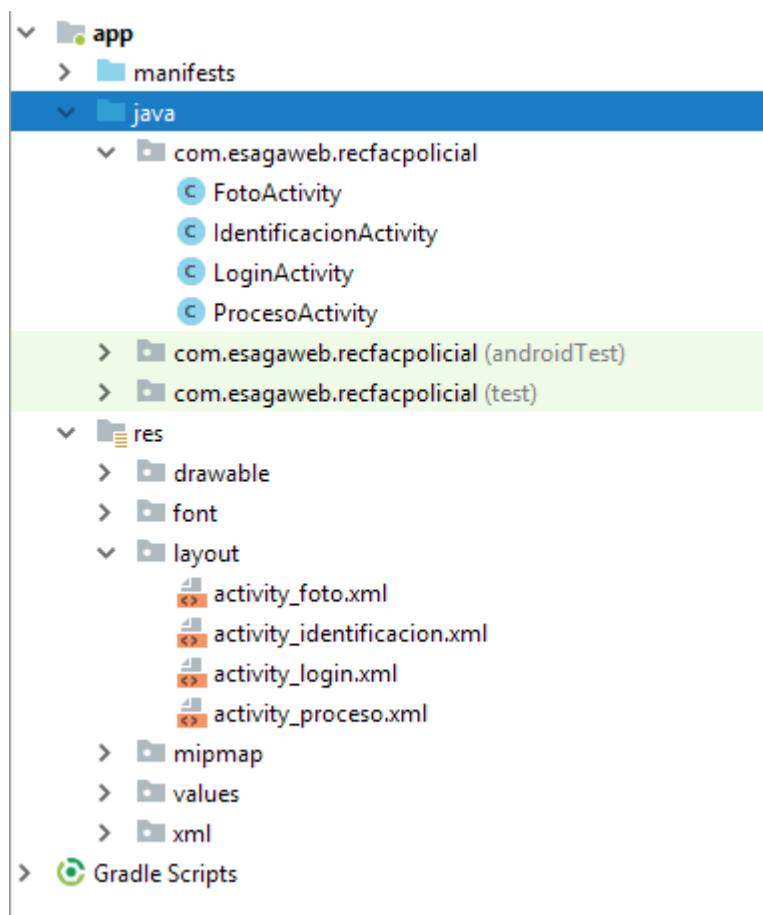
```
..
```

Diseño de la aplicación en Android Studio.

Para la implementación fue necesario la creación de 4 clases:

- ✓ FotoActivity
- ✓ IdentificacionActivity
- ✓ LoginActivity
- ✓ ProcesoActivity

Cada actividad tiene su layout como se muestra en la captura:



Ejecutar aplicación

La aplicación se puede ejecutar en un dispositivo real o en un emulador.

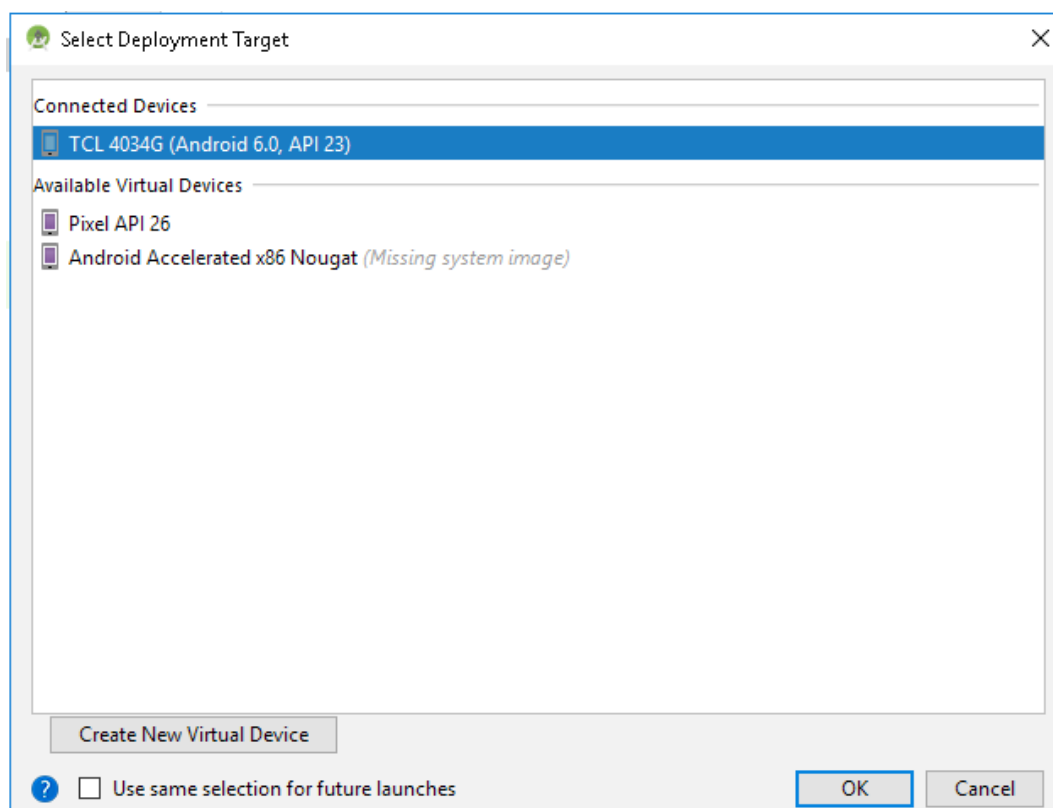
Dispositivo real

Para ejecutar la aplicación en un dispositivo real, se debe conectar este al equipo de desarrollo mediante un cable USB. El equipo debe tener los drivers del dispositivo instalado, sino no lo reconocerá.

Una vez conectado el dispositivo:

Presionar el botón Run.

Si el equipo reconoce el dispositivo se mostrará su nombre debajo de “Connected Devices”.



Seleccionar el dispositivo y pulsa Ok.

Se transferirá el ejecutable de la aplicación y se instalará.

Una vez instalada, se podrá utilizar la aplicación desde el dispositivo.

Emulador

Un emulador (denominados Android Virtual Device - AVD) es una aplicación que simula el funcionamiento de un dispositivo real Android. La creación y gestión de los emuladores se hace a través de AVD Manager.

Para ejecutar la aplicación en un emulador:

Presionar el botón de Run.

Si ya se posee algún emulador instalado, se mostrará en la lista de Android Virtual Devices.

Si no, presionar el botón “Create New Virtual Device”.

Seleccionar las características que se deseen para el emulador y pulsa finalizar.

Seleccionar el emulador creado y pulsar Ok.

Se iniciará el emulador y se instalará la aplicación en él.

Una vez instalada, se podrá utilizar la aplicación desde el emulador.

