



UNIVERSIDAD TÉCNICA PARTICULAR DE LOJA
La Universidad Católica de Loja

ÁREA TÉCNICA

TÍTULO DE INGENIERO EN INFORMÁTICA

Uso de algoritmos de visión artificial para el diseño e implementación de un sistema de control de acceso a través de reconocimiento facial

TRABAJO DE TITULACIÓN.

AUTOR: Solís Alulima, Darío Javier

DIRECTOR: Barba Guamán, Luis Rodrigo, Mgtr.

CENTRO UNIVERSITARIO QUITO

2017



Esta versión digital, ha sido acreditada bajo la licencia Creative Commons 4.0, CC BY-NY-SA: Reconocimiento-No comercial-Compartir igual; la cual permite copiar, distribuir y comunicar públicamente la obra, mientras se reconozca la autoría original, no se utilice con fines comerciales y se permiten obras derivadas, siempre que mantenga la misma licencia al ser divulgada. <http://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>

Septiembre, 2017

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN

Magister.

Luis Rodrigo Barba Guamán.

DOCENTE DE LA TITULACIÓN

De mi consideración:

El presente trabajo de titulación: Uso de algoritmos de visión artificial para el diseño e implementación de un sistema de control de acceso a través de reconocimiento facial, realizado por Darío Javier Solís Alulima, ha sido orientado y revisado durante su ejecución, por cuanto se aprueba la presentación del mismo.

Loja, septiembre de 2017

f)

DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS

“Yo Darío Javier Solís Alulima declaro ser autor del presente trabajo de titulación: **Uso de algoritmos de visión artificial para el diseño e implementación de un sistema de control de acceso a través de reconocimiento facial**, de la Titulación de Ingeniero en Informática, siendo el Mgtr. Luis Rodrigo Barba Guamán director del presente trabajo, y eximo expresamente a la Universidad Técnica Particular de Loja y a sus representantes legales de posibles reclamos o acciones legales. Además, certifico que las ideas, conceptos, procedimientos y resultados vertidos en el presente trabajo investigativo, son de mi exclusiva responsabilidad.

Adicionalmente declaro conocer y aceptar la disposición del Art. 88 del Estatuto Orgánico de la Universidad Técnica Particular de Loja que en su parte pertinente textualmente dice: “Forman parte del patrimonio de la Universidad la propiedad intelectual de investigaciones, trabajos científicos o técnicos y tesis de grado o trabajos de titulación que se realicen con el apoyo financiero, académico o institucional (operativo) de la Universidad”

Firma.

Autor: Solís Alulima, Darío Javier

Cédula: 1717186918

DEDICATORIA

Dedicatoria.

A mis padres quienes me apoyaron todo el tiempo y son mi ejemplo de vida, este logro es para ustedes.

A mis hermanos quienes son los mejores amigos que la vida me pudo dar.

A mi esposa Lily quien me apoyó y alentó para continuar, y a quien amo incondicionalmente y amaré siempre.

A mis maestros quienes supieron guiarme en todo este proceso de aprendizaje.

Y a todas mis queridas mascotas Chester, Chavito, Micho, Micha, Peluchito, Tigrecito, Tigre viejo por su cariño en mis momentos de estrés.

Darío.

AGRADECIMIENTOS

El presente trabajo va dirigido con una expresión de gratitud:

A Dios por brindarme la sabiduría, entendimiento y salud para poder culminar satisfactoriamente un meta más en mi vida.

A mis papás y hermanos quienes con su ejemplo y amor forjaron la persona que soy.

A mi esposa quien es mi fortaleza, apoyo incondicional y que con su amor me ayuda a cumplir mis metas.

A la Universidad Técnica Particular de Loja (UTPL), por darme la oportunidad de continuar con mis estudios.

Al Mgtr. Luis Rodrigo Barba Guamán quien supo guiarme satisfactoriamente en el desarrollo del presente trabajo.

Darío.

ÍNDICE DE CONTENIDOS

APROBACIÓN DEL DIRECTOR DEL TRABAJO DE TITULACIÓN.....	II
DECLARACIÓN DE AUTORÍA Y CESIÓN DE DERECHOS.....	III
DEDICATORIA.....	IV
AGRADECIMIENTOS.....	V
ÍNDICE DE CONTENIDOS.....	VI
INDICE DE TABLAS.....	IX
INDICE DE FIGURAS.....	X
RESUMEN.....	1
ABSTRACT.....	2
INTRODUCCIÓN.....	3
FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Herramientas de procesamiento de imágenes.....	6
1.1.1 OpenCV (Open Source Computer Visión).....	6
1.1.2 Sistema de Infrarojos.....	8
1.2 Métodos de autenticación biométricas.....	9
1.2.1 Reconocimiento del Iris.....	9
1.3 Técnicas de reconocimiento facial.....	11
1.3.1 Los sistemas de reconocimiento facial automático.....	11
1.3.2 Reconocimiento facial a partir de imágenes 3D.....	13
1.3.3 Técnicas de adquisición de imágenes 3D.....	13
1.3.4 Técnicas de reconocimiento 3D basadas en rasgos locales.....	16
1.4 Algoritmos de reconocimiento facial.....	18
1.4.1 Algoritmo CamShift.....	18
1.4.2 Algoritmo de alineamiento iterativo de Lucas Y Kanade.....	19
1.4.3 PCA (Análisis de Componentes Principales).....	20
1.4.4 LDA (Análisis Discriminante Lineal).....	20
1.4.5 LPP (Proyecciones de Preservación de la Localidad).....	22

1.4.6 DCT (Transformada Discreta de Coseno).	22
1.4.7 (LBPH) Histogramas de patrones binarios locales.	24
1.5 Visión por computadora.	27
1.6 Lenguajes de Programación.	28
1.6.1 PHP.	28
1.6.2 C#.	29
1.6.3 Python.	29
1.7 Bases de datos.	30
1.7.1 Oracle.	31
1.7.2 MySQL.	31
1.7.3 PostgreSQL.	32
1.7.4 SQLServer.	32
1.8 Metodología de desarrollo Ágiles.	33
1.8.1 Modelo Scrum.	33
1.8.2 Modelo de Programación Extrema (XP).	34
1.9 Entorno de Desarrollo Integrado(IDE).	35
1.9.1 Eclipse.	35
1.9.2 NetBeans.	36
1.9.3 Anaconda.	36
1.10 Trabajos previos de reconocimiento facial.	36
METODOLOGÍA	38
2.1 Principales procesos identificados.	39
2.1. 1 Adicionar usuario.	41
2.1.2 Entrenar motor	42
2.1.3 Reconocimiento facial	43
2.1.4 Borrar Usuario.	44
2.2 Metodología XP como metodología de desarrollo.	45
2.2.1 Planeación - Historias de usuario.	45
2.2.1.1 Historia de usuario Insertar usuario.	45

2.2.1.2 Historia de usuario Eliminar usuario.	46
2.2.1.3 Historia de usuario Reconocimiento facial	46
2.2.1.4 Historia de usuario Entrenar motor.	47
2.3 Planeación – Criterios de aceptación.	48
2.4 Parámetros necesarios para el Reconocimiento Facial.	48
2.4.1 Nivel de confianza de Predicción en la aplicación.	49
2.4.2 Distancia de Reconocimiento.	49
2.4.3 Nivel de Iluminación.	49
PRUEBAS Y RESULTADOS DE FUNCIONAMIENTO	51
3.1 Calibración del Sistema.	52
3.1.1 Nivel de Confianza.	52
3.1.2 Distancias para Reconocimientos.	53
3.1.3 Nivel de Iluminación.	54
3.2 Pruebas - Pruebas de aceptación.	55
3.2.1 Caso de prueba: Adicionar usuario.	55
3.2.2 Caso de prueba: Eliminar usuario.	57
3.2.3 Caso de prueba: Reconocer usuario.	58
3.3 Pruebas y Resultados Obtenidos.	59
3.3.1 Usuarios de Prueba.....	59
3.3.2 Entorno Físico de Pruebas.	60
3.3.3. Pruebas con los parámetros de funcionamiento.	61
3.3.4 Exactitud del sistema ante los diferentes escenarios.....	70
CONCLUSIONES	71
RECOMENDACIONES	72
BIBLIOGRAFÍA.....	73
ANEXOS.....	77
Manual para la realización de las pruebas para usuarios.....	78

INDICE DE TABLAS

Tabla 1. Estructura del Directorio Opencv.....	6
Tabla 2. Comparación de algoritmos de reconocimiento facial	25
Tabla 3.Comparación de lenguajes de programación	29
Tabla 4. Comparación de bases de datos.....	32
Tabla 5. Componentes del esquema de desarrollo	39
Tabla 6. Conjunto de archivos necesarios para el programa.....	40
Tabla 7. HU_Insertar usuario.	45
Tabla 8. HU_Eliminar usuario.....	46
Tabla 9. HU_Reconocimiento facial	46
Tabla 10. HU_Entrenar motor	47
Tabla 11. Requisitos y criterios de aceptación.	48
Tabla 12. Niveles de Luz natural en el día.....	50
Tabla 13. Pruebas de funcionamiento variando los niveles de confianza.....	52
Tabla 14. Distancia para el reconocimiento	53
Tabla 15. Luminosidad baja.....	54
Tabla 16. Luminosidad alta	54
Tabla 17. CP_Adicionar_Usuario.....	55
Tabla 18. CP_Eliminar_usuario	57
Tabla 19. CP_Reconocer_usuario	58
Tabla 20. Registro de algunos Usuarios en pruebas.....	60
Tabla 21. Posibles escenarios para las pruebas	61
Tabla 22. Resultados del Escenario 1.....	62
Tabla 23. Resultados del Escenario 2.....	63
Tabla 24. Resultados del Escenario 3.....	64
Tabla 25. Resultados del Escenario 4.....	65
Tabla 26. Resultados del Escenario 5.....	66
Tabla 27. Resultados del Escenario 6.....	67
Tabla 28. Resultados del Escenario 7.....	68
Tabla 29. Resultados del Escenario 8.....	69

INDICE DE FIGURAS

Figura 1. Comunicación inalámbrica desde diferentes perspectivas	8
Figura 2. Proceso de infrarrojo.....	9
Figura 3. Diagrama representado en bloques del reconocimiento del iris	9
Figura 4.Reconocimiento, autenticación e identificación	10
Figura 5. Elementos básicos para la obtención de la imagen.....	10
Figura 6. Reconocimiento genérico.....	12
Figura 7. Reconocimiento genérico con visión	12
Figura 8. Radar.....	14
Figura 9. Detección de caras a través de la luz.....	15
Figura 10. Puntos esenciales para el reconocimiento facial mediante la luz	15
Figura 11. Reconocimiento con 3 coordenadas	16
Figura 12.Camshift inicial.....	18
Figura 13.Camshift final	19
Figura 14. Datos característicos de la clase LDA.....	21
Figura 15. LPP.....	22
Figura 16. DCT	24
Figura 17. LBP Histograms	25
Figura 18. Componentes de la visión por computadora	27
Figura 19. Ilusión visual	28
Figura 20. Representación de una iteración de Scrum.....	33
Figura 21.Ciclo de Programación Extrema.....	34
Figura 22. Flujo Adicionar usuario.....	41
Figura 23. Flujo Entrenar motor	42
Figura 24. Flujo Reconocimiento Facial	43
Figura 25. Flujo Borrar Usuarios	44
Figura 26. Porcentaje de aciertos y fallas en el nivel de confianza.....	52
Figura 27. Porcentaje de aciertos y falla en la distancia.....	53
Figura 28. Luminosidad baja.....	54
Figura 29. Luminosidad alta.....	55
Figura 30. Entorno físico de trabajo	61
Figura 31. Resultados Escenario 1	62
Figura 32. Resultados Escenario 2	63
Figura 33. Resultados Escenario 3	64
Figura 34. Resultados Escenario 4	65

Figura 35. Resultados Escenario 5	66
Figura 36. Resultados Escenario 6	67
Figura 37. Resultados Escenario 7	68
Figura 38. Resultados Escenario 8	69
Figura 39. Tabla comparativa en los escenarios propuestos.....	70

RESUMEN

Para la realización del trabajo de investigación basado sobre USO DE ALGORITMOS DE VISIÓN ARTIFICIAL PARA EL DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE CONTROL DE ACCESO A TRAVÉS DE RECONOCIMIENTO FACIAL, se realizó el estudio de varios trabajos que hacen uso del mismo en la actualidad. La implementación del mismo se basa en el algoritmo de reconocimiento facial Local Binary Patterns Histograms para dar solución a la problemática planteada en la investigación. Como tecnologías para el desarrollo se utiliza Python como lenguaje de programación, se utiliza OpenCV para el procesamiento de imagen, como entorno de desarrollo integrado Anaconda, por las facilidades que brinda al usarlo con el lenguaje Python. Como metodología que guía el proceso de desarrollo se utiliza XP. Se realizan pruebas de aceptación basada en las funcionalidades de la aplicación las cuales arrojaron resultados satisfactorios.

ABSTRACT

In order to carry out the research work based on the USE OF ARTIFICIAL VISION ALGORITHMS FOR THE DESIGN AND IMPLEMENTATION OF AN ACCESS CONTROL SYSTEM THROUGH FACIAL RECOGNITION, a study was carried out of several works that make use of it today. The implementation of the same is based on the Local Binary Patterns Histograms facial recognition algorithm to solve the problems raised in the research. As development technologies Python is used as a programming language, OpenCV is used for image processing, as an integrated development environment Anaconda, for the facilities it provides when using it with the Python language. As methodology guiding the development process is used XP. Acceptance tests are performed based on the application's functionalities which have yielded satisfactory results.

INTRODUCCIÓN

En la actualidad existe una gran tendencia en pequeñas, medianas y grandes empresas (en lo adelante PYMES), que hacen uso de los mecanismos para el control de acceso, los cuales se establecen para el cuidado de elementos, los mismos pueden ser materiales o de información delicada que podría dañar la integridad de las empresas, instituciones o algún otro lugar que sea caracterizado como delicado.

Para evitar lo antes planteado las empresas en aras del desarrollo, implementan medidas que restringen el paso a lugares específicos que pueden tener datos de alta confidencialidad o solo para personas autorizadas. Con la nueva era tecnológica surge el término seguridad tecnológica, la cual presenta diferentes áreas.

Según el experto Roberto Arbelaez (Arbelaez, 2016) de la corporación Microsoft nos muestra siete claves para lograr el éxito en la seguridad tecnológica de una empresa

- Infraestructura robusta: se debe hacer una buena gestión de riesgos
- Arquitectura bien definida
- Diseño: todos los componentes aislados deben fusionarse y diseñarse como un todo
- Operaciones seguras: buena configuración en los servicios
- Mejores prácticas: estudiar las mejores prácticas para determinar cuál se va a aplicar
- Administración de riesgos: se debe tener definido los riesgos en base a los datos con que se trabaja
- Cómputo en la Nube: reduce la exposición al riesgo, donde el proveedor es el encargado de crear las configuraciones

Antes las problemáticas que podrían presentarse si no se tiene un lugar que posea información delicada con seguridad extrema, surge el tema de investigación para realizar el proceso de autenticación a las personas que ingresan a un determinado lugar, ya que obtener algunos de los sistemas ya implementados en el mundo generaría altos costos de adquisición, instalación y mantenimiento.

Teniendo como base lo antes mencionado se tiene como:

Objetivo General: Desarrollar un sistema de control de acceso con la finalidad de resolver el problema de seguridad, abaratando costos, en un área restringida, usando técnicas de visión artificial.

Específicos:

- Investigar el tipo de sistemas de accesos que en la actualidad se utilizan para el ingreso a áreas restringidas a nivel mundial.
- Diseñar un sistema de reconocimiento facial para control de acceso en áreas restringidas.
- Implementación del sistema de reconocimiento facial.
- Pruebas de funcionamiento.