Exportar aplicación

Para exportar la aplicación como un fichero .apk:

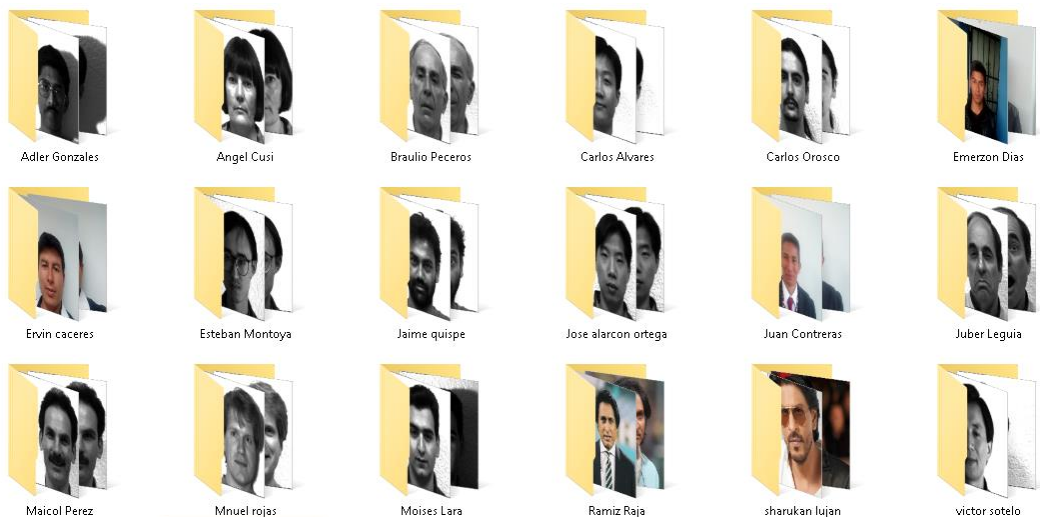
Menú Build > Generate APK.

Se generará un archivo apk y se guardará en build/output/apk.

Anexo D: tabla de personas con fotografías para pruebas en MySQL.

Base de datos: recfapolicial, Tabla: persona, Propósito: Volcado de datos

Id Persona	Id Sexo	Nombre	Alias	Lugar	Estado Persona	Estado
46526534	2	Ervin Lewis Caceres Mariño	salvaje	Talavera - Andahuaylas - Apurimac	Requisitorado	A
46582998	2	Manuel Rojas Castillo	moustro del lago	Andahuaylas - Andahuaylas - Apurimac	Requisitorado	A
70401234	2	Victor Sotelo	diablo	Andahuaylas - Andahuaylas - Apurimac	Requisitorado	A
70404615	2	Sharukan Lujan	indio loco	San Borja-Lima - Lima	Requisitorado	A
72073288	2	Juber Leguia	Narizon	Andahuaylas - Andahuaylas - Apurimac	Requisitorado	A
72257051	2	Braulio Peceros	mountro de los andes	Andahuaylas - Andahuaylas - Apurimac	Requisitorado	A
72780799	2	Jose Alarcon Ortega	Lloron	Andahuaylas - Andahuaylas - Apurimac	Requisitorado	A
72872424	2	Esteban Montoya	coyote	Lima - Lima -lima	Requisitorado	A
73127000	1	Salome Cusi	serrano	Andahuaylas - Andahuaylas - Apurimac	Requisitorado	A
74496511	2	Juan Contreras	Maricon	San Juan Del Urigancho - Lima -Lima	Requisitorado	A
74984523	2	Wimar Aparco	Desconocido	Santa Maria de chico - Andahuaylas - Apurimac	Requisitorado	A
76056640	2	Adler Gonzales	maton	Andahuaylas - Andahuaylas - Apurimac	Requisitorado	A
76130114	2	Jaime Quispe		Abancay -Abancay - Apurimac	Requisitorado	A
76209968	2	Carlos Alvares	el cojo	Andahuaylas - Andahuaylas - Apurimac	Requisitorado	A
76329709	2	Moises Lara	maiquito	Pampachiri -Andahuaylas - Apurimac	Requisitorado	A
76647320	2	Carlos Orosco	zorro	Andahuaylas -Andahuaylas - Apurimac	Requisitorado	A
76818748	2	Maicol Perez Huayta	casador	Abancay - Abancay - Apurimac	Requisitorado	A
77200935	2	Ramiz Raja	loco flaco	Talavera -Andahuaylas - Apurimac	Requisitorado	A
77346301	2	Maicol Perez	loco del acha	Andahuaylas -Andahuaylas - Apurimac	Requisitorado	A
77917148	2	Emerzon Diaz	El chato	Andahuaylas -Andahuaylas - Apurimac	Requisitorado	A



Descripción de los campos.

Id_Persona: en este campo se almacena los DNI de las personas con antecedentes.

Nombre: en este campo se almacena los nombres de los sujetos de prueba.

Alias: apodos con los que son conocidos por las personas o vecinos o se hacen llamar estos delincuentes, términos usualmente usado por la policía para identificar a estos delincuentes.

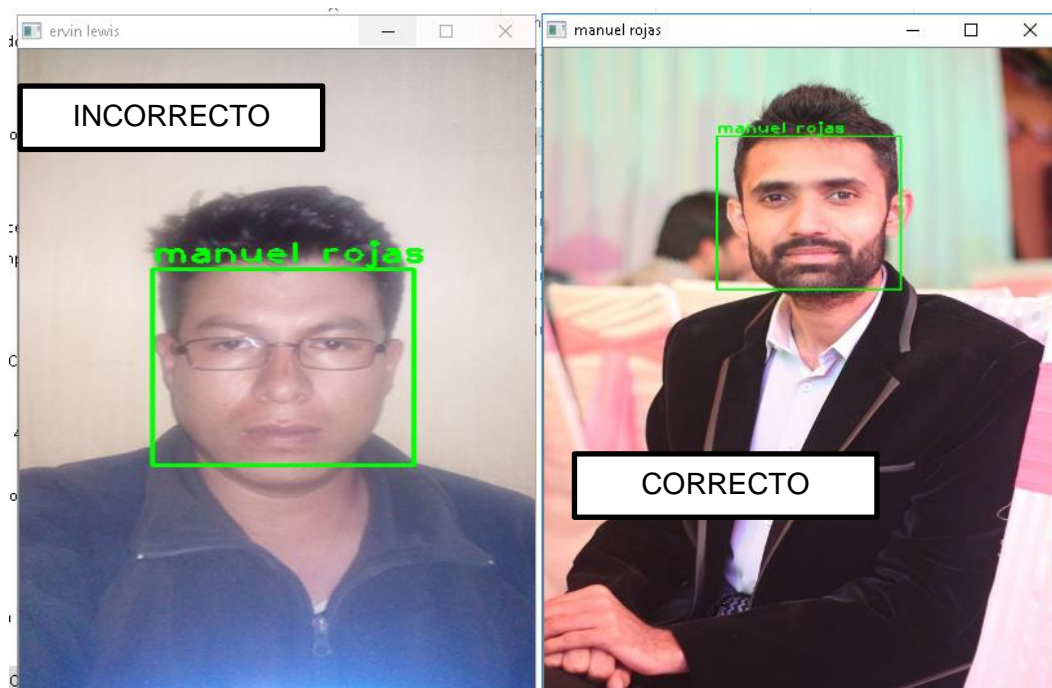
Lugar: Lugar donde mantienen procesos judiciales o son requisitorados por la policía.

Estado_Persona: Estado de la persona si cuento con requisitoria o está detenido, al momento de realizar el reconocimiento si es identificado como requisitorado este cambia a estado detenido.

Estado: Una de las ventajas de MySQL es nos automatiza este campo si la persona está viva o llano en caso de que no esté viva se pone inactivo.