FUNDAMENTACIÓN TEÓRICA

1.1 Herramientas de procesamiento de imágenes.

1.1.1 OpenCV (Open Source Computer Visión).

Es una librería software open-source de visión artificial y machine learning. La misma presenta una construcción para sistemas que contengan visión artificial. Su esquema de licenciamiento es BSD, lo que facilita que se utilice y se modifique el código. OpenCV es muy utilizada a nivel comercial, la misma presenta más de 2500 algoritmos que reconocen objetos, caras, encuentran imágenes similares entre otros. Presenta varias interfaces las cuales están desarrolladas en C++, C, Python, Java y MATLAB interfaces, las cuales pueden ser vistas en Windows, Linux, Android y Mac OS. O. (Gracia, 2013).

OpenCV presenta 4 módulos:

Cv: contiene las funciones primordiales de la biblioteca.

Cvaux: tiene las funciones auxiliares o de experimento.

Cxcore: presenta todo lo que es estructuras de datos y funciones de soporte para Álgebra Lineal.

Highgui: presenta las funciones para el manejo de la Interfaz Gráfica de Usuario (GUI) (Furfaro, 2010).

Como otros de los aspectos importantes que se pueden destacar en OpenCV es la estructura que presenta en su directorio, la cual se presenta a continuación y se muestra el contenido que posee cada uno:

Tabla 1. Estructura del Directorio Opencv.

Directorio	Contenido
core	Funcionalidades básicas
imagproc	Parte del procesamiento de imagen de la visión por el computador(CV)
Highhui	Proporciona una interfaz de usuario, códecs de imágenes, de vídeos.

video	Hace el análisis de los videos, utilizando algoritmos de seguimiento
objdetect	Presenta algoritmos que se utilizan para la detección y el reconocimiento de objetos
ml	Tiene la Interfaces Gráficas de Usuario
video	Tiene el Flujo óptico, Plantilla de movimiento y Filtro kalman.
calib3d	Posee calibración de cámara, geometría epipolar y correspondencia
Features2d	Presenta detector de funciones 2d y descriptores
objdetect	Haar/LBP & HOG detectores de objetos, planar en base a los detectores
legacy	Funcionalidades obsoletas
User_contrib	Contribuyente del usuario
python	Interfaces de Python con nuevos estilos
ffmpeg	ffmpeg Fijaciones
triancascade	Haar/LBP aplicación de formación

Fuente: (Garage, 2010)

Otros de los aspectos para tener en cuenta en OpenCV es la calibración de cámaras. Para un arreglo estéreo que ya esté calibrado, el sistema puede realizar la triangulación de todos los puntos que se encuentren en el espacio, para de esta forma obtener la ubicación de cada objeto en correspondencia a la ubicación de la cámara y la distancia de la toma de la fotografía. Para la realización de un arreglo estéreo la mayor parte del código que se utiliza es de OpenCV, esos códigos forman parte de ejemplos que se encuentran en el directorio de dicha librería (Barranco Gutiérrez, Martínez Díaz, & Gómez Torres, 2016).

1.1.2 Sistema de Infrarojos.

Los sistemas infrarrojos facilitan la opción de cruzar objetos que se encuentren a poca luz en un escenario nocturno. Los mismos presentan varios sensores asociados a un conjunto de aplicaciones, ejemplo de una de ellas es la vigilancia de objetos y personas.

Como ejemplo de estos se tienen las cámaras infrarrojas, las cuales usan matrices para detectar y obtener la imagen. En la misma los pirómetros brindan una medida puntual y los escáneres de línea, que facilitan a la medida la composición de una línea del campo de visión y perfeccionan la imagen con un limpiado en el tiempo (Ferruz Melero, 2011).

A continuación, se muestra una imagen que tiene un sistema de infrarrojo que usan comunicaciones de manera inalámbrica:

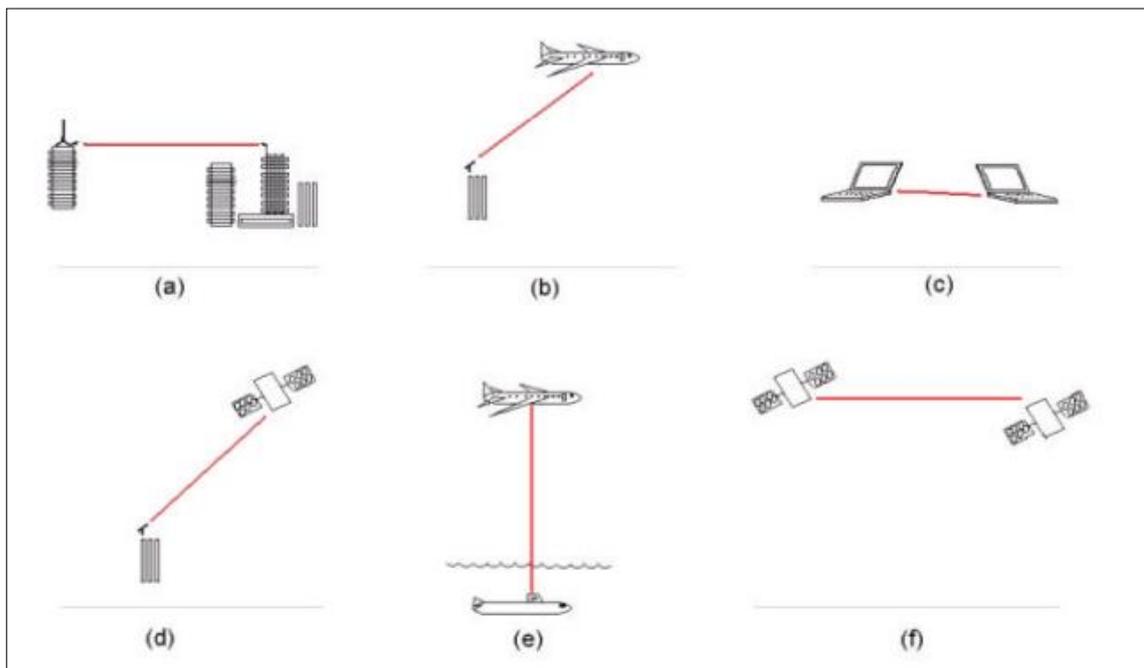


Figura 1. Comunicación inalámbrica desde diferentes perspectivas
Fuente: (Zamorano Flores & Serrano Moya, 2002)

En la figura anterior se muestra la comunicación desde diferentes puntos: a) comunicación terrestre; b) comunicación entre la tierra y el aire; c) entre dispositivos; d) entre la tierra y un satélite; e) entre el aire y un submarino; f) inter- satelital (Zamorano Flores & Serrano Moya, 2002).

La siguiente imagen muestra el proceso desde que la óptica detecta al objeto, hasta que el mismo es reflejado en alguna pantalla.

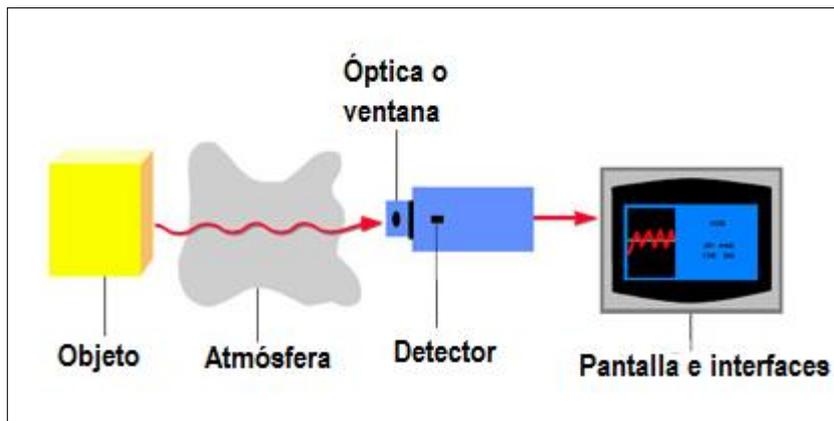


Figura 2. Proceso de infrarrojo
 Fuente: http://www.sapiensman.com/tecnoficio/optica/sensores_de_fibra_optica.php

1.2 Métodos de autenticación biométricas

1.2.1 Reconocimiento del Iris.

El iris es un órgano que se encuentra detrás de la córnea y el humor acuoso del ojo. El patrón del iris va a posibilitar el registro de imágenes a una cierta distancia, además de establecer un intervalo de tiempo para el sujeto, el cual no se altera.

A continuación, se muestran imágenes que ejemplifican el reconocimiento del iris:

En esta primera representación se va a obtener la imagen de la persona que se almacenará en la base de datos, tomando el patrón del iris y asociándole un código a dicha muestra.

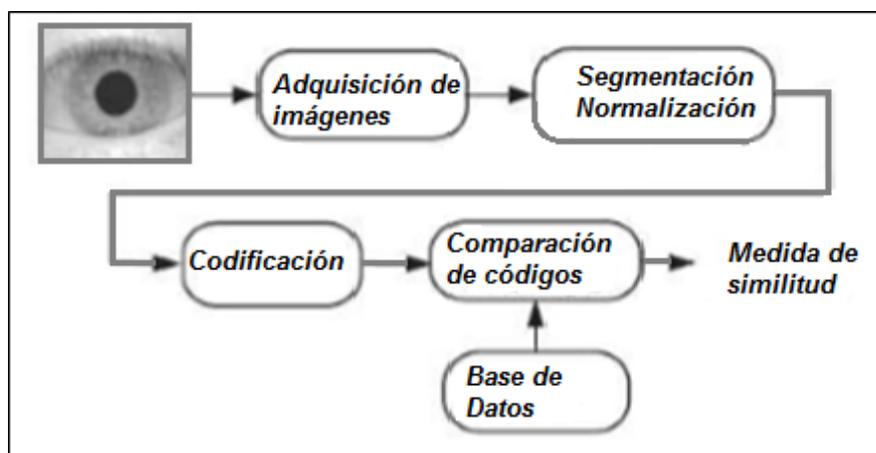


Figura 3. Diagrama representado en bloques del reconocimiento del iris
 Fuente: (Florian & Carranza, 2006)

En la siguiente figura se comprueba que el sujeto que está accediendo está registrado en la base de datos.

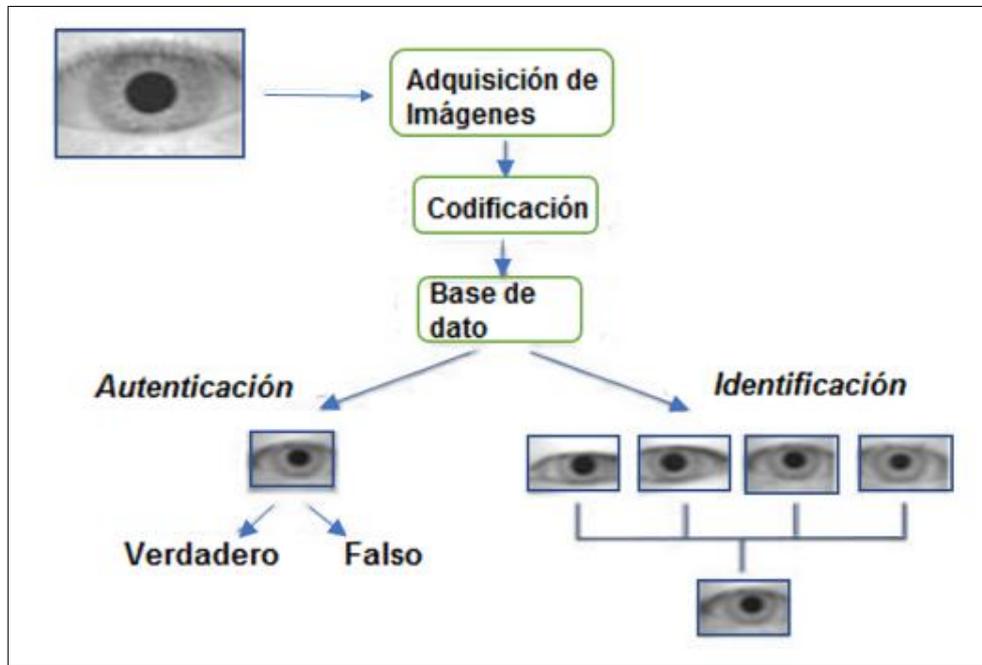


Figura 4. Reconocimiento, autenticación e identificación
Fuente: (Florian & Carranza, 2006)

Para adquirir imágenes existen varios métodos, la tendencia es usar una cámara de video, en donde haya sistemas de iluminación sofisticado (Florian & Carranza, 2006)

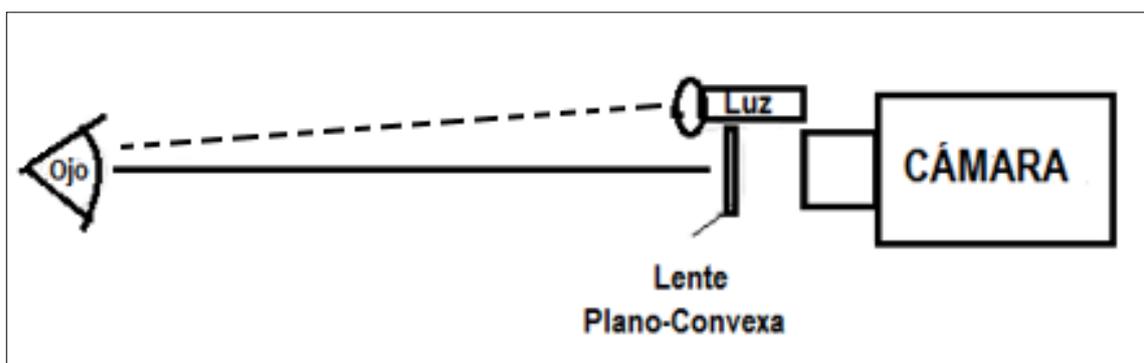


Figura 5. Elementos básicos para la obtención de la imagen
Fuente: (Florian & Carranza, 2006)

En la figura anterior se muestra una cámara que está tomando una imagen de un ojo a través del lente y donde hay una proyección de luz.

Sensores:

Para el reconocimiento del iris se debe usar cámara de video de tipo CCD (Dispositivo de Carga Acoplada), el cual posee miles de elementos independientes expresado en pixeles.

1.3 Técnicas de reconocimiento facial.

Las técnicas de reconocimiento facial constituyen un elemento fundamental para el reconocimiento, ya que el rostro es una de las partes más importantes del ser humano, los objetivos de las mismas son detectar y localizar el rostro dada una imagen, además de determinar la identidad de dichos rostros.

Numerosas son las técnicas que existen en la actualidad, las cuales pueden estar basadas en imágenes que pueden encontrarse en dos dimensiones (2D) por solo citar algún ejemplo y que para el reconocimiento utilizan plantillas, aproximaciones, redes neuronales entre otros aspectos.

A continuación, se enuncian algunas de las técnicas de reconocimiento facial.

1.3.1 Los sistemas de reconocimiento facial automático.

Su objetivo principal es dado una o varias imágenes las cuales pueden ser tomadas estáticas o en movimiento de un rostro desconocido, escoger en la base de datos de los rostros de personas conocidas, y devolver la que tengan mayor grado de exactitud en el parecido. Si se hace un análisis de un rostro desconocido para el sistema, en donde no puede hallar esa cara que no está registrada, se mostrará alguna alerta que indique que no está identificada la persona (Moreno Díaz , 2004).

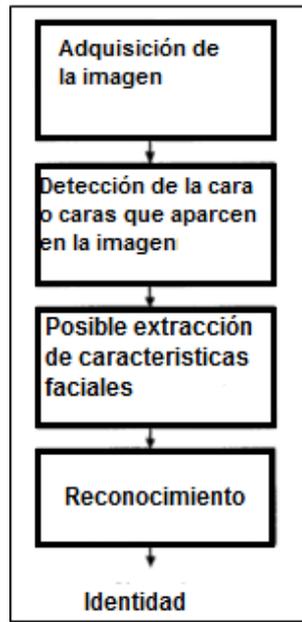


Figura 6. Reconocimiento genérico
Fuente: (Moreno Díaz , 2004).

La siguiente figura muestra las diferentes etapas por la pasa un proceso de reconocimiento de genérico con visión.