Anexo E: Errores encontrados al momento del reconocimiento con OpenCV.

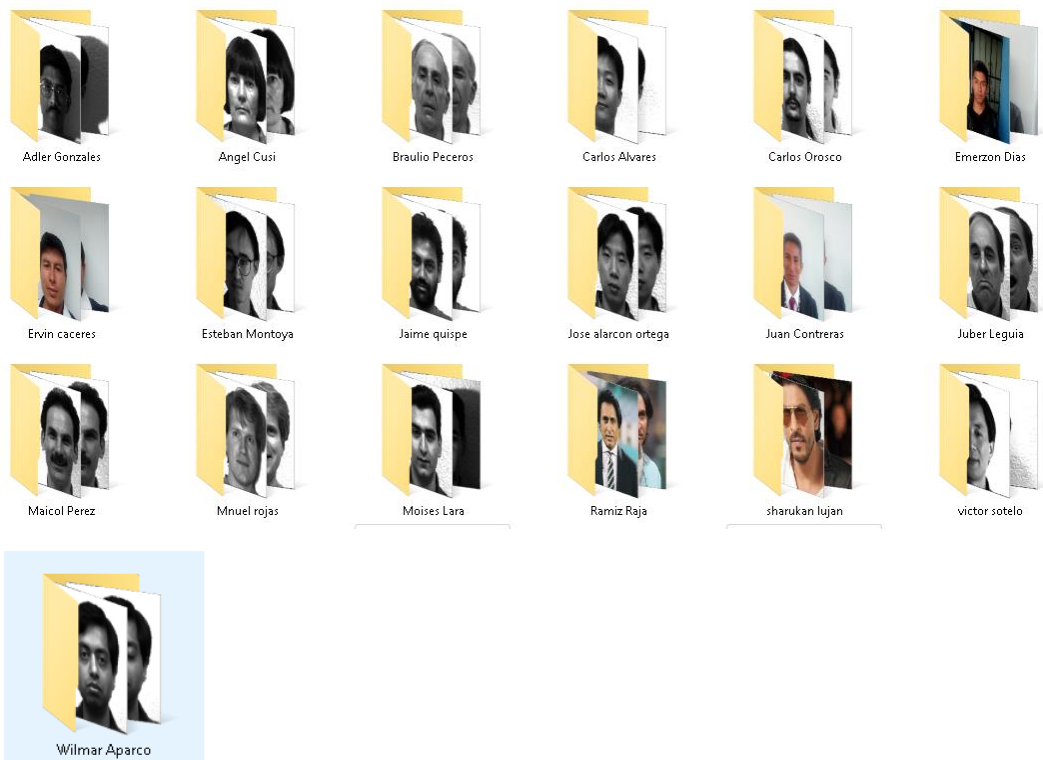
El sistema confundió al sujeto Ervin Lewis con Manuel rojas, esto aplicado a la seguridad de identificar a personas con antecedentes es muy grave ya que el sistema debe ser confiable.



Reporte Final de la base de datos de las personas escaneadas con OpenCV.


Anexo F: Sujetos de prueba para reconocimiento Facial.

En cada carpeta existen 3 fotografías que fueron subidas a la base de datos de la policía y uno para realizar las pruebas que es diferentes a los 3 ya subidos



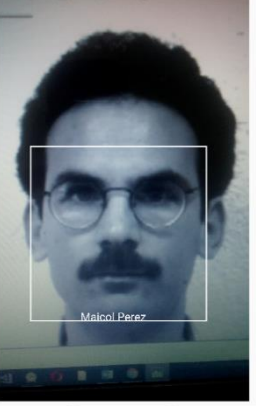
Como se observa las personas de prueba expresan diferentes estados emocionales alegre, triste, llorando, cerrando los ojos, con anteojos entre otros objetos

9:25 a.m. 10:43 a.m.




Nombre: Emerzon Diaz
 Alias: El chato
 Lugar de requisitoria: Andahuaylas - Andahuaylas - Apurimac
 Delito(s): Violación
 Sexo: Masculino
 Estado: Requisitoriado

REPORTAR




DETECTAR ROSTRO IDENTIFICAR PERSONA REINICIAR PROCESO

9:20 a.m. 10:45 a.m. 9:19 a.m.




Nombre: Maicol Perez
 Alias: loco del acha
 Lugar de requisitoria: Andahuaylas - Andahuaylas - Apurimac
 Delito(s): Violación, Violación
 Sexo: Masculino
 Estado: Detenido

REPORTAR



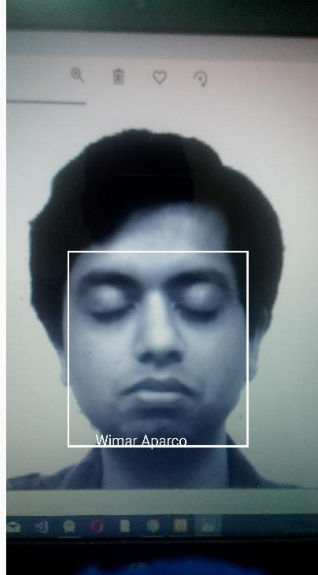
DETECTAR ROSTRO IDENTIFICAR PERSONA REINICIAR PROCESO




Nombre: Victor Sotelo
 Alias: diablo
 Lugar de requisitoria: Andahuaylas - Andahuaylas - Apurimac
 Delito(s): Violación
 Sexo: Masculino
 Estado: Requisitoriado

REPORTAR

9:14 a.m. 9:14 a.m. 9:11 a.m.




DETECTAR ROSTRO IDENTIFICAR PERSONA REINICIAR PROCESO



Nombre: Wimar Aparco
 Alias: Desconocido
 Lugar de requisitoria: Santa Maria de chimo - Andahuaylas - Apurimac
 Delito(s): Violación
 Sexo: Masculino
 Estado: Requisitoriado

REPORTAR



DETECTAR ROSTRO IDENTIFICAR PERSONA REINICIAR PROCESO

Finalizo Deteccion. 1 cara(s) detectadas

9:12 a.m. 9:08 a.m. 9:09 a.m.

Nombre: Moises Lara
Alias: maiquito
Lugar de requisitoria: Pampachiri -Andahuaylas - Apurimac
Delito(s): Violación
Sexo: Masculino
Estado: Requisitoriado

Nombre: Juber Leguia
Alias: Narizon
Lugar de requisitoria: Andahuaylas -Andahuaylas - Apurimac
Delito(s): Violación
Sexo: Masculino
Estado: Requisitoriado

REPORTAR

DETECTAR ROSTRO IDENTIFICAR PERSONA REINICIAR PROCESO

REPORTAR

10:46 a.m. 9:07 a.m. 9:00 a.m.

Nombre: Jose Alarcon Ortega
Alias: LLoron
Lugar de requisitoria: Andahuaylas -Andahuaylas - Apurimac
Delito(s): Violación
Sexo: Masculino
Estado: Requisitoriado

Nombre: Esteban Montoya

REPORTAR

DETECTAR ROSTRO IDENTIFICAR PERSONA REINICIAR PROCESO

DETECTAR ROSTRO IDENTIFICAR PERSONA REINICIAR PROCESO

9:01 a.m. 8:55 a.m. 8:54 a.m.