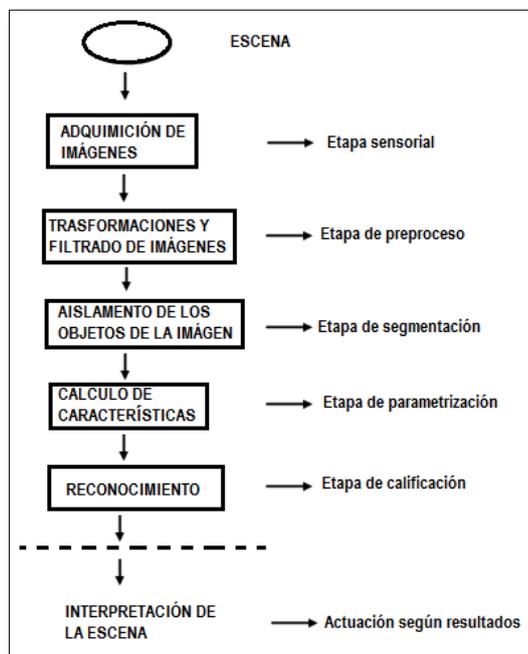


Figura 7. Reconocimiento genérico con visión
Fuente: (Moreno Díaz , 2004)

Por otra parte, se puede decir que estos sistemas utilizan técnicas con cierto nivel de aproximación facial que podrían clasificarse como un acercamiento holística o analítica. El acercamiento holístico realiza consideraciones globales en cuanto a las propiedades del patrón, mientras que el analítico lo hace mediante las características de la cara de forma geométrica. Además, se puede decir que hay 2 divisiones en la parte analítica que está dada por el acercamiento basado en vectores que son extraídos de característica de un perfil y por los vectores extraídos de una manera frontal de la cara (Espinosa Duró, 2014) .

1.3.2 Reconocimiento facial a partir de imágenes 3D.

El resultado de esta técnica va a depender de cómo se obtuvo la imagen, ya que la misma es sensible a variaciones de las imágenes en la iluminación, escala, orientación de la cara y expresiones faciales. Cuando se transforma un objeto 3D a 2D se pierde mucha información que serviría para la precisión en el sistema.

Una de las informaciones que se pierde es la escala real. Además de características esenciales de la persona, por ejemplo, algún gesto, la precisión de la cara en el momento que se toma la foto, la posición de la luz, en fin, estos y muchos elementos hacen que varíen las imágenes de una misma persona (Moreno Díaz , 2004).

Para el reconocimiento facial a partir de imágenes 3D se puede hacer uso de descriptores los cuales van a usar un conjunto de herramientas matemáticas y un conjunto de procedimientos que facilitarán la obtención de diferentes medidas sobre el entorno y las distribuciones morfológicas en las imágenes (Cerón Correa , Salazar, & Prieto Ortiz, 2013)

1.3.3 Técnicas de adquisición de imágenes 3D.

De acuerdo con la investigación realizada por (Moreno Díaz , 2004) en tu tesis doctoral sobre Reconocimiento Facial Automático mediante Técnicas de Visión Tridimensional, existen varias técnicas que se utilizan para obtener la adquisición 3D, siendo estas:

1.3.3.1. Técnicas pasivas.

Hacen uso de sensores pasivos, dígase cámara fotográfica o de video solamente, las cuales capturan imágenes 2D de intensidad o secuencias de vídeo para escenas 3D.

Esta técnica utiliza la de Visión Estereoscópica la cual permite la recuperar la información en 3D, de ella se derivan otras técnicas que permiten la extracción a partir del movimiento, la misma permite la obtención de la imagen 3 a partir de 2 o más imágenes de lo mismo en diferentes posiciones (Moreno Díaz , 2004).

1.3.3.2 Técnicas activas.

Hacen uso de sensores activo participante de una operación, ejemplo, un sensor que permita la variación del enfoque en el momento de tomar las imágenes exponiendo un haz de luz reconocidos de energía (luz o sonido) a partir de una posición y orientación reconocidas. Ejemplos: radares, sonoros, interferometría holográfica, enfoque o triangulación activa. El último elemento mencionado es el más utilizado en Visión Artificial (Moreno Díaz , 2004).



Figura 8. Radar
Fuente. <https://es.dreamstime.com/stock-de-ilustraci%C3%B3n-ejemplo-del-radar-image70022299>

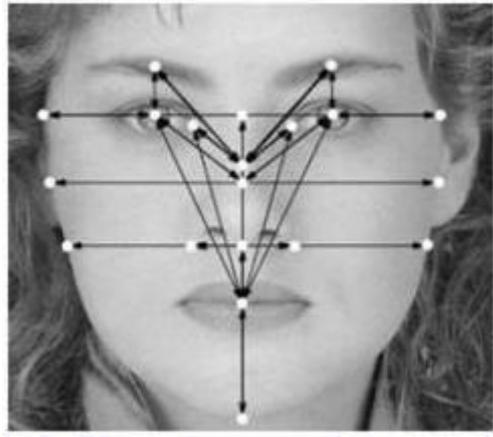


Figura 9. Detección de caras a través de la luz
 Fuente: <https://www.xatakafoto.com/cameras/como-funciona-la-deteccion-de-caras>

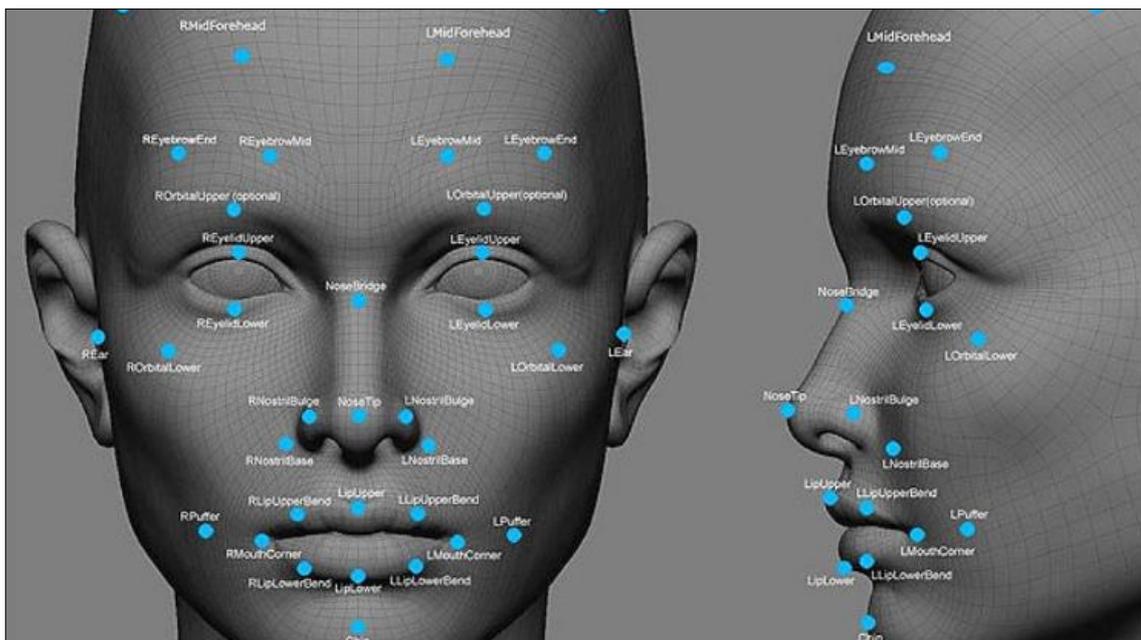


Figura 10. Puntos esenciales para el reconocimiento facial mediante la luz

Fuente: http://www.elconfidencial.com/tecnologia/2015-03-03/zte-lanza-el-primer-smartphone-que-se-desbloquea-con-la-mirada_721281/

Las imágenes de rango: son aquellas que se codifican en la superficie 3D, y representan el objeto en una superficie tridimensional de forma real.

Las imágenes 3D de rango pueden venir dadas según Moreno Díaz (Moreno Díaz , 2004) por:

- Las coordenadas (x, y, z) de puntos 3D, se refiere a un sistema de referencia del espacio tridimensional, que no posee especificación de ordenación (a eso se le llama forma xyz o nube de puntos).
- Las coordenadas (x, y, z) de puntos 3D, pueden referirse a un sistema de referencia del espacio tridimensional y las conexiones que se forma es un triángulo.
- Las matrices de valores z de profundidad de puntos, plantea que direcciones de los ejes Y y X de la imagen, son llamada de la forma ry , y estarán de forma explícita. Esas imágenes se nombran imágenes de profundidad o mapas de profundidad.

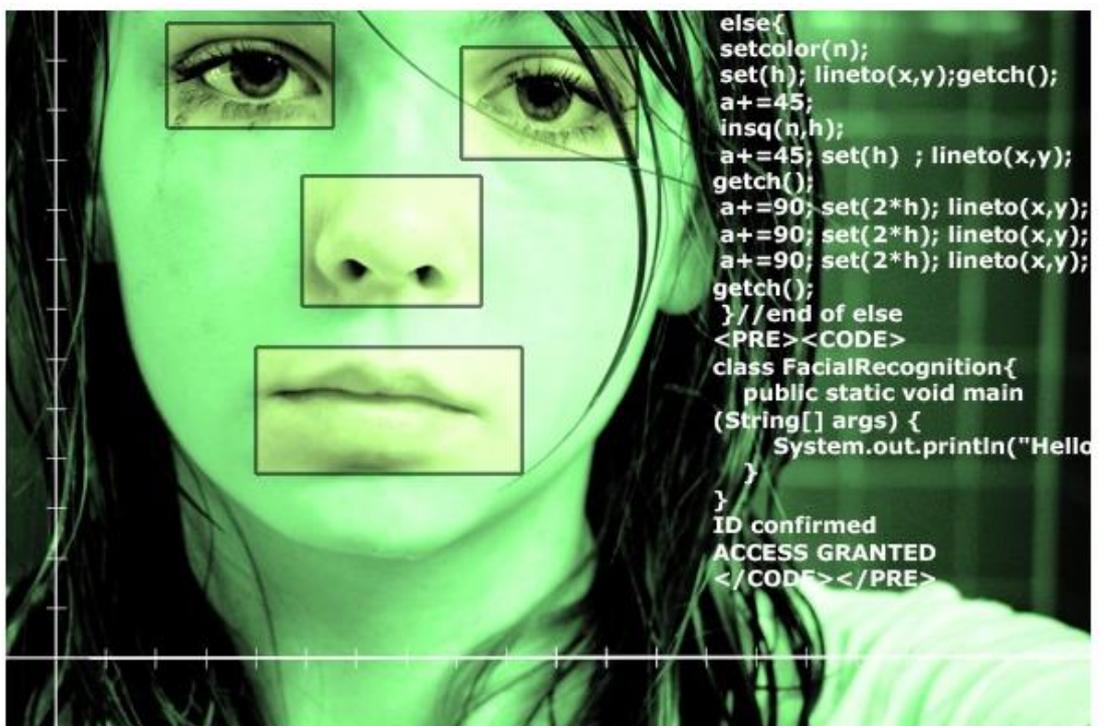


Figura 11. Reconocimiento con 3 coordenadas

Fuente: <http://www.viro.com.co/index.php/services/seguridad/reconocimiento-facial>

1.3.4 Técnicas de reconocimiento 3D basadas en rasgos locales

Las técnicas de reconocimiento 3D basados en rasgos locales, utilizan la geometría diferencial basada en superficies para obtener descriptores. Las mismas a través de mallas con polígonos inmersos se van a encontrar en la superficie, dicha superficie no será variable cuando se realicen las transformaciones geométricas en la curvatura que se encuentra en los puntos del objeto 3D. Para describir la superficie 3D se utiliza la Visión Tridimensional (Moreno Díaz , 2004).

A continuación, se muestran varios métodos que usan estas técnicas de reconocimiento 3D basadas en rasgos locales:

- a) Métodos basados en la curvatura local de la superficie:** en estos métodos la curvatura local y las diferencias angulares entre normales son invariantes ante las transformaciones geométricas.

La curvatura siempre va a variar en dependencia de las direcciones que tome en la superficie. Ellas codifican el promedio del cambio en la orientación. Además extraen 12 descriptores 3D de regiones del rostro en términos segmentados que puede estar dado por: anchura de los ojos, separación entre los ojos, distancia entre las esquinas extremas de los dos ojos, altura de la nariz, anchura de la nariz, profundidad de la nariz, máxima curvatura Gaussiana en la línea de la parte frontal de la nariz, mínima curvatura en la línea de la parte frontal de la nariz, curvatura Gaussiana en el punto de silla superior de la nariz y curvatura Gaussiana en la curva de la base de la nariz (Moreno Díaz , 2004).

- b) Métodos basados en plantillas:** los mismos describen las plantillas, en la que se hace la segmentación, a través de una clasificación de las partes en la superficie, ya sean planas, esféricas o en superficies de revolución.

Emplea descriptores basados en cálculos de la curvatura en las imágenes de rango, donde dichas imágenes tienen una resolución menor de 0.4 mm y los datos se almacenan en coordenadas cilíndricas (Moreno Díaz , 2004).

- c) Métodos basados en el cálculo de la firma:** se basa en calcular la firma desde algunos puntos de la superficie 3D con el objetivo de obtener las características. La firma en los puntos de las imágenes de rango se usa para la representación de objetos 3D de forma libre. Con estos métodos se resuelve el problema de reconocimiento facial automático mediante el siguiente método: se crea una librería de rasgos con partes rígidas de las caras de entrenamiento. Dada una cara de muestra o prueba, se comparan dichas partes para obtener cuál es la de mayor parecido en términos de su firma. Este método se experimentó con éxito utilizando imágenes de rango de 6 individuos, presentando expresiones faciales (Moreno Díaz , 2004).

1.4 Algoritmos de reconocimiento facial.

1.4.1 Algoritmo CamShift.

Este algoritmo en su estructura presenta una implementación del algoritmo Mean Shift (Vega Aquino, 2011), en donde se describe una interfaz perceptual completa, a partir del conjunto de actividades desde el seguimiento hasta la transformación de cada uno de los resultados en variables de control para las interfaces. Para crear el color de la piel se hace con el histograma del canal Hue del espacio HSV (Vega Aquino, 2011) en donde se desechan los píxeles que no presentan un nivel alto de saturación; además se eliminan las más claros u oscuros.

El algoritmo inicia desde una ventana en donde realiza una búsqueda inicial (v) y luego se realizan los siguientes pasos (Vega Aquino, 2011):

- 1: Realizar el Cálculo de la media de la imagen de probabilidad (p) desde la ventana (v).
- 2: Mover el medio de la ventana, teniendo como base el resultado del paso anterior y cambiar el tamaño por el resultado de la suma de probabilidad (p) en el medio de la ventana (v).
- 3: Hacer cada uno de los pasos anteriores hasta que se logre alcanzar la convergencia. La aplicación del algoritmo resulta sencilla por lo que se puede desarrollar sin que haya complicaciones, ya que adaptará los movimientos de manera rápida y eficiente (Vega Aquino, 2011) .

En imagen que está a continuación se muestra el proceso inicial a través del marcado las partes principales que se analizará para la aplicación del algoritmo de Camshift

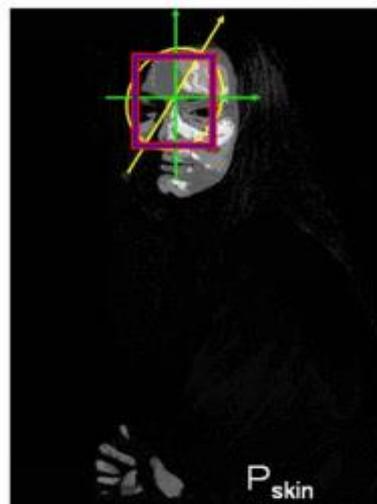


Figura 12. Camshift inicial
Fuente: (Camshift, 2016)

En imagen que está a continuación se muestra el resultado una vez aplicado el algoritmo de Camshift

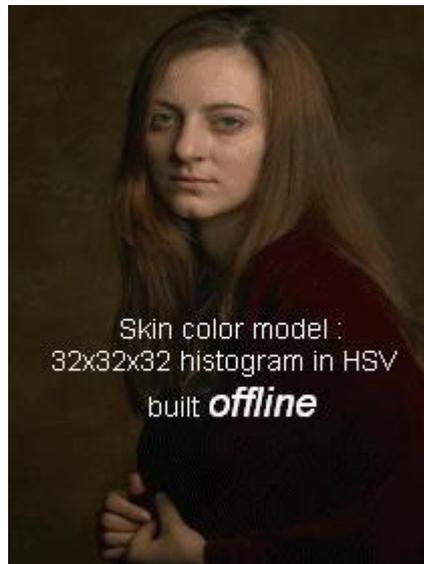


Figura 13. Camshift final
Fuente: (Camshift, 2016)

1.4.2 Algoritmo de alineamiento iterativo de Lucas Y Kanade.

En dicho algoritmo se toman 2 imágenes en instantes sucesivos, con el objetivo hallar el desplazamiento en un píxel que represente (x,y) y se denota como (v_x, v_y) donde los mismos constituyen vectores en desplazamiento con un valor mínimo tanto para las (x) como para las (y) , y $I_t(x,y)$ no es más que el desplazamiento total de (x,y) .