Nombre: Esteban Montoya
Alias: coyote
Lugar de requisitoria: Lima - Lima -lima
Delito(s): Violación
Sexo: Masculino
Estado: Requisitoriado

Nombre: Carlos Orosco
Alias: zorro
Lugar de requisitoria: Andahuaylas -Andahuaylas - Apurimac
Delito(s): Violación
Sexo: Masculino
Estado: Requisitoriado

REPORTAR

Finalizo Deteccion. 1 cara(s) detectadas

DETECTAR ROSTRO IDENTIFICAR PERSONA REINICIAR PROCESO

REPORTAR

8:52 a.m. 8:52 a.m. 10:01 p.m.

Nombre: Carlos Alvares
 Alias: el cojo
 Lugar de requisitoria: Andahuaylas -Andahuaylas - Apurimac
 Delito(s): Violación, Violación, Violación
 Sexo: Masculino
 Estado: Requisitoriado

Finalizo Deteccion. 1 cara(s) detectadas

DETECTAR ROSTRO IDENTIFICAR PERSONA REINICIAR PROCESO

REPORTAR

DETECTAR ROSTRO IDENTIFICAR PERSONA REINICIAR PROCESO

10:35 a.m. 9:58 p.m. 10:38 a.m.

Nombre: Sharucan Lujan
 Alias: indio loco
 Lugar de requisitoria: San Borja-Lima - Lima
 Delito(s): Violación
 Sexo: Masculino
 Estado: Requisitoriado

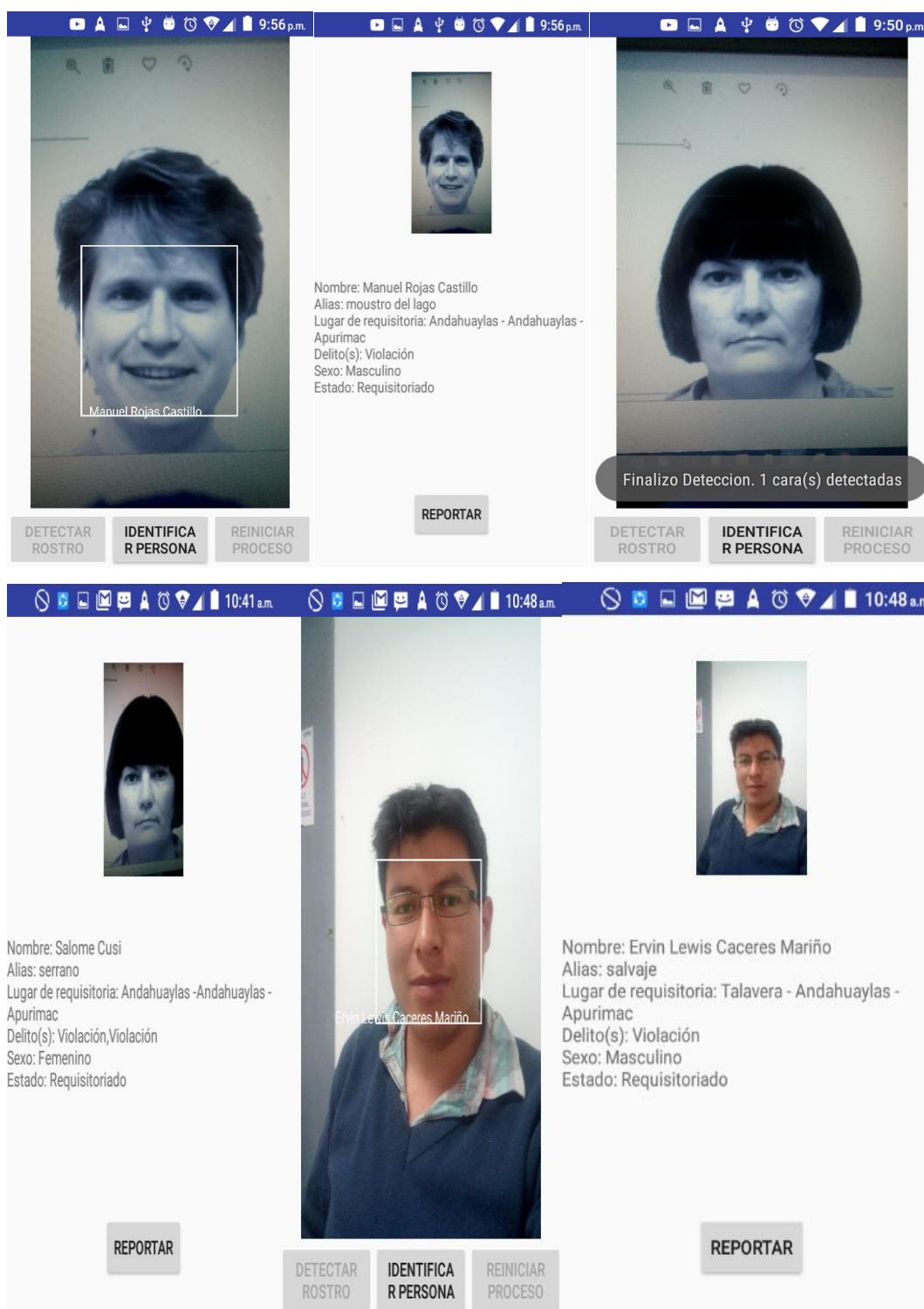
Nombre: Ramiz Raja
 Alias: loco flaco
 Lugar de requisitoria: Talavera -Andahuaylas - Apurimac
 Delito(s): Violación, Violación
 Sexo: Masculino
 Estado: Requisitoriado

Finalizo Deteccion. 1 cara(s) detectadas

REPORTAR

DETECTAR ROSTRO IDENTIFICAR PERSONA REINICIAR PROCESO

REPORTAR



Reporte final de la base de datos después de las pruebas en formato pdf.

Las 20 personas fueron reconocidas por el prototipo de reconocimiento facial y actualizo la base de datos a todas ellas como detenidas.

Base de datos: recfacpolicial, Tabla: persona, Propósito: Volcado de datos

Id_Persona	Id_Sexo	Nombre	Alias	Lugar	Estado_Persona	Estado
46526534	2	Ervin Lewis Caceres Mariño	salvaje	Talavera - Andahuaylas - Apurimac	Detenido	A
46582998	2	Manuel Rojas Castillo	moustro del lago	Andahuaylas - Andahuaylas - Apurimac	Detenido	A
70401234	2	Victor Sotelo	diablo	Andahuaylas - Andahuaylas - Apurimac	Detenido	A
70404615	2	Sharucan Lujan	indio loco	San Borja-Lima - Lima	Detenido	A
72073288	2	Juber Leguia	Narizon	Andahuaylas - Andahuaylas - Apurimac	Detenido	A
72257051	2	Braulio Peceros	mountró de los andes	Andahuaylas - Andahuaylas - Apurimac	Detenido	A
72780799	2	Jose Alarcon Ortega	LLoron	Andahuaylas - Andahuaylas - Apurimac	Detenido	A
72872424	2	Esteban Montoya	coyote	Lima - Lima - Lima	Detenido	A
73127000	1	Salome Cusi	serrano	Andahuaylas - Andahuaylas - Apurimac	Detenido	A
74496511	2	Juan Contreras	Maricon	San Juan Del Urigancho - Lima - Lima	Detenido	A
74984523	2	Wimar Aparco	Desconocido	Sanra Maria de chicmo - Andahuaylas - Apurimac	Detenido	A
76056640	2	Adler Gonzales	maton	Andahuaylas - Andahuaylas - Apurimac	Detenido	A
76130114	2	Jaime Quispe		Abancay - Abancay - Apurimac	Detenido	A
76209968	2	Carlos Alvares	el cojo	Andahuaylas - Andahuaylas - Apurimac	Detenido	A
76329709	2	Moises Lara	maiquito	Pampachiri - Andahuaylas - Apurimac	Detenido	A
76647320	2	Carlos Orozco	zorro	Andahuaylas - Andahuaylas - Apurimac	Detenido	A
76818748	2	Maicol Perez Huayta	casador	Abancay - Abancay - Apurimac	Detenido	A
77200935	2	Ramiz Raja	loco flaco	Talavera - Andahuaylas - Apurimac	Detenido	A
77346301	2	Maicol Perez	loco del acha	Andahuaylas - Andahuaylas - Apurimac	Detenido	A
77917148	2	Emerson Diaz	El chato	Andahuaylas - Andahuaylas - Apurimac	Detenido	A

Anexo G: Código Fuente

Logueo al sistema de Reconocimiento facial

```
public class LoginActivity extends AppCompatActivity {

    TextInputLayout tilUsuario, tilContrasenia;
    TextInputEditText txtUsuario, txtContrasenia;
    Button btnLogin;
    String JsonURL;
    Context context;
    String sUsuario, sContrasenia;
    //REALIZAMOS LA CONSULTA SI SE LOGRO CARGAR CORRECTAMENTE LA CAMARA
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);

        if (ContextCompat.checkSelfPermission(LoginActivity.this,
Manifest.permission.WRITE_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED &&
ActivityCompat.checkSelfPermission(LoginActivity.this,
Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(LoginActivity.this, new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE,
Manifest.permission.CAMERA}, 1000);
        }

        context = this;
        tilUsuario = (TextInputLayout) findViewById(R.id.til_Usuario);
        txtUsuario = (TextInputEditText) findViewById(R.id.txtUsuario);
        tilContrasenia = (TextInputLayout)
findViewById(R.id.til_Contrasenia);
        txtContrasenia = (TextInputEditText)
findViewById(R.id.txtContrasenia);
        btnLogin = (Button) findViewById(R.id.btnLogin);

        btnLogin.setOnClickListener(new View.OnClickListener() {
            @Override
            //INGRESAMOS A LA BASE DE DATOS MEDIANTE WEB SERVICES
            public void onClick(View view) {
                sUsuario = txtUsuario.getText().toString();
                sContrasenia = txtContrasenia.getText().toString();
                JsonURL = "http://172.16.21.175:8080/RecFacPolicial/verificar_login.php";
                JsonURL = JsonURL + "?Usuario=" + sUsuario + "&Contrasenia="
+ sContrasenia;

                RequestQueue requestQueue =
Volley.newRequestQueue(context);

                JsonObjectRequest obreq = new
JsonObjectRequest(Request.Method.GET, JsonURL, null,
new Response.Listener<JSONObject>() {
                    @Override
                    public void onResponse(JSONObject response) {
                        try
                        {
                            if(response.getString("estado") == null
||
```

```

response.getString("estado").equals("2") ||
response.getString("estado").equals("-1") ||
response.getString("estado").equals("3"))
    {
        Toast.makeText(context, "Usuario no
esta registrado", Toast.LENGTH_LONG).show();
        tilUsuario.setError("Error de
ingreso Usuario");
        tilContrasenia.setError("Error de
ingreso Contraseña");
    }
    else
    {
        Intent
intent = new Intent(LoginActivity.this, FotoActivity.class);
        startActivity(intent);
        finish();
    }
}
catch (JSONException e)
{
    e.printStackTrace();
}
},
new Response.ErrorListener() {
@Override
public void onErrorResponse(VolleyError error) {
    Log.e("Volley", "Error");
    Toast.makeText(context, "Error de
Conexión", Toast.LENGTH_LONG).show();
}
}
);

requestQueue.add(obreq);
});
}
}
}

```

Activar cámara

```

public class FotoActivity extends AppCompatActivity {

    static final int REQUEST_TAKE_PHOTO = 1;
    String mCurrentPhotoPath;
    static final int REQUEST_IMAGE_CAPTURE = 1;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_foto);
    }
}

```

```

        if (ContextCompat.checkSelfPermission(FotoActivity.this,
Manifest.permission.WRITE_EXTERNAL_STORAGE) !=
PackageManager.PERMISSION_GRANTED
ActivityCompat.checkSelfPermission(FotoActivity.this,
Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
            ActivityCompat.requestPermissions(FotoActivity.this,
new
String[]{Manifest.permission.WRITE_EXTERNAL_STORAGE,
Manifest.permission.CAMERA}, 1000);
        }

        Intent takePictureIntent =
new
Intent(MediaStore.ACTION_IMAGE_CAPTURE);
        // CONSULTA SI HAY UNA ACTIVIDAD DE CÁMARA PARA MANEJAR LA INTENCIÓN.
        if (takePictureIntent.resolveActivity(getPackageManager()) != null)
{
            // CREA EL ARCHIVO DONDE DEBERÍA IR LA FOTO.
            File photoFile = null;
            try {
                photoFile = createImageFile();
            } catch (IOException ex) {

            }
            // CONTINUAR SOLO SI EL ARCHIVO FUE CREADO EXITOSAMENTE
            if (photoFile != null) {
                Uri photoURI =
new
FileProvider.getUriForFile(this, "com.example.android.fileprovider",
photoFile);
                takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT,
photoURI);
                startActivityForResult(takePictureIntent,
REQUEST_TAKE_PHOTO);
            }
        }

        private File createImageFile() throws IOException {
            // CREAR UN NOMBRE DE ARCHIVO DE IMAGEN
            String timeStamp =
new
SimpleDateFormat("yyyyMMdd_HH:mm:ss").format(new Date());
            String imageFileName = "Backup_" + timeStamp + "_";
            File storageDir =
new
getExternalFilesDir(Environment.DIRECTORY_PICTURES);
            File image = File.createTempFile(imageFileName, ".jpg", storageDir);

            mCurrentPhotoPath = image.getAbsolutePath();
            return image;
        }

        @Override
        protected void onActivityResult(int requestCode, int resultCode, Intent
data) {
            if (requestCode == REQUEST_IMAGE_CAPTURE && resultCode == RESULT_OK)
{
                File mFile = new File(mCurrentPhotoPath);
                Intent intent = new Intent(this, ProcesoActivity.class);
                intent.putExtra("img", mFile);
                startActivity(intent);
            }
        }
    }
}