Dicho de otra manera, se reitera la búsqueda total y admite el desplazamiento de manera pequeña, para hallar los valores de (v_x, v_y) se utiliza la siguiente ecuación:

$$I_t(x, y) = I_{t+1}(x + v_x + y + v_y) \quad (1)$$

Con una ejecución se obtendría el valor de (v_x, v_y) que forma el residuo menor. Se necesita realizar diferentes iteraciones hasta lograr alcanzar la convergencia.

Se debe cumplir los siguientes supuestos de movimiento uniforme:

1: Tamaño de las ventanas: facilita la definición de una escala de posibilidades desde pequeñas regiones (ejemplo de éstas: fosas nasales, esquina de la boca, entre otras), hasta la aplicación de la imagen de la cara completa.

2: Imagen de referencia: tiene como referencia un esquema, que podría ser una imagen de la misma secuencia o un esquema estadístico entrenado.

3: Funciones de modelado del movimiento: necesita de funciones complejas para representar el movimiento facial.

4: Movimientos no rígidos e iluminación; se utilizan esquemas de malla deformables; o se puede utilizar auto-espacios.

Este algoritmo se centra en la apariencia con el objetivo de obtener mayor precisión, referencia y estabilidad en 3D de las caras. Presenta un elevado costo computacional, no es adaptable en circunstancias de movimientos rápidos, puede provocar pérdida en el seguimiento de la imagen (Vega Aquino, 2011).

1.4.3 PCA (Análisis de Componentes Principales).

Su objetivo principal es transformar una cantidad de variables en un número menor. Este algoritmo reduce la dimensión para encontrar cierta cantidad de vectores para mejorar la representación de distribución de un grupo de imágenes. Entre sus características está la representación de una imagen de una cara, para el análisis usa observación de varias caras; además facilita la separación entre las clases (Gimeno Hernández, 2010).

PCA hace uso de las covarianzas la cual muestra la probabilidad en las variaciones de dos variables en correspondencia con sus medias, formando con los valores que no sean nulos una matriz de covarianza S que establece una base ortogonal en el subespacio con diferencia B . Para la aplicación del algoritmo se debe tener un conjunto de datos para aplicar determinados cálculos, entre ellos se encuentra la resta de la media, el cálculo en matriz de covarianza y los valores de la misma, la elección de componentes y formación de la base con los vectores, así como el procesamiento de los datos (Armengot Iborra, 2006).

El objetivo fundamental de PCA es hallar una base que permita una mejor distribución de cada una de las imágenes que se encuentran en el espacio (López, Alvarez, & Castrillon, 2008).

1.4.4 LDA (Análisis Discriminante Lineal).

Su objetivo principal es reducir la dimensión sin supervisión. Si es supervisado utiliza la siguiente fórmula para maximizar la información de los datos (Gimeno Hernández, 2010):

$$J(w) = \frac{w^T \cdot SB \cdot W}{W^T \cdot Sw \cdot W} \quad (2)$$

De la representación anterior se tiene que w^T es la dispersión, SB es la “matriz de dispersión entre clases” y SW es la “matriz de dispersión intra clase” y se definen como (Gimeno Hernández, 2010) :

$$SB = \sum_c N_c (U_c - X^-) (U_c - X^-)^T \quad (3)$$

$$Sw = \sum_c \sum_{i \in c} (X_i - U_c) (X_i - U_c)^T \quad (4)$$

$$U_c = \frac{1}{N_c} \sum_{i \in c} X_i \quad (5)$$

$$X^- = \frac{1}{N} \sum_i X_i = \frac{1}{N} \sum_c N_c * U_{ci} \quad (6)$$

N_c : # de casos en la clase c .

Después que se resuelve el problema se logra la matriz de proyección la cual permite saltar de un problema de alta dimensión a uno de baja (Gimeno Hernández, 2010).

A continuación, se muestra datos característicos de la clase LDA

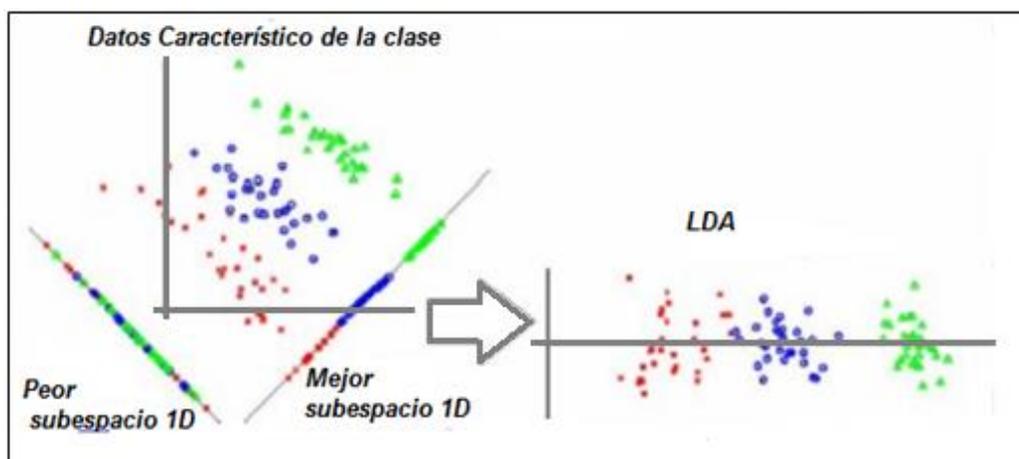


Figura 14. Datos característicos de la clase LDA
Fuente: <https://mlalgorithm.wordpress.com/tag/linear-discriminant-analysis/>

1.4.5 LPP (Proyecciones de Preservación de la Localidad).

Es un algoritmo lineal que efectúa una reducción de dimensión en los datos. Es rápido y muy útil para la realización de aplicaciones de uso práctico. El mismo conserva la estructura local en vez de global. Sus planos están hechos para reducir algún criterio objetivo que provenga de las técnicas lineales. Además, permite la representación de nuevos subespacios sin tener que recalcular las matrices de proyección (Gimeno Hernández, 2010).

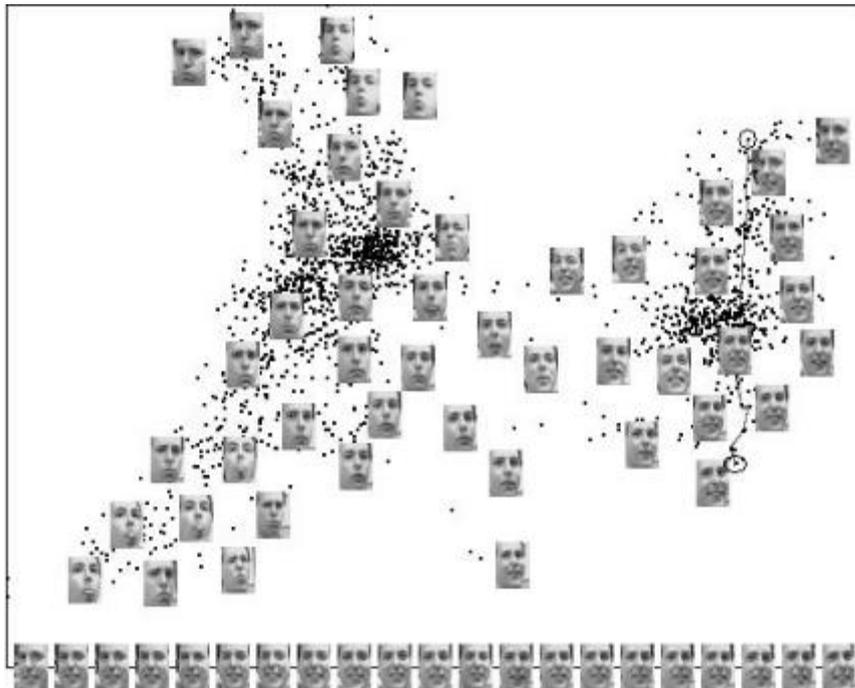


Figura 15. LPP

Fuente: <http://www.cad.zju.edu.cn/home/xiaofeihe/LPP.html>

1.4.6 DCT (Transformada Discreta de Coseno).

Hace una representación de secuencias finita de los datos a través de la suma de una serie de funciones cosenoidales oscilando a diferentes frecuencias (Gimeno Hernández, 2010):

Se utiliza en aplicaciones que procesan señal y establece 8 variantes, siendo las más relevantes DCT-I, DCT-II, DCT-III y DCT-IV.

En lo que es reconocimiento facial, utiliza el DCT-II, la cual se define como:

$$v(k_1, k_2) = \alpha(k_1) * \alpha(k_2) \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} u(n, m) * \cos\left(\frac{(2n+1)\pi * K_1}{2N}\right) * \cos\left(\frac{(2m+1)\pi * K_2}{2M}\right) \quad (7)$$

$$0 \leq k_1 \leq N - 1$$

$$0 \leq k_2 \leq M - 1$$

Donde $u(n)$ es una secuencia de N elementos, $v(k)$ es transformada a:

$$\alpha(0) = \sqrt{\frac{1}{N * M}}, \alpha(k_1) = \sqrt{\frac{2}{N}}, \alpha(k_2) = \sqrt{\frac{2}{M}} \quad (8)$$

$$0 \leq k_1 \leq N - 1$$

$$0 \leq k_2 \leq M - 1$$

De la formula anterior se tiene una entrada de una imagen u y la DCT sería la matriz de transformación para tener W , además se obtiene una expresión donde $W = c(k, \xi) = c(k_1, n) = c(k_2, m)$ y k y ξ son las filas y columnas : (Gimeno Hernández, 2010).

$c(k, c)$

$$= \begin{cases} \frac{1}{\sqrt{N * M}}, k = 0 & 0 \leq n \leq N - 1, 0 \leq m \leq M - 1 \\ \sqrt{\frac{2}{N}} \sqrt{\frac{2}{M}} * \cos\left(\frac{(2n+1)\pi * K_1}{2N}\right) * \cos\left(\frac{(2m+1)\pi * K_2}{2M}\right), & 0 \leq k_1 \leq N - 1 & 0 \leq k_1 \leq N - 1 \\ & 0 \leq k_2 \leq M - 1 & 0 \leq k_2 \leq M - 1 \end{cases} \quad (9)$$

Así es como se obtiene un sistema que presenta una matriz de proyección, para representar imágenes en el subespacio de baja dimensionalidad como lo hace PCA (Gimeno Hernández, 2010).

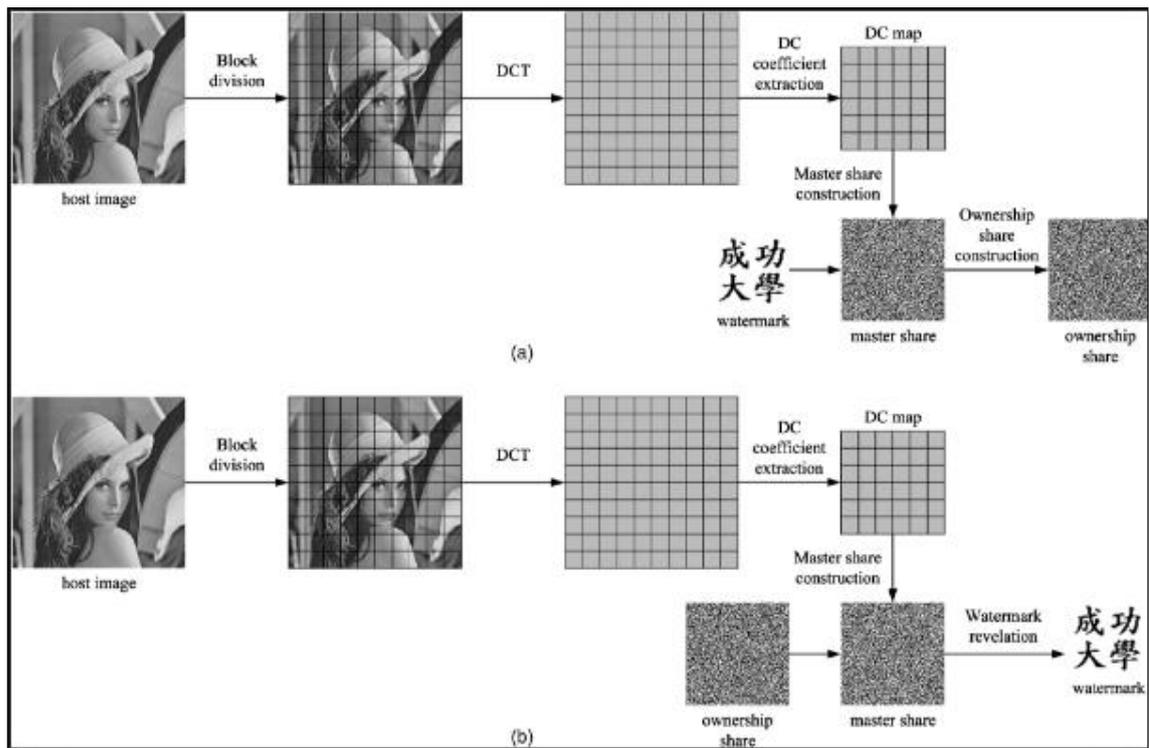


Figura 16. DCT
Fuente: (Ming-Shi Wang ; Wei-Che Chen, 2008)

1.4.7 (LBPH) Histogramas de patrones binarios locales.

Este algoritmo se encarga de describir toda la textura, mediante un operador que va etiquetando cada uno de los píxeles en la imagen a través del umbral, considerando cada vecindad en 3x3 valorado en cada píxel, partiendo del valor central y considerándolo como número binario. En este algoritmo el histograma se utiliza como descriptor de la textura mediante las etiquetas. El mismo se utiliza para clasificar la visión por computadora. Hay versiones de Python que contienen paquete que implementan este algoritmo, además de versiones de la biblioteca OpenCV que lo aceptan a partir de la versión 2 (Wagner, 2011) .

Por otra parte, se puede decir que presenta un poder bastante alto discriminativo y de simplicidad a nivel computacional, debido al operador que presenta de textura, ya que mismo se utiliza en diferentes aplicaciones en la actualidad por la robustez que posee ante los cambios a gran escala, ejemplo de este puede ser la variación ante la iluminación. La simplicidad de éste a nivel computacional está dada por el análisis de imágenes de manera desafiante a través de configuraciones que se realizan en tiempo real (Pietikäinen, 2010).

A continuación, se muestra una imagen que ejemplifica las diferentes fases del algoritmo

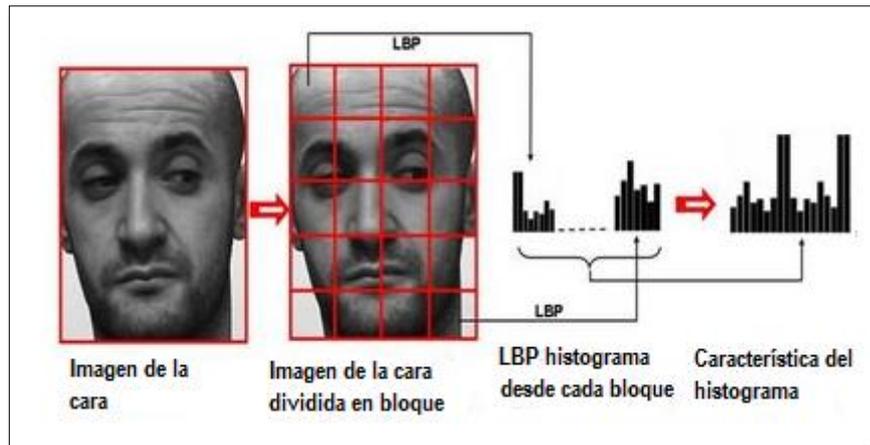


Figura 17. LBP Histograms
Fuente: (Pietikäinen, 2010)

A continuación, se muestra una comparación de varios algoritmos de reconocimiento facial basado en características esenciales.