```

Menu principal.

```
public class ProcesoActivity extends AppCompatActivity {

    private File mFile;
    private Button btn_Detectar, btn_Identificar, btn_Reiniciar;
    private ImageView img_Foto;
    private FileInputStream fis = null;
    private Bitmap bitmap;
    private ByteArrayOutputStream output;
    private ByteArrayInputStream inputStream;
    // CREAMOS UNA VARIABLE CONTENEDOR DE LA CLAVE GRATUITA DE FACE API
    private FaceServiceClient faceServiceClient = new
FaceServiceRestClient("https://southcentralus.api.cognitive.microsoft.com/f
ace/v1.0",
        "e279abfb2ed745f28b43d85337986248");
    private final String personGroupId = "bddenlucuentes";
    private Face[] facesDetected;
    private UUID[] mFaceId = null;
    private Face face;
    private String[] persona = null;
    //VENTANA PRINCIPAL DE LOS 3 MOMENTOS DETECCION, RECONOCIMIENTO, REPORTE.
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_proceso);

        Intent intent = getIntent();
        mFile = (File) intent.getSerializableExtra("img");

        btn_Detectar = (Button) findViewById(R.id.btnDetectar);
        btn_Identificar = (Button) findViewById(R.id.btnIdentificar);
        btn_Reiniciar = (Button) findViewById(R.id.btnReiniciar);
        img_Foto = (ImageView) findViewById(R.id.imgFoto);

        procesar_imagen(mFile);
        img_Foto.setImageBitmap(bitmap);

        btn_Detectar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                new DetectionTask().execute(inputStream);

                /*Intent intent = new Intent(ProcesoActivity.this,
IdentificacionActivity.class);
                intent.putExtra("img", mFile);
                intent.putExtra("identificacion", "123456787");
                startActivity(intent);
                finish();*/
            }
        });

        btn_Identificar.setOnClickListener(new View.OnClickListener()
        {
            @Override
            public void onClick(View view) {
                mFaceId = new UUID[facesDetected.length];

                for(int i = 0; i < facesDetected.length; i++)
```

```

        {
            mFaceId[i] = facesDetected[i].faceId;
        }

        new IdentificationTask(personGroupId).execute(mFaceId);
    }
});

btn_Reiniciar.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Intent intent = new Intent(ProcesoActivity.this,
FotoActivity.class);
        startActivity(intent);
        finish();
    }
});
}

private void procesar_imagen(File lFile) {
    try {
        fis = new FileInputStream(mFile);
    } catch (Exception e) {
        e.printStackTrace();
    }

    bitmap = BitmapFactory.decodeStream(fis);
    output = new ByteArrayOutputStream();
    bitmap.compress(Bitmap.CompressFormat.JPEG, 100, output);
    inputStream = new ByteArrayInputStream(output.toByteArray());
}

private class DetectionTask extends AsyncTask<InputStream, String,
Face[]> {
    private ProgressDialog mDialog = new
ProgressDialog(ProcesoActivity.this);

    @Override
    protected Face[] doInBackground(InputStream... params) {
        try {
            publishProgress("Detectando...");
            Face[] results = faceServiceClient.detect(params[0],
                true, // returnFaceId
                false, // returnFaceLandmarks
                null // returnFaceAttributes:
UNA CADENA COMO "EDAD, GÉNERO"
            );

            if (results == null) {
                publishProgress("Finalizo Deteccion. No detecto nada");
                return null;
            } else {
                publishProgress(String.format("Finalizo Deteccion. %d
cara(s) detectadas", results.length));
                return results;
            }
        } catch (Exception e) {
            publishProgress("Fallo Deteccion");
            Log.e("Error al detectar", e.toString());
            return null;
        }
    }
}
}

```

```

    }

    @Override
    protected void onPreExecute() {
        //TODO: show progress dialog
        mDialog.show();
    }

    @Override
    protected void onProgressUpdate(String... progress) {
        //TODO: update progress
        mDialog.setMessage(progress[0]);
    }

    @Override
    protected void onPostExecute(Face[] faces) {
        //TODO: update face frames
        mDialog.dismiss();

        facesDetected = faces;

        if (faces == null || faces.length == 0) {
            Toast.makeText(getApplicationContext(),
                String.format("No detecto una cara en la imagen,
volver a reiniciar el proceso"),
                Toast.LENGTH_LONG).show();
            btn_Reiniciar.setEnabled(true);
            btn_Detectar.setEnabled(false);
        } else {
            Toast.makeText(getApplicationContext(),
                String.format("Finalizo Deteccion. %d cara(s)
detectadas", faces.length),
                Toast.LENGTH_LONG).show();
            btn_Identificar.setEnabled(true);
            btn_Detectar.setEnabled(false);
        }
    }
}

private class IdentificationTask extends AsyncTask<UUID, String,
IdentifyResult[]> {

    private String personGroupId;
    private ProgressDialog mDialog = new
ProgressDialog(ProcesoActivity.this);

    public IdentificationTask(String personGroupId) {
        this.personGroupId = personGroupId;
    }

    @Override
    protected IdentifyResult[] doInBackground(UUID... uuids) {
        try {
            publishProgress("Obteniendo el estatus del grupo de
personas...");

            TrainingStatus trainingStatus =
faceServiceClient.getPersonGroupTrainingStatus(this.personGroupId);

            if (trainingStatus.status !=
TrainingStatus.Status.Succeeded) {

```



```

        publishProgress("Estado training del grupo de personas
es " + trainingStatus.status);
        return null;
    }

    publishProgress("Identificando...");

    return faceServiceClient.identity(personGroupId, //person
group id
        uuids, //face ids
        1 //max number of candidates return
    );

} catch (Exception ex) {
    publishProgress("Fallo al Obtener estatus del grupo");
    Log.e("Error al detectar", ex.toString());
    return null;
}
}

@Override
protected void onPreExecute() {
    mDialog.show();
}

@Override
protected void onPostExecute(IdentifyResult[] identifyResults) {
    mDialog.dismiss();

    if (identifyResults[0].candidates.isEmpty()) {
//RECORDAR QUE ESTE CODIGO ES SI ES POR ERROR DE NO PODER IDENTIFICAR
        Intent intent = new Intent(ProcesoActivity.this,
IdentificacionActivity.class);
        intent.putExtra("img", mFile);
        startActivity(intent);
        finish();
    } else {
        for (IdentifyResult identifyResult : identifyResults) {
            new
PersonDetectionTask(personGroupId).execute(identifyResult.candidates.get(0)
.personId);
        }
    }
}

@Override
protected void onProgressUpdate(String... values) {
    mDialog.setMessage(values[0]);
}
}

private class PersonDetectionTask extends AsyncTask<UUID, String,
Person> {

    private String personGroupId;
    private ProgressDialog mDialog = new
ProgressDialog(ProcesoActivity.this);

    public PersonDetectionTask(String personGroupId) {
        this.personGroupId = personGroupId;
    }
}

```

```

@Override
protected Person doInBackground(UUID... uuids) {
    try
    {
        publishProgress("Obteniendo estatus del person group...");

        return faceServiceClient.getPerson(personGroupId,
uuids[0]);
    }
    catch (Exception ex)
    {
        publishProgress("Fallo al detectar a la persona");
        Log.e("Error al detectar", ex.toString());
        return null;
    }
}

@Override
protected void onPreExecute() {
    mDialog.show();
}

@Override
protected void onPostExecute(Person person) {
    mDialog.dismiss();

    persona = person.name.split("-");

    //img_Foto.setImageBitmap(drawFaceRectangleOnBitmap(bitmap,
facesDetected, person.name));
    img_Foto.setImageBitmap(drawFaceRectangleOnBitmap(bitmap,
facesDetected, persona[0]));

    Intent intent = new Intent(ProcesoActivity.this,
IdentificacionActivity.class);
    intent.putExtra("img", mFile);
    intent.putExtra("identificacion", persona[1]);
    startActivity(intent);
    finish();
}

@Override
protected void onProgressUpdate(String... values) {
    mDialog.setMessage(values[0]);
}
}

private Bitmap drawFaceRectangleOnBitmap(Bitmap bitmap, Face[]
facesDetected, String name) {

    Bitmap lbitmap = bitmap.copy(Bitmap.Config.ARGB_8888, true);
    Canvas canvas = new Canvas(lbitmap);

    //Rectangle
    Paint paint = new Paint();
    paint.setAntiAlias(true);
    paint.setStyle(Paint.Style.STROKE);
    paint.setColor(Color.WHITE);
    paint.setStrokeWidth(12);

```

```

    if(facesDetected != null)
    {
        for(Face face: facesDetected)
        {
            FaceRectangle faceRectangle = face.faceRectangle;
            canvas.drawRect(faceRectangle.left,
                faceRectangle.top,
                faceRectangle.left + faceRectangle.width,
                faceRectangle.top + faceRectangle.height,
                paint);
            drawTextOnCanvas(canvas,
                100,
                ((faceRectangle.left+faceRectangle.width)/2)+100,
                (faceRectangle.top+faceRectangle.height)+50,
                Color.WHITE,
                name);
        }
    }

    return lbitmap;
}

private void drawTextOnCanvas(Canvas canvas, int textSize, int x, int y,
int color, String name) {
    Paint paint = new Paint();
    paint.setAntiAlias(true);
    paint.setStyle(Paint.Style.FILL);
    paint.setColor(color);
    paint.setTextSize(textSize);

    float textWidth = paint.measureText(name);

    canvas.drawText(name, x-(textWidth/2), y-(textSize/2), paint);
}
}

```

Reporte de la persona identificada y detención

```

public class IdentificacionActivity extends AppCompatActivity {

    private File mFile;
    private String sIdentificacion = null, sEstado = null, sRespuesta = null;
    private Button btn_reportar;
    private ImageView imgFotof;
    private TextView txv_informacion;
    private FileInputStream fis = null;
    private Bitmap bitmap;
    private ByteArrayOutputStream output;
    private ByteArrayInputStream inputStream;
    String JsonURL;
    Context context;
    private static final ArrayList<String> antecedentes = new ArrayList<>(),
    persona = new ArrayList<>();
    //CONSULTA A LA BD SI LA PERSONA TIENE ANTECEDENTES POR VIOLACION SEXUAL Y
    PRESENTA REQUISITORIA
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_identificacion);
    }
}

```



```

persona.add(jsonObject.getString("Alias"));
persona.add(jsonObject.getString("Lugar"));
persona.add(jsonObject.getString("Estado_Persona"));
antecedentes.add(jsonObject.getString("antecedentes"));
}
else
{
antecedentes.add(jsonObject.getString("antecedentes"));
}

}

String antecedente = null;

for(int i = 0; i < antecedentes.size(); i++)
{
if(i == 0)
antecedente = antecedentes.get(i);
else
antecedente = antecedente + "," + antecedentes.get(i);
}

String parrafo = "Nombre: " + persona.get(0) + "\n" +
"Alias: " + persona.get(2) + "\n" +
"Lugar de requisitoria: " + persona.get(3) + "\n" +
"Delito(s): " + antecedente + "\n" +
"Sexo: " + persona.get(1) + "\n" +
"Estado: " + persona.get(4);

txv_informacion.setText(parrafo);
}
else

```

```

{
sRespuesta = response.getString("mensaje");
}

btn_reportar.setEnabled(true);
} catch (JSONException e) {
e.printStackTrace();
}
}
},

new Response.ErrorListener() {
@Override

//MANEJA LOS ERRORES QUE SE PRODUCEN
public void onErrorResponse(VolleyError error) {

Log.e("Volley", error.getMessage());

Toast.makeText(context, "Volver a intentar", Toast.LENGTH_LONG).show();

btn_reportar.setText("Volver");

btn_reportar.setEnabled(true);

btn_reportar.setOnClickListener(new View.OnClickListener() {
@Override

public void onClick(View view) {

Intent intent = new Intent(IdentificacionActivity.this, FotoActivity.class);

startActivity(intent);

finish();

}

});

});

requestQueue.add(jsonObjectRequest);

btn_reportar.setOnClickListener(new View.OnClickListener() {
//ACTUALIZA LA BASE DE DATOS SI SE REPORTA COMO DETENIDO POR LA WEB SERVICIOS.
@Override
public void onClick(View view) {
JsonURL
"http://172.16.21.175:8080/RecFacPolicial/update_persona.php";
=

```



```

        btn_reportar.setText("Volver");
        btn_reportar.setEnabled(true);
        btn_reportar.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Intent intent = new Intent(IdentificacionActivity.this,
FotoActivity.class);
                startActivity(intent);
                finish();
            }
        });
    }
}

private void procesar_imagen(File lFile) {
    try {
        fis = new FileInputStream(mFile);
    } catch (Exception e) {
        e.printStackTrace();
    }

    bitmap = BitmapFactory.decodeStream(fis);
    output = new ByteArrayOutputStream();
    bitmap.compress(Bitmap.CompressFormat.JPEG, 100, output);
    inputStream = new ByteArrayInputStream(output.toByteArray());
}
}
}

```

Entrenar al Face API

```
namespace CargaAzureMySQL
```

```

{
    class Program
    {
        public static bool creaGrupo = false;
        public static bool creaPersona = false;
        public static bool eliminaGrupo = false;
        public static bool tStatus = false;
        public static bool reconocimientoPersona = false;
        // CLAVE PARA ACSEDER AL FACE API
        FaceServiceClient faceServiceClient = new
FaceServiceClient("3aec41ef60c14f6eae8e9693d3a7a54f",
"https://southcentralus.api.cognitive.microsoft.com/face/v1.0");

        public async void CreatePersonGroup(string personGroupId, string
personGroupName)
        {
            try
            {
                await
faceServiceClient.CreatePersonGroupAsync(personGroupId, personGroupName);
                Console.WriteLine("Se creo el personGroupId");
                creaGrupo = true;
            }
            catch(Exception ex)
            {

```



```

        Console.WriteLine("Error Create Person Group\n" +
ex.Message);
    }
}

public async void DeletePersonGroup(string personGroupId)
{
    try
    {
        await
faceServiceClient.DeletePersonGroupAsync(personGroupId);
        Console.WriteLine("Se elimino el personGroupId");
        eliminaGrupo = true;
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error Delete Person Group\n" +
ex.Message);
    }
}

public async void AddPersonToGroup(string personGroupId, string
name, string pathImage)
{
    try
    {
        await faceServiceClient.GetPersonGroupAsync(personGroupId);
        CreatePersonResult person = await
faceServiceClient.CreatePersonAsync(personGroupId, name);

        DetectFaceAndRegister(personGroupId, person, pathImage);
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error Add Person to Group\n" +
ex.Message);
    }
}

public async void AddPersonToGroup(string personGroupId, string
name, Stream imagenS)
{
    try
    {
        await faceServiceClient.GetPersonGroupAsync(personGroupId);

        CreatePersonResult person = await
faceServiceClient.CreatePersonAsync(personGroupId, name);

        DetectFaceAndRegister(personGroupId, person, imagenS);
        Console.WriteLine("Se adiciono la persona en el grupo");
        creaPersona = true;
    }
    catch (Exception ex)
    {
        Console.WriteLine("Error Add Person to Group\n" +
ex.Message);
    }
}
}