Tabla 2. Comparación de algoritmos de reconocimiento facial

Algoritmos	Características principales	Ventajas	Desventajas
(LBPH) Histogramas de patrones binarios locales	Describe la textura de la imagen. Trabaja con entornos que tienen tamaño variable.	Facilita el ahorro de tiempo cuando se trabaja con una única máscara genérica en la parte del entrenamiento	No se obtienen resultados satisfactorios cuando al adiestrar una máscara para cada una de las galerías de imágenes
(DCT) Transformada Discreta de Coseno	La transformación que realiza es independiente a los datos, por lo que transforma de manera discreta Todo el procesamiento de señales digitales y de imágenes.	Utiliza estándares que se utilizan para la compresión de las imágenes	Si el desarrollo presenta una alta complejidad desde el punto de vista computacional, su implementación pueda que no sea factible

(LPP) Proyecciones de Preservación de la Localidad	Su diseño se basa en la conservación de las estructuras	Resguarda una buena estructura de manera local de los datos y posee un buen rendimiento ante distintas situaciones.	No realiza una buena captura si se realizan cambios con respecto a la iluminación.
Algoritmo CamShift	Posee espacios para trabajar encima de los histogramas.	Resulta eficiente a la hora de calcular la posición del objeto.	No realiza un buen seguimiento del rostro si la iluminación se encuentra afectada.
Algoritmo de alineamiento iterativo de Lucas Y Kanade	Realiza cálculo de desplazamiento de los puntos esenciales en la imagen.	Facilita la obtención y mayor precisión en 3D de las caras que se tienen por referencia.	Presenta un elevado costo computacional, no se obtienen resultados favorables cuando existen movimientos rápidos.
PCA Análisis de Componentes Principales	Busca el predominio según los datos que estén mejor representados en procesos de mínimos cuadrados.	Reduce las dimensiones de un conjunto de datos.	Se asume que los datos observados son combinación lineal de una cierta base.
LDA Análisis Discriminante Lineal	Reduce la dimensión sin supervisión.	El resultado de la covarianza se puede estimar a través del número del entrenamiento.	Con el cálculo de la discriminante en números reales, el resultado de cada muestra puede superar el número de cada clase.

Fuente: Tesista

Después de analizar varios algoritmos de reconocimiento facial, y haciendo una comparación basada en las características esenciales, ventajas y desventajas de los mismos, la investigación hará uso de LBP Histogram para dar solución a la problemática planteada en la investigación.

1.5 Visión por computadora.

Es una rama de inteligencia artificial que estudia el procesamiento, el análisis y la interpretación de imágenes de manera automática. La misma se usa en distintos ámbitos como la medicina o como medio de seguridad mediante la inspección de manera automática. Su objetivo fundamental es detectar objetos.

Para realizar la visión por computadora se necesita conocer de varios componentes, los cuales se muestran en la figura que viene a continuación.

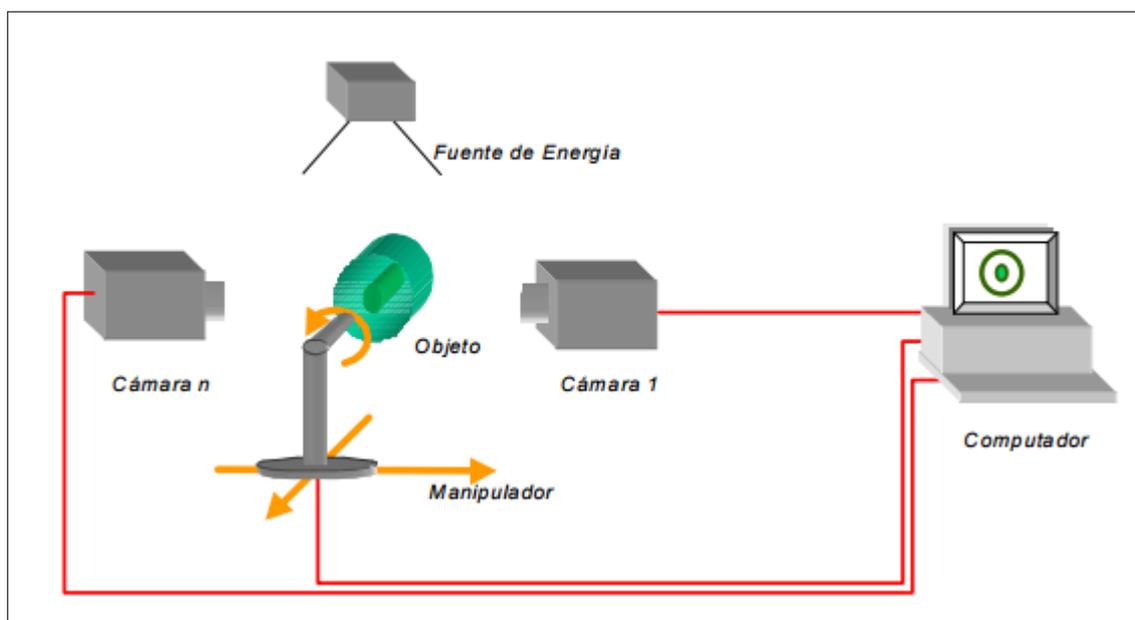


Figura 18. Componentes de la visión por computadora
Fuente: (Mery, 2004)

De la figura anterior se puede decir que la fuente de energía es el componente que se va a utilizar para la toma de la imagen, la misma puede representarse con luz, con rayos X, radiografía, tomografía entre otros. El manipulador es el que se va a mover y va a ubicar el objeto y el computador va a procesar toda la información captada el convertidor.

Existen otros componentes que se debe tener en cuenta para la visión por computadora como es sensor de imagen el cual es sensible a la fuente de energía que se esté utilizando, además del convertidor analógico digital el cual convierte la señal eléctrica en código binario para que

sea interpretado por la computadora para hacer la imagen que se ha tomado digital en el objeto para el estudio (Mery, 2004).

La visión por computadora realiza la codificación dada una imagen digital, la cual es codificada en de manera binaria, llevando cada pixel que se encuentra en la imagen, determinando colores y combinación entre ellos, y de esta forma va a crear patrones para encontrar un objeto intrínseco en la imagen (Computer Vision, 2014).

A continuación, se muestra una imagen que te hace sentir ilusiones visuales dado la combinación de los colores

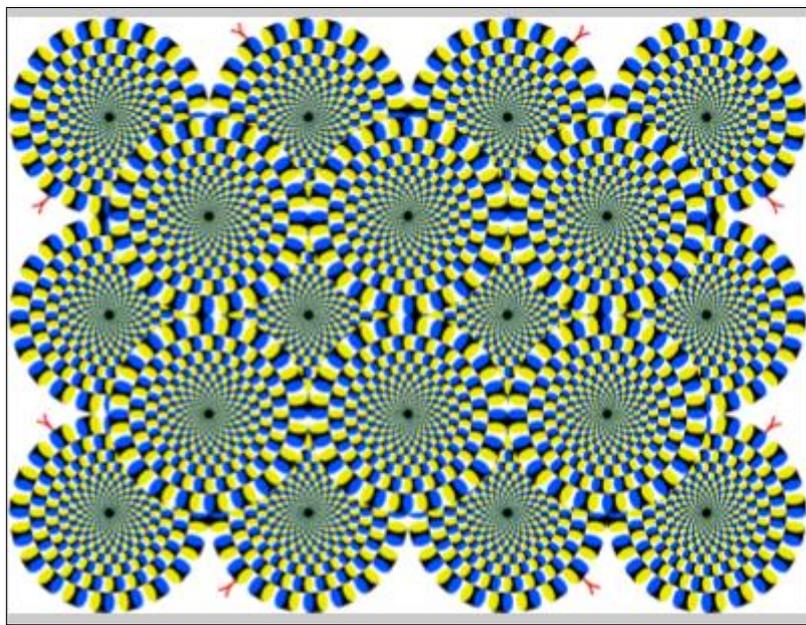


Figura 19. Ilusión visual
Fuente: (Computer Vision, 2014).

Si se observa la imagen puede que se note cierto movimiento dado el contraste de los colores, la iluminación, la geometría que presenta y la dimensión.

1.6 Lenguajes de Programación.

1.6.1 PHP.

Es un lenguaje de código del lado del servidor. Su código se interpreta por un servidor web. Se usa en la mayoría de los servidores web, sistemas operativos y plataformas sin costo alguno.

Tiene varios framework entre los que se encuentran: Laravel, CodeIgniter, CakePHP, Symfony, Zend Framework, Phalcon, Yii, Aura. Todos facilitan el aumento en la productividad

de los programadores reduciendo el tiempo de programación, a través de los componentes y subrutinas que se reutilizan.

Se puede utilizar en varios gestores de bases de datos, ejemplo de estos son: MySQL, PostgreSQL, Oracle y Microsoft SQL Server. Además de distintos sistemas operativos como Unix, Linux y Windows (Alvarez & Alvarez, 2016).

1.6.2 C#.

Es orientado a objetos, se puede usar varios sistemas operativos ejemplos de estos son: GNU/Linux, Microsoft Window, de manera general todos aquellos sistemas asentados Unix. Se estandarizó por Microsoft como segmento de su plataforma .NET, la cual se provee servicios automáticos al código que se establece, entre las actividades que posee tiene: que carga las clases, compila MSIL a nativo, posee un motor de seguridad y otro para depurar, un recolector de basura gestiona el código y las excepciones (Hilyard & Stephen, 2015).

1.6.3 Python.

Es multiplataforma, de código abierto, fácil de entender y usar, presenta buena respuesta en la ejecución por la implementación que presenta de instrucciones en el lenguaje de programación C. Al ser un lenguaje de programación interpretado no requiere recompilar el código fuente para los cambios que se realicen en la aplicación, al no presentar fortaleza en el tema de tipado, tiende presentar dificultades en entender grandes volúmenes de código fuente (Eslava Muñoz, 2012).

A continuación, se muestra una tabla comparativa con los lenguajes de programación analizados.

Tabla 3.Comparación de lenguajes de programación

Lenguaje de programación	Características	Ventajas	Desventajas	Opinión de la comunidad
PHP	Se usa para hacer páginas web dinámicas. Se ejecuta en el servidor.	Fácil de aprender. Multiplataforma. Libre y gratuito.	Necesita de un servidor para funcionar.	Presenta bastante documentación.

	Se considera lenguaje de alto nivel.			
C#	Está orientado a objetos Usa la plataforma .Net de Microsoft	Se pueden crear aplicaciones web, de escritorio y móviles.	Necesita mínimo 4 GB para su instalación	Ideal para desarrollar aplicaciones en Windows
Python	Permite crear todo tipo de programas No requiere de compilación para interpretar el código	Libre y de código abierto Multiplataforma	suelen ser lento, porque es interpretado	Excelente para hacer páginas web dinámicas

Fuente: Tesista

Al realizar el análisis de varios lenguajes de programación y los resultados a los que se quiere llegar, la investigación usara para el desarrollo Python.

1.7 Bases de datos.

Base de Datos (BD): “Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular”

Sistema de Gestión de Base de Datos (SGBD): su objetivo fundamental es brindar servicios entre la interfaz entre la base de datos, teniendo como actor fundamental al usuario, y las aplicaciones que se estén utilizando:

Entre las principales características que posee se encuentran:

- “Independencia lógica y física de los datos.

- Redundancia mínima.
- Posee un acceso concurrente.
- Presenta integridad de todos sus datos.
- Optimiza las consultas complejas.
- Presenta seguridad en cuanto al acceso y la auditoría.
- Posee respaldo y recuperación (Pérez Valdés, 2006).”

A continuación, se ejemplifican algunos gestores de base de datos:

1.7.1 Oracle.

Se utiliza en gran parte de las empresas en el mundo, utiliza plataforma base para suministrar un BD MS completo, la administración de los datos se hace de manera rápida y eficiente, por lo que las consultas responden de manera rápida.

Posee una arquitectura cliente/servidor que provee la gestión de datos, en el mercado su costo es alto y solo las grandes empresas multinacionales la obtienen. Presenta poderosas herramientas que brindan seguridad a los datos, se puede acceder de manera remota y en redes, es considerado el sistema de bases de datos mejor estructurado, porque soporta diferentes transacciones, es sólido, escalable y es multiplataforma. Además, facilita ejecución de respaldos y recuperación de los datos. El acceso a los datos se hace mediante la gestión de privilegios., esta actividad la realiza el administrador del BD, el cual tiene que instalar, crear y controlar a los usuarios. Es estable y escalable (Vegas, 1998).

1.7.2 MySQL.

Es código abierto, es una de las más utilizadas en el mundo, resulta llamativo por su arquitectura que presenta dos capas, una de las capas es de SQL y la otra es donde están los motores de almacenamiento.

El mismo presenta herramientas multi-usuario, trabaja en plataforma de Unix y Windows; aguanta diferentes lenguajes de programación, por ejemplo: C, C++, Java, Perl, PHP, Pitón, entre otros. El gestor de base de datos más usado para MySQL es PHPMyAdmin desde donde se puede gestionar todos los datos. Se considera la más rápida y robusta tanto para volúmenes de datos tanto grandes como pequeños. Como aspecto relevante, presenta es rápida y robusta; hace poco consumo de recursos en la CPU y en la memoria; es multiproceso;

tiene buen rendimiento; con la misma se puede trabajar en diferentes y SO; puede ser integrada con PHP; el método de autenticación y privilegios es seguro y flexible; las claves en la red viajan de maneja encriptadas (ROLLET, 2014).

1.7.3 PostgreSQL.

Es de código abierto, multiplataforma, soporta gran tamaño de almacenamiento de objetos y soporta transacciones, vistas, triggers, joins, foreign keys y stored procedures. El esquema de seguridad que presenta usa un protocolo de comunicación encriptado por SSL. No tiene costo su esquema de licencia; es escalable y confiable; su estructura soporta grandes volúmenes de datos; verifica con ACID (Atomicidad, Consistencia, Integridad y Durabilidad de los datos); tiene integridad referencial; facilita la duplicidad de bases de datos maestras en varias zonas de réplica; aguanta lenguajes de programación como: C, C++, PHP, Perl, Python y Ruby, entre otros (Lopez Acevedo, 2013).

1.7.4 SQLServer.

Este gestor es propietario, por lo que posee un precio para su distribución. La misma hace uso de la plataforma de Windows, y se utiliza en lenguaje de programación asentado Visual Basic, Visual C++, Cobol entre otros. Entre las características relevantes que posee está: que es escalable, es multiusuario, provee la gestión, tiene una interfaz gráfica que facilita manejar los datos sin necesidad de grandes complicaciones.

Es potente en cuanto al manejo de grandes volúmenes de datos. Posibilita que no haya inconsistencia en los datos. Sus características avanzadas se centralizan en la integridad de la base de datos, como son los triggers, y ofrece soporte completo ACID (Atomicidad, Consistencia, Integridad y Durabilidad de los datos), además de optimizar las consultas; muestra seguridad en su código ya que las operaciones que están almacenadas quedan encriptados; y tiene un único identificador de login para la red (Batra, 2015).