```

```

        private async void DetectFaceAndRegister(string personGroupId,
        CreatePersonResult person, string pathImage)
        {
            foreach (var imgPath in Directory.GetFiles(pathImage, "*.jpg"))
            {
                using (Stream a = File.OpenRead(imgPath))
                {
                    await
                    faceServiceClient.AddPersonFaceAsync(personGroupId, person.PersonId, a);
                }
            }
        }

        private async void DetectFaceAndRegister(string personGroupId,
        CreatePersonResult person, Stream imagenS)
        {
            try
            {
                await faceServiceClient.AddPersonFaceAsync(personGroupId,
                person.PersonId, imagenS);
                Console.WriteLine("Se detecto y se registro la persona en el
                grupo");
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
            }
        }

        public async void TrainingAI(string personGroupId)
        {
            try
            {
                await
                faceServiceClient.TrainPersonGroupAsync(personGroupId);
                TrainingStatus trainingStatus = null;

                while (true)
                {
                    trainingStatus =
                    await
                    faceServiceClient.GetPersonGroupTrainingStatusAsync(personGroupId);

                    if (trainingStatus.Status != Status.Running)
                        break;

                    await Task.Delay(5000);
                }
                tStatus = true;
                Console.WriteLine("Training AI Completed");
            }
            catch (Exception ex)
            {
                Console.WriteLine(ex.Message);
            }
        }

        public async void RecognitionFace(string personGroupId, string
        imgPath)
        {
            using (Stream s = File.OpenRead(imgPath))

```

```

        {
            var faces = await faceServiceClient.DetectAsync(s);
            var faceIds = faces.Select(face => face.FaceId).ToArray();

            try
            {
                var results = await
                faceServiceClient.IdentifyAsync(personGroupId, faceIds);

                foreach (var identifyResult in results)
                {
                    Console.WriteLine($"Resultado de caras:
                    {identifyResult.FaceId}");
                    if (identifyResult.Candidates.Length == 0)
                        Console.WriteLine("No se identifico");
                    else
                    {
                        //OBTENER TOP 1 ENTRE TODOS LOS CANDIDATOS DEVUELTOS
                        var candidateId =
                        identifyResult.Candidates[0].PersonId;
                        var person = await
                        faceServiceClient.GetPersonAsync(personGroupId, candidateId);
                        Console.WriteLine($"Identificado es
                        {person.Name}");
                    }
                }
                reconocimientoPersona = true;
            }
            catch (Exception ex)
            {
                Console.WriteLine("Error reconocimiento cara: " +
                ex.Message);
            }
        }

        public void crearGrupo()
        {
            //SE CREA EL GRUPO PARA ADICIONAR LAS PERSONAS Y SUS ROSTROS
            CreatePersonGroup("bddelincuentes", "Total Antecedentes");
        }

        public void deleteGrupo()
        {
            //PARA ELIMINAR EL GRUPO Y VOLVER A REALIZAR LA PRUEBA
            DeletePersonGroup("bddelincuentes");
        }

        public void adicionaPersona()
        {
            /** SE LEE DE LA BASE DE DATOS LAS 3 FOTOS POR PERSONAS PARA PODER IDENTIFICAR
            QUE ES LA CORRECTA
            * COMO OBSERVACION PARA PRUEBAS SE ESTA UTILIZANDO PERSONGROUP, PERO PARA
            CANTIDADES MAYORES SE DEBE USAR LARGEPERSONGROUP
            **/
            string nombre;
            byte[] imagen1;
            byte[] imagen2;
            byte[] imagen3;

```

```

        using ( MySqlConnection con = new
        MySqlConnection("Server=localhost; database=recfacpolicial; UID=root;
        password=;SSL Mode=None"))
        {
            try
            {
                con.Open();
            }
            catch ( MySqlException ex)
            {
                switch (ex.Number)
                {
                    case 0:
                        Console.WriteLine("Cannot connect to server.
Contact administrator");
                        break;

                    case 1045:
                        Console.WriteLine("Invalid username/password,
please try again");
                        break;
                }
            }

            String sql = "select CONCAT(Nombre, '-', Id_Persona) as
nombre, " +
                " Imagen1, " +
                " Imagen2, " +
                " Imagen3 " +
                "from persona " +
                "where Estado = 'A'";

            // CREAR COMANDO
            MySqlCommand cmd = new MySqlCommand(sql, con);
            // CREAR UN LECTOR DE DATOS Y EJECUTAR EL COMANDO.
            MySqlDataReader reader = cmd.ExecuteReader();

            while (reader.Read())
            {
                nombre = reader.GetString(0);
                imagen1 = (byte[]) reader["Imagen1"];
                imagen2 = (byte[]) reader["Imagen2"];
                imagen3 = (byte[]) reader["Imagen3"];

                Stream imagen1S = new MemoryStream(imagen1);
                Stream imagen2S = new MemoryStream(imagen2);
                Stream imagen3S = new MemoryStream(imagen3);

                for (int i = 0; i < 3; i++)
                {
                    if (i == 0)
                    {
                        AddPersonToGroup("bddenlincuentes", nombre,
imagen1S);
                    }

                    if (i == 1)
                    {
                        AddPersonToGroup("bddenlincuentes", nombre,
imagen2S);
                    }
                }
            }
        }
    }
}

```

```

        }
        if (i == 2)
        {
            AddPersonToGroup("bddelincuentes", nombre,
imagen3S);
        }
    }

    reader.Close();

    con.Dispose();

    try
    {
        con.Close();
    }
    catch (MySqlException ex)
    {
        Console.WriteLine(ex.Message);
    }
}

public void trainingGrupo()
{
    //PARA SABER SI YA SE INTEGRO TODO EL GRUPO DE PERSONAS PARA LA
VERIFICACION
    TrainingAI("bddelincuentes");
}

public void verificarReconocimiento()
{
    //PARA VERIFICACION DE LA IMAGEN CON EL GRUPO CREADO, SI ES
ACERTADO O NO
    RecognitionFace("bddelincuentes", @"C:\ejemplo\personal.jpg");
}

static void Main(string[] args)
{
    bool done = false;

    do
    {
        Console.WriteLine("Selecccione las siguientes opciones en el
orden indicado");
        Console.WriteLine("\t1 -- Crea Grupo");
        Console.WriteLine("\t2 -- Adiciona Persona al Grupo");
        Console.WriteLine("\t3 -- Verifica Sincronizacion");
        Console.WriteLine("\t4 -- Verifica Reconocimiento");
        Console.WriteLine("\t5 -- Elimina Grupo");
        Console.WriteLine("\t0 -- Salir");
        Console.WriteLine("Ingrese la opcion (0 para salir): ");
        string strSeleccion = Console.ReadLine();
        int iSeleccion;

        try
        {
            iSeleccion = int.Parse(strSeleccion);
        }
    }
}

```