En la siguiente tabla se muestra una comparación con características relevantes:

Tabla 4. Comparación de bases de datos

Características	Base de Datos			
	SQLServer	PosgreSQL	Oracle	MySQL
Licencia de software	Propietario	Libre	Propietario	Libre

Soporte de secuencias	Si	Si	Si	No
Multiplataforma	No	Si	Si	Si
SSL	Si	Si	Si	Si
Configuración	Si	Si	Si	Si

Fuente: Tesista

1.8 Metodología de desarrollo Ágiles.

1.8.1 Modelo Scrum: es una de las más utilizadas hoy en día ya que permite el trabajo colaborativo entre el equipo de trabajo y el cliente, permite las entregas gradualmente para la verificación del avance del proyecto, además de una estructuración flexible ante los posibles cambios e innovación. Para obtener buenos resultados se requiere de un gran esfuerzo por parte de todos los involucrados, ya que no hay un plan definido para todo el seguimiento, sino que un ajuste a las situaciones que se presente (Proyectos ágiles, 2017).

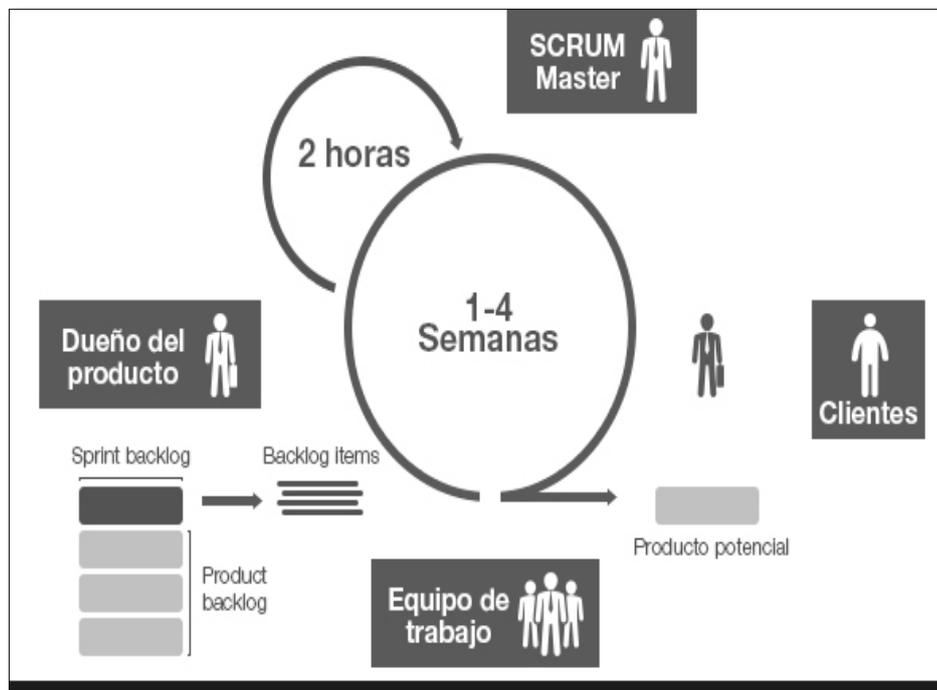


Figura 20. Representación de una iteración de Scrum

Fuente: <http://www.i2btech.com/blog-i2b/tech-deployment/para-que-sirve-el-scrum-en-la-metogologia-agil/>

1.8.2 Modelo de Programación Extrema (XP).

Es una metodología que se puede adaptar de acuerdo con las necesidades y requerimientos que se tengan que implementar, con la misma se puede hacer uso de cualquiera de los modelos que se han planteado anterior para el desarrollo, tiene como ventaja la libertad de adaptar el trabajo a cualquiera de los modelos (Meléndez Valladarez, Gaitan, & Pérez Reyes, 2016).

Permite la realización de pruebas unitarias continuas con el objetivo de hacer entregas del código sin errores. Posee como características la presencia de un desarrollo iterativo e incremental, la programación es en parejas, además permite refactorizar y simplificar el código. Roles que intervienen programador, cliente, especialistas de pruebas, especialistas de seguimiento (Tracker), entrenador (Coach), consultor, jefe del Proyecto (Mendes, Fco, & Rodriguez Cotorrelo, 2013).

La documentación en esta metodología no es necesaria, por lo que se va comentando todo el código que se está realizando, aunque de ser necesario se puede realizar si la solicita una de las partes.

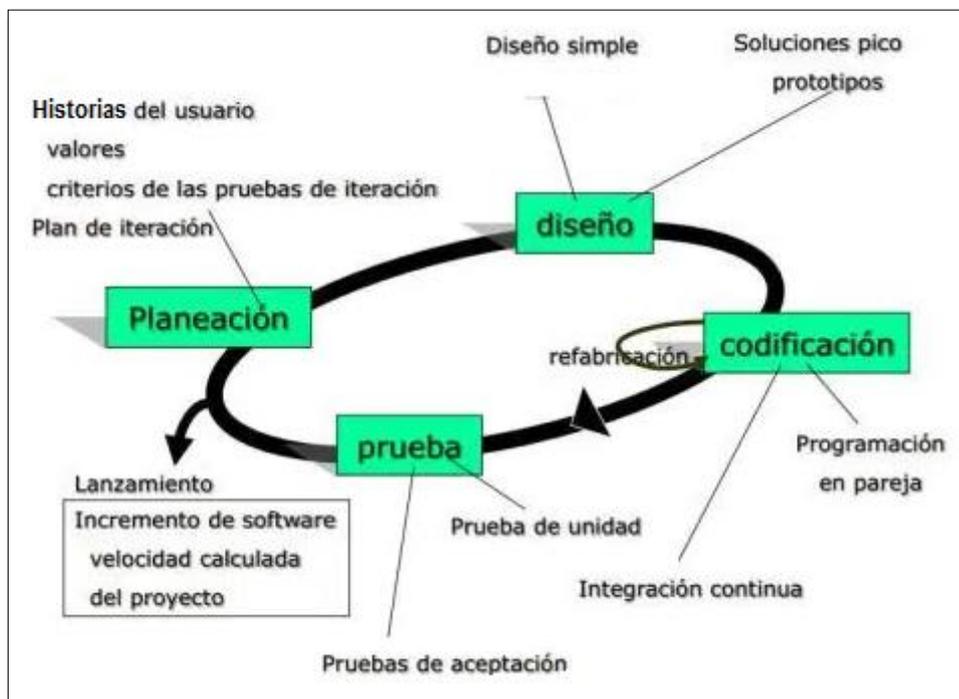


Figura 21. Ciclo de Programación Extrema

Fuente: <https://grupo13sistemaacademicoescolar.wordpress.com/2-3-metodologias/>

1.9 Entorno de Desarrollo Integrado(IDE).

Es un entorno de programación que ha sido empaquetado como un programa de aplicación, el cual cuenta con un editor de código, un compilador, un depurador y un constructor de interfaz gráfica (GUI). El mismo provee un marco de trabajo amigable para varios lenguajes de programación ejemplos de estos: C++, PHP, Python, Java, C#, Delphi, Visual Basic, entre otros.

Entre las características que presenta se encuentran:

- Multiplataforma
- Soporte para varios lenguajes de programación
- Integración con Sistemas de Control de Versiones
- Reconocimiento de Sintaxis
- Extensiones y Componentes para el IDE
- Integración con Framework
- Depurador
- Importar y Exportar proyectos
- Múltiples idiomas
- Manual de Usuarios y Ayuda

A continuación, se muestra un conjunto de IDE:

- Eclipse
- NetBeans
- IntelliJ IDEA
- JBuilder de Borland
- JDeveloper de Oracle
- KDevelop
- Anjuta
- Clarion
- MS Visual Studio
- Visual C++

1.9.1 Eclipse.

Software libre. Es uno de los entornos Java más utilizados a nivel profesional. El paquete básico de Eclipse se puede expandir mediante la instalación de plugins para añadir funcionalidades a medida que se vayan necesitando. Es multiplataforma y muy potente en la compilación y el desarrollo tanto de sitios web como se programas en Java o C++, además posee una interfaz agradable y fácil de usar (fergarcia, 2013).

1.9.2 NetBeans.

Software libre. Otro de los entornos Java muy utilizados, también expandible mediante plugins. Facilita bastante el diseño gráfico asociado a aplicaciones Java, se puede realizar cualquier tipo de aplicación, es fácil de instalar y actualizar. Presenta soporte para PHP (fergarcia, 2013).

1.9.3 Anaconda.

Es libre, es muy utilizado en la programación con Python, se pone en funcionamiento se instalan varias versiones, y otros paquetes para el tratamiento de grandes volúmenes de datos, cálculos científicos y análisis de predicciones (manejando datos, 2014).

La investigación hará uso del IDE Anaconda, por las facilidades que brinda al usarlo con Python.

1.10 Trabajos previos de reconocimiento facial.

Muchos son los trabajos que se han hecho en la actualidad sobre el reconocimiento facial, ejemplos de estos es el realizado por (Gimeno Hernández, 2010) en el que expone diferentes técnicas que se pueden tener en cuenta para el reconocimiento facial, en el mismo el detalla cada uno de ellos basándose en características esenciales que lo hacen diferenciarse de otros algoritmos y plasma ejemplos de cómo es el resultado de los mismos una vez aplicado.

Por otra parte, se encuentra el trabajo de (Bronte Palacios, 2008) en el que basado en algoritmos realiza un sistema para detectar y reconocer el rostro de determinados conductores a través de sistemas de visión computacional.

Además de las investigaciones antes mencionadas también se encuentra la realizada en México por (Villegas Quezada, 2005) en el que basado en técnicas que hacen uso de Matlab

se realiza el sistema para la detección del rostro y se realizan las pruebas con diferentes fotografías correspondiente a varias personas, las cuales se encontraban en bases de datos, arribando a las conclusiones de que dicha técnica en ocasiones tenía resultados satisfactorios y en otros se mostraban con dificultades, por lo que este método presenta dificultades y determinadas limitantes que impiden un buen reconocimiento facial.

METODOLOGÍA

La metodología de desarrollo a utilizar es la XP ya que como se mencionó en el Capítulo 1, sección 1.8.2 se puede adaptar de acuerdo con las necesidades y requerimientos que se tengan que implementar y la documentación en esta metodología no es necesaria, por lo que se va comentando todo el código que se está realizando.

Para el presente proyecto como anteriormente se ha mencionado se utilizará la librería OpenCV, cuya versión es la 2.4.11 y Python cuya versión es la 2.7.13.

2.1 Principales procesos identificados.

Ante la problemática planteada de la investigación se han identificado cuatro procesos fundamentales para dar solución a la misma y además de eso se ha planteado el esquema del sistema las cuales vamos a detallar:

- ❖ Los procesos fundamentales son los siguientes:
 - Adicionar usuario
 - Reconocimiento facial
 - Eliminar usuario
 - Entrenar el motor
- ❖ El esquema que vamos a desarrollar tiene los siguientes componentes como se indica en la Tabla 5:

Tabla 5. Componentes del esquema de desarrollo

Componente	Descripción
dataSet	Contiene las imágenes de los usuarios (base de datos). Las dimensiones de las imágenes van a depender de la cámara con la que se tomen las fotos, actualmente las misma contienen una dimensión de 350 pixeles * 350 pixeles.
trainer	Contenedor del archivo binario del motor de reconocimiento de openCV.

Fuente: Tesista

Estos componentes cuentan con varios archivos que se explicarán a continuación para un mejor entendimiento de los principales flujos. En la tabla 6 se muestra la descripción de cada uno de los archivos:

Tabla 6. Conjunto de archivos necesarios para el programa.

Nombre del Archivo	Descripción
trainer.yml	Archivo binario que contiene información sobre las imágenes en dataset a qué usuario pertenece.
dataset.py	Archivo python ejecutable que adiciona imágenes de usuarios a la base de datos (dataset).
config.conf	Archivo de configuración del sistema (parámetros configurables).
init.py	Archivo de constantes utilizados en la aplicación.
trainReconizer.py	Archivo Python ejecutable que crea el archivo trainer.yml, el mismo realiza el entrenamiento del reconocedor.
loadReconozzer.py:	Archivo Python ejecutable que se encarga de realizar el reconocimiento facial a partir de la detección realizada y guardada en el archivo trainer.yml
deletedataSet.py:	Archivo Python ejecutable que se encarga de eliminar todas las imágenes de la base de datos (dataset) o las imágenes de un usuario específico dado su id
haacascade_frontface_default.xml:	Archivo plantilla haarcascade de openCV que se utiliza para realizar la detección de rostros humanos presentes en un frame o imagen.

Fuente: Tesista

A continuación, se ejemplifican los procesos fundamentales a través de los siguientes diagramas de flujos:

2.1. 1 Adicionar usuario

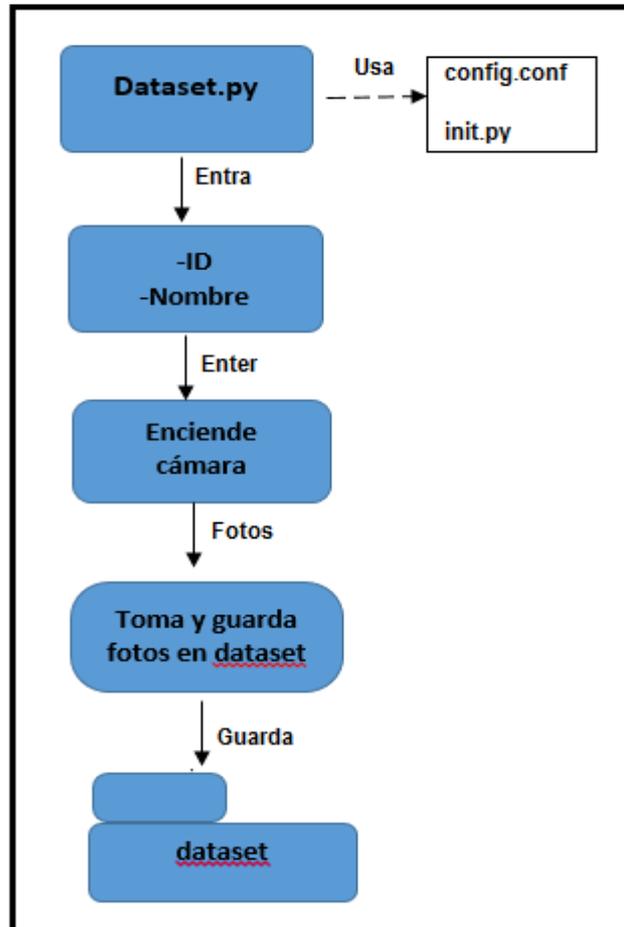


Figura 22. Flujo Adicionar usuario
Fuente: Tesista

2.1.2 Entrenar motor

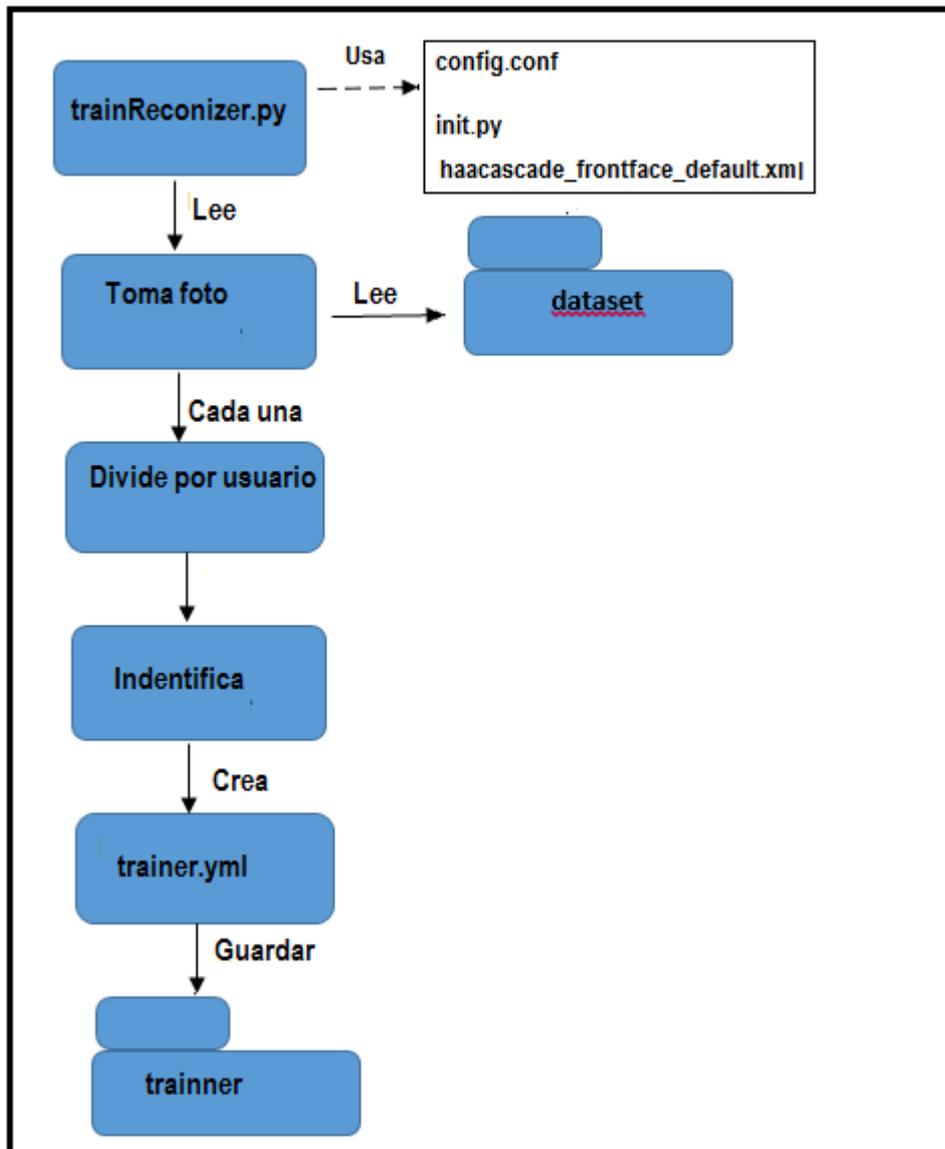


Figura 23. Flujo Entrenar motor
Fuente: Tesista

2.1.3 Reconocimiento facial

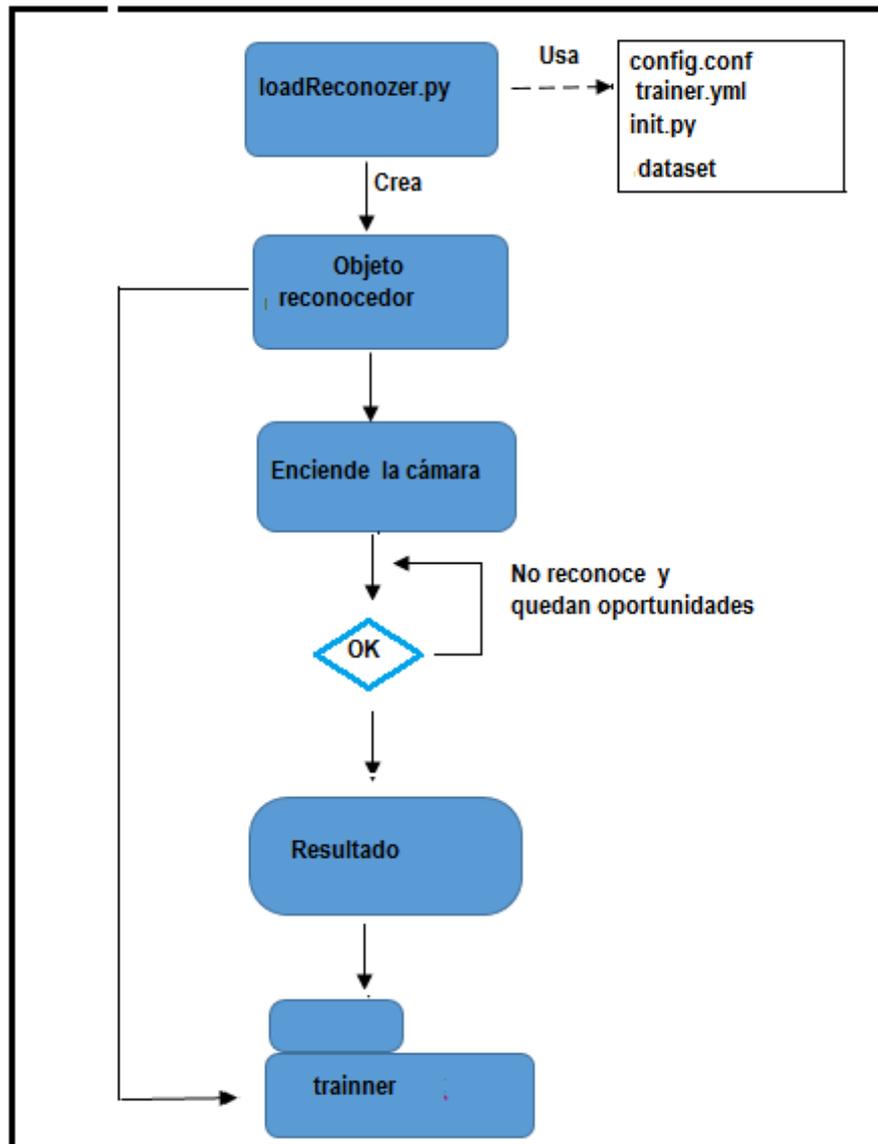


Figura 24. Flujo Reconocimiento Facial
Fuente: Tesista

2.1.4 Borrar Usuario.

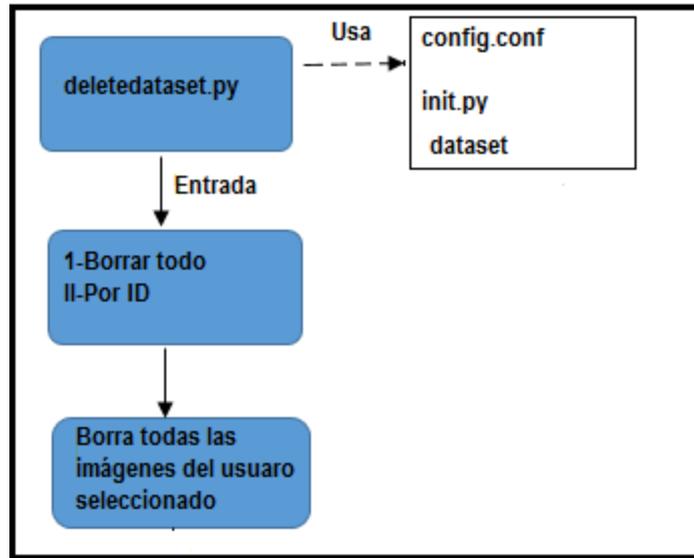


Figura 25. Flujo Borrar Usuarios
Fuente: Tesista

2.2 Metodología XP como metodología de desarrollo.

2.2.1 Planeación - Historias de usuario.

Con el objetivo de describir de manera resumida las funcionalidades que tendrá el sistema se plantean las siguientes Historias de Usuario(HU).

2.2.1.1 Historia de usuario Insertar usuario.

A continuación, se muestra en la tabla 7 la historia de usuario para insertar usuario.

Tabla 7. HU_Insertar usuario.

Historias de usuario	
Número: 001	Nombre HU: Insertar Usuario
Modificación o extensión de HU (Nombre o número): 1	
Usuario: Darío Javier Solís Alulima	Iteración asignada: 1
Prioridad en el negocio: Alto	Puntos estimados: 1
Riesgo en desarrollo: Bajo	
Descripción: tiene como opción insertar usuario una vez que se hayan ingresado los datos correspondientes al mismo, así como la captura de la imagen a la hora de adicionarse.	
Observaciones:	

Fuente: Tesista

2.2.1.2 Historia de usuario Eliminar usuario.

A continuación, se muestra en la tabla 8 la historia de usuario para eliminar un usuario.

Tabla 8. HU_ Eliminar usuario

Historias de usuario	
Número: 2	Nombre HU: Eliminar Usuario
Modificación o extensión de HU (Nombre o número): 1	
Usuario: Darío Javier Solís Alulima	Iteración asignada: 1
Prioridad en el negocio: Alto	Puntos estimados: 1
Riesgo en desarrollo: Bajo	
Descripción: tiene como opción eliminar usuario a través de dos condiciones: la de ingresar 1 para que borre todas las imágenes contenidas en la base de datos, y la otra presionar cualquier tecla para que te de la opción de ingresar el identificador del usuario que de seas eliminar.	
Observaciones:	

Fuente: Tesista

2.2.1.3 Historia de usuario Reconocimiento facial

A continuación, se muestra en la tabla 9 la historia de usuario para reconocimiento facial a un usuario.

Tabla 9. HU_ Reconocimiento facial

Historias de usuario	
Número: 003	Nombre HU: Reconocimiento facial
Modificación o extensión de HU (Nombre o número): 1	

Usuario: Darío Javier Solís Alulima	Iteración asignada: 1
Prioridad en el negocio: Alto	Puntos estimados: 4
Riesgo en desarrollo: Bajo	
Descripción: se basa en ejecutar el archivo de reconocimiento para identificar si el usuario que está accediendo está registrado en la base de datos, de no reconocerlo en sus inicios el proceso se va a ejecutar mientras le queden oportunidades.	
Observaciones:	

Fuente: Tesista

2.2.1.4 Historia de usuario Entrenar motor.

A continuación, se muestra en la tabla 10 la historia de usuario para entrenar el motor del sistema.

Tabla 10. HU_ Entrenar motor

Historias de usuario	
Número: 1	Nombre HU: Entrenar motor
Modificación o extensión de HU (Nombre o número): 1	
Usuario: Darío Javier Solís Alulima	Iteración asignada: 1
Prioridad en el negocio: Alto	Puntos estimados: 4
Riesgo en desarrollo: Bajo	
Descripción: verifica que la foto está en base de dato, y por cada foto que tome se la asigna a un usuario, y luego crea el ejecutable trainer para que realice el entrenamiento del reconocedor.	
Observaciones:	

Fuente: Tesista

2.3 Planeación – Criterios de aceptación.

Para validar ciertos criterios de aceptación en el sistema, se plantean varios requerimientos de alto nivel que se consideran especificaciones importantes.

A continuación, se muestra en la tabla 11 los requisitos y los criterios de aceptación para la aplicación

Tabla 11. Requisitos y criterios de aceptación.

Requisitos	Criterios de aceptación
Entrenar motor	Permite verificar si la imagen se encuentra en la base de dato y genera un ejecutable para el entrenamiento del reconocedor
Adicionar usuario	Permite adicionar usuario con datos específico y tener adjunto a este un conjunto de imágenes asociadas para su próximo reconocimiento a la hora de acceder al local
Eliminar usuario	Permite la eliminación de todas las imágenes que se encuentran en la base de datos, o la eliminación de un usuario específico mediante du ID (identificador)
Reconocimiento facial	Permita reconocer el rostro de la persona si es que está en base de datos, y en caso de que no esté no permitirle el acceso

Fuente: Tesista

2.4 Parámetros necesarios para el Reconocimiento Facial.

Para obtener los resultados esperados, el sistema debe cumplir con la función de reconocer los distintos rostros de los usuarios ingresados y configurados, por lo que se hace necesario determinar cuáles son los parámetros que debe cumplir el mismo.

A continuación, se describe cada parámetro para determinar los valores que van hacer usados para calibrar el sistema a través de las pruebas y el análisis del funcionamiento.

2.4.1 Nivel de confianza de Predicción en la aplicación.

La librería Open contiene la función **FaceRecognizer: predict ()** la cual se describe en OpenCV 2.4.13.2 documentation como que la misma: “predice una etiqueta y la confianza asociada (por ejemplo, distancia) para una imagen de entrada dada”, a modo de resumen lo que hace es crear un mapa de descriptores para cada rostro que se ingresa y se entrena al algoritmo. Sucesivamente para cada referencia de datos se genera una medida de distancia entre la imagen de entrada que vendría siendo el nuevo rostro por identificar por el sistema y la base de datos existente.

La medida de distancia no es más que la distancia euclidiana entre los puntos que representan el rostro en las dimensiones de eigenspace. Cada persona se representa como una combinación de los diferentes eigenfaces (cada uno crea una dimensión).

Para el sistema si la distancia es igual a cero, significa una coincidencia exacta del reconocimiento facial, mientras que los valores superiores a 30 (valor de referencia obtenido en las pruebas del algoritmo, Capítulo 3, sección 3.1.1) significa que casi no hay coincidencia entre ambos.

2.4.2 Distancia de Reconocimiento.

Es aquella distancia o rango que permite ubicar un objeto que en el caso del presente proyecto es un rostro para posteriormente procesar la imagen dentro del sistema e identificarlo.

Para el sistema el rango de distancia de reconocimiento está entre los 20 y 30 cm. (valores de referencia obtenidos en las pruebas del algoritmo, Capítulo 3, sección 3.1.2).

2.4.3 Nivel de Iluminación.

La iluminancia o nivel de iluminación se define como el flujo luminoso que incide sobre una superficie. Su unidad de medida es el Lux.

Para el sistema se debe definir qué es lo que se entiende por nivel de iluminación alta y baja, ya que para las pruebas de funcionamiento se realizó baja estos dos parámetros para determinar finalmente cual es la indicada para el sistema.

El nivel de iluminación baja y alta en del proyecto está determinado por el factor del entorno físico donde se encuentra el sistema instalado, en este caso, el sistema se encuentra ubicado en un sitio donde la iluminancia o nivel de iluminación natural es muy baja tomando en cuenta la tabla 12 donde se indica los niveles normales de iluminancia.

Tabla 12. Niveles de Luz natural en el día.

Condición	Iluminancia (lux)
Luz de sol	107,527
Plena luz del día	10,752
Día nublado	1,075
Día oscuro	107
Crepúsculo	10.8
Noche nublada	0.0001

Fuente: Tesista

Por lo que en el sitio se debe utilizar permanentemente luz artificial en los rangos recomendados para un sitio de operaciones de soporte técnico con equipos electrónicos (PC, Wifi, servidores) entre 500 a 700 Lux.

Con lo anteriormente mencionado podemos concluir que para nuestro sitio de trabajo en el cual está instalado el sistema de reconocimiento facial lo siguiente:

- **Nivel de iluminación Baja:** es la que se encuentra dentro del rango normal actualmente utilizando el entorno físico del sistema.
- **Nivel de iluminación Alta:** se logra con ayuda de una lampara artificial fluorescente de aproximadamente 200 lux adicionales a los que ya se encuentra trabajando el entorno físico del sistema.

Para el sistema se concluyó que debe tener un Nivel de iluminación Alta (resultados obtenidos en las pruebas del algoritmo, Capítulo 3, sección 3.1.3)

PRUEBAS Y RESULTADOS DE FUNCIONAMIENTO

3.1 Calibración del Sistema.

A continuación, mostraremos las pruebas realizadas para cada parámetro con el fin de determinar los valores necesarios para un correcto funcionamiento del sistema:

3.1.1 Nivel de Confianza.

Para poder determinar el nivel de confianza de predicción para el sistema de reconocimiento facial en el que se está trabajando y que haya seguridad de que el usuario al cual va ser reconocido se obtenga la certeza que es el correcto, se realizó varias pruebas de funcionamiento variando los niveles de confianza, de dichas variaciones se obtuvo los siguientes resultados.

A continuación, la tabla 13 y la figura 26 muestra los resultados obtenidos.

Tabla 13. Pruebas de funcionamiento variando los niveles de confianza.

Nivel de Confianza	Porcentaje de Aciertos (%)
0 - 10	8
11 - 20	20
21 - 25	45
26 - 30	92
31 - 35	24
36 - 40	15
41 - 45	0
mayor a 45	0

Fuente: Tesista

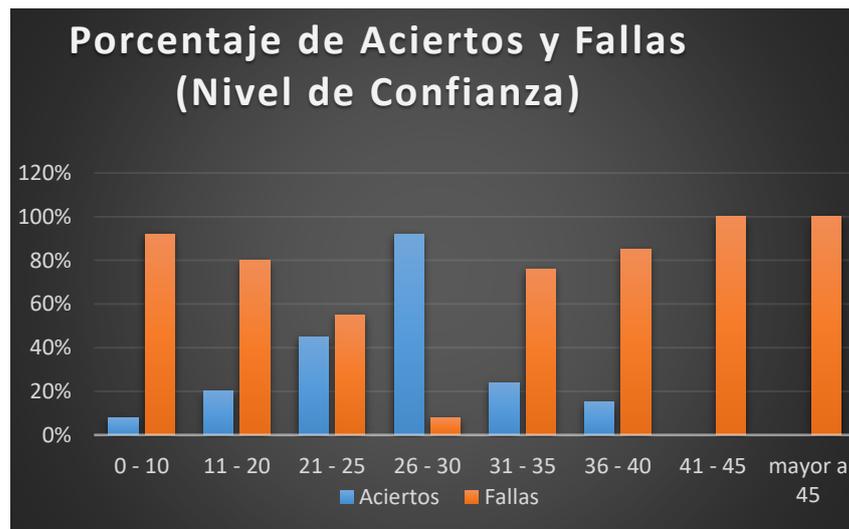


Figura 26. Porcentaje de aciertos y fallas en el nivel de confianza
Fuente: Tesista

Después del análisis realizado se llega a la conclusión que el sistema trabaja con un nivel de confianza entre 25 y 30, ya que si se configura el sistema; si se pasa de este rango, tiende a presentar problemas de reconocimiento, proporcionando datos errados de usuarios, dicho de otra manera, confunde usuarios uno de otros o como usuarios desconocidos.

3.1.2 Distancias para Reconocimientos.

Para determinar la distancia adecuada se ha realizado pruebas de funcionamiento del sistema a diferentes rangos de distancia, para un total de 25 intentos se obtuvieron los siguientes resultados.

A continuación, la tabla 14 y en la figura 27 muestra los resultados obtenidos.

Tabla 14. Distancia para el reconocimiento

Distancia	Porcentaje de aciertos
20-24 cm	88
25-29 cm	96
30-34 cm	40
35-39 cm	4
40-44 cm	0
45-50 cm	0

Fuente: Tesista

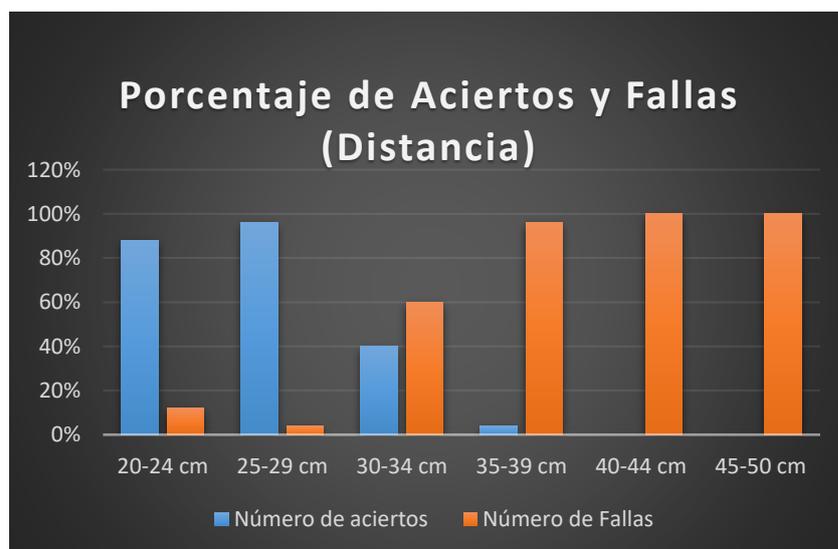


Figura 27. Porcentaje de aciertos y falla en la distancia

Fuente: Tesista

Determinando así que el rango que se debe utilizar para el sistema de reconocimientos está entre los **20 a 30 cm** de distancia a la cámara.

3.1.3 Nivel de Iluminación.

Se realizó pruebas de funcionamiento con un nivel de iluminación baja y alta en el entorno físico seleccionado para el funcionamiento del sistema, además de una distancia determinada con anterioridad que garantizaba el reconocimiento, una vez realizado se obtuvo los siguientes resultados.

A continuación, en la tabla 15, 16 y en la figura 28 y 29 se muestra el caso de prueba para Reconocer un Usuario.

Tabla 15. Luminosidad baja

Nivel de iluminación Baja	Número de Aciertos	Número de Fallas
20-30 cm	4	21

Fuente: Tesista



Figura 28. Luminosidad baja
Fuente: Tesista

Tabla 16. Luminosidad alta.

Nivel de iluminación Alta	Número de Aciertos	Número de Fallas
20-30 cm	23	2

Fuente: Tesista



Figura 29. Luminosidad alta
Fuente: Tesista

Con lo cual se llega a determinar que su funcionamiento se lo debe hacer en ambiente donde el Nivel de Iluminación debe ser alta para garantizar el funcionamiento del sistema.

3.2 Pruebas - Pruebas de aceptación.

Para la verificación del funcionamiento de la aplicación, se hizo un conjunto de pruebas de aceptación que permitió validar las posibles entradas y salidas en base a las funcionalidades.

3.2.1 Caso de prueba: Adicionar usuario.

A continuación, la tabla 17 muestra el caso de prueba para Adicionar un Usuario.

Tabla 17. CP_Adicionar_Usuario

Nombre del Caso de prueba	Adicionar usuario
Identificador caso de prueba	CP_Adicionar_Usuario.
Función probar	Adicionar usuario a través de la aplicación
Descripción	Verificar que la funcionalidad se ejecute correctamente y que el usuario se adicione

Criterios de éxito	Que los datos introducidos estén correctamente, es decir cumplan con la validación del código		
Criterios de falla	Que se introduzcan datos incorrectos		
Precondiciones			
Perfil del usuario	Administrador		
Necesidades para el caso de prueba	Debe estar el ambiente de la aplicación en ejecución		
Autor	Darío Javier Solís Alulima		
Fecha de creación	23/07/2017		
Flujo del caso de prueba	No paso	Usuario del sistema	Sistema
	1	Ingresar nombre y cédula de identidad	El sistema realiza la toma de captura de imágenes
	2	Ingresar nombre o cédula de identidad de manera incorrecta	El sistema no realiza la toma de captura de imágenes
Resultado de la prueba	Satisfactoria		

Fuente: Tesista

3.2.2 Caso de prueba: Eliminar usuario.

A continuación, la tabla 18 muestra el caso de prueba para Eliminar un Usuario.

Tabla 18. CP_Eliminar_usuario

Nombre del Caso de prueba	Eliminar usuario		
Identificador caso de prueba	CP_Eliminar_usuario		
Función probar	Eliminar usuario a través de la aplicación		
Descripción	Verificar que la funcionalidad se ejecute correctamente y que el usuario se elimine		
Criterios de éxito	Que el usuario se elimine de la base de datos		
Criterios de falla	Que no se elimine el usuario de la base de datos		
Precondiciones	Que el usuario exista en la base de datos		
Perfil del usuario	Administrador		
Necesidades para el caso de prueba	Debe estar el ambiente de la aplicación en ejecución		
Autor	Darío Javier Solís Alulima		
Fecha de creación	23/07/2017		
Flujo del caso de prueba	No paso	Usuario del sistema	Sistema

	1	Seleccionar el usuario a eliminar	El sistema borra el usuario seleccionado
	2	Escoger borrar todos los usuarios de la base de datos	El sistema borra todos los usuarios de la base de datos.
Resultado de la prueba	Satisfactoria		

Fuente: Tesista

3.2.3 Caso de prueba: Reconocer usuario.

A continuación, la tabla 19 muestra el caso de prueba para Reconocer un Usuario.

Tabla 19. CP_ Reconocer_usuario

Nombre del Caso de prueba	Reconocer usuario
Identificador caso de prueba	CP_ Reconocer_usuario
Función probar	Reconocimiento del usuario
Descripción	Verificar que se realice el reconocimiento correctamente
Criterios de éxito	Que reconozca el usuario si este se encuentra registrado
Criterios de falla	Que no reconozca el usuario aun cuando este esté en la base de datos
Precondiciones	Que el usuario exista en la base de datos

Perfil del usuario	Administrador		
Necesidades para el caso de prueba	Debe estar el ambiente de la aplicación en ejecución		
Autor	Darío Javier Solís Alulima		
Fecha de creación	23/07/2017		
Flujo del caso de prueba	No paso	Usuario del sistema	Sistema
	1	Si reconoce el rostro	El sistema confirma el acceso
	2	Si no reconoce el rostro	El sistema informa que el usuario no está autorizado.
Resultado de la prueba	Satisfactoria		

Fuente: Tesista

3.3 Pruebas y Resultados Obtenidos.

3.3.1 Usuarios de Prueba.

Las pruebas de funcionamiento se realizaron con aproximadamente 25 usuarios. Los mismos que pertenecen al departamento de Redes de la Pontificia Universidad Católica del Ecuador con la finalidad de validar el presente trabajo de investigación, todas las pruebas se las realizaron durante 10 días laborales entre la semana del 17 al 21 de julio y la semana del 24 al 29 de julio del presente año.

A continuación, la tabla 20 muestra los usuarios que participaron en las pruebas de funcionamiento.

Tabla 20. Registro de algunos Usuarios en pruebas.

ID	Apellidos	Nombres	Cargo	Horario de entrada	Horario de salida	Foto
1717186918	Solís Alulima	Darío Javier	Programador	10.30	19.00	
1709287708	Zumárraga Lozada	Germán Vinicio	Jefe de Cableado estructurado	8.00	17.00	
1711005031	Zapata Aguinaga	Raúl Gerardo	Analista en redes y sistemas web	8.00	17.00	
1712760518	Pérez Jurado	Marcelo	Analista en redes y sistemas web	8.00	17.00	
1712940830	Albán Lituma	Santiago Alejandro	Jefe de Área	7.00	17.00	
1713160750	Quezada Marín	Pablo	Analista en redes y sistemas web	8.00	17.00	

Fuente: Tesista

Para saber el procedimiento a seguir por parte de los usuarios se ha creado un manual para la realización de las pruebas, mismo que se encuentra adjunto en el anexo 1.

3.3.2 Entorno Físico de Pruebas.

En las pruebas realizadas se utilizó un entorno físico el cual se validó que debe tener dos características principales que es una iluminación Alta y una distancia adecuada, ambos parámetros fueron especificados anteriormente. La cámara se encuentra en un lugar fijo donde los usuarios deben acercarse y realizar el procedimiento para el reconocimiento y su posterior registro.

A continuación, en la figura 30 se muestra el entorno físico donde se encuentra ubicado el sistema de reconocimiento facial.



Figura 30. Entorno físico de trabajo
Fuente: Tesista

3.3.3. Pruebas con los parámetros de funcionamiento.

Para evaluar el sistema se debe tomar en cuenta los 3 parámetros de funcionamiento que son: Nivel de Confianza, Distancia de Reconocimiento y Nivel de Iluminación, dando como resultado ocho posibles escenarios a probar tal como se muestra en la tabla 21.

Tomando en cuenta que los valores determinados para que funcione correctamente el sistema (resultados obtenidos en las pruebas del algoritmo, Capítulo 3, sección 3.1) son los siguientes:

- **Nivel de Confianza:** entre 25 a 30.
- **Distancia de Reconocimiento:** entre 20 a 30 cm.
- **Nivel de Iluminación:** Alto.

Tabla 21. Posibles escenarios para las pruebas

Escenarios	Nivel de confianza	Distancia de Reconocimiento	Nivel de Iluminación
1	> 30	> 30 cm.	Bajo
2	> 30	> 30 cm.	Alto
3	> 30	entre 20 a 30 cm.	Bajo

4	> 30	entre 20 a 30 cm.	Alto
5	entre 25 a 30	> 30 cm.	Bajo
6	entre 25 a 30	> 30 cm.	Alto
7	entre 25 a 30	entre 20 a 30 cm.	Bajo
8	entre 25 a 30	entre 20 a 30 cm.	Alto

Fuente: Tesista

Una vez determinados los posibles escenarios se realizaron las pruebas de funcionamiento de cada escenario, teniendo los siguientes resultados:

❖ Escenario número 1.

A continuación, tabla 22 y la figura 31 muestra resultados del escenario número 1

Tabla 22. Resultados del Escenario 1

Nombres de Usuarios	ID	Días de Aciertos entre los 10 de pruebas	Promedio de Nivel de Confianza	Porcentaje de Aciertos (%)
Darío Javier Solís Alulima	1717186918	1	29.45	10
Germán Vinicio Zumárraga	2	0	45.3	0
Raúl Gerardo Zapata Aguinaga	1711005031	0	38.4	0
Marcelo Pérez Jurado	1712760518	1	28.9	10
Santiago Alejandro Albán Lituma	1712940830	0	37.9	0
Pablo Quezada Marín	1713160750	0	41.3	0

Fuente: Tesista



Figura 31. Resultados Escenario 1

Fuente: Tesista.

❖ **Escenario número 2.**

A continuación, tabla 23 y figura 32 muestra resultados del escenario número 2.

Tabla 23. Resultados del Escenario 2

Nombres de Usuarios	ID	Días de Aciertos entre los 10 de pruebas	Promedio de Nivel de Confianza	Porcentaje de Aciertos (%)
Darío Javier Solís Alulima	1717186918	0	35.2	0
Germán Vinicio Zumárraga	2	1	28.8	10
Raúl Gerardo Zapata Aguinaga	1711005031	1	29.5	10
Marcelo Pérez Jurado	1712760518	0	39.7	0
Santiago Alejandro Albán Lituma	1712940830	0	41.4	0
Pablo Quezada Marín	1713160750	0	38.6	0

Fuente: Tesista

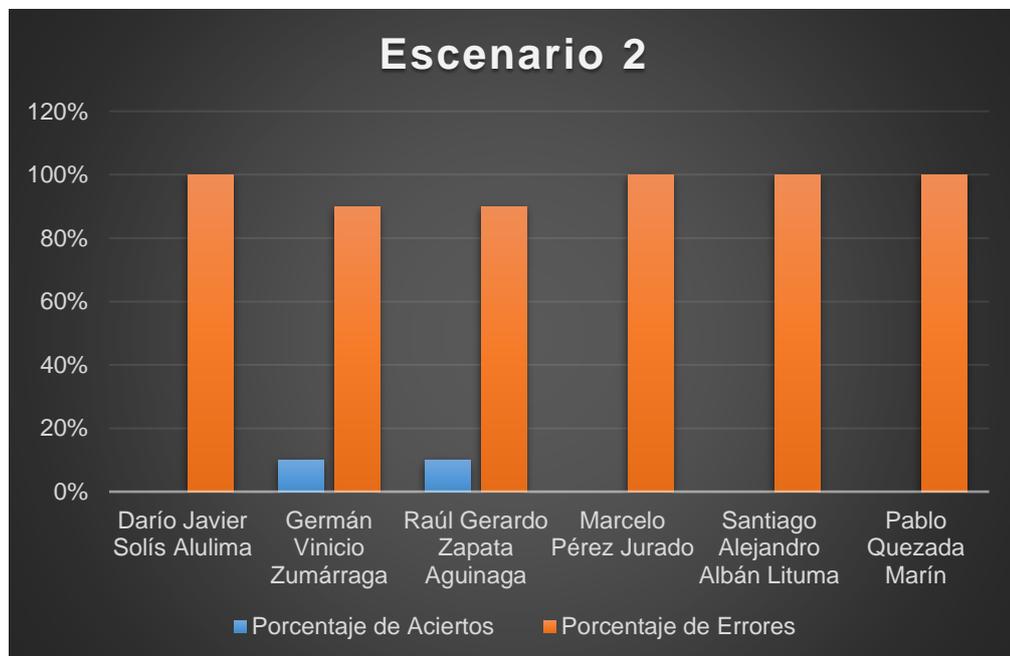


Figura 32. Resultados Escenario 2

Fuente: Tesista

❖ **Escenario número 3.**

A continuación, tabla 24 y figura 33 muestra resultados del escenario número 3.

Tabla 24. Resultados del Escenario 3

Nombres de Usuarios	ID	Días de Aciertos entre los 10 de pruebas	Promedio de Nivel de Confianza	Porcentaje de Aciertos (%)
Darío Javier Solís Alulima	1717186918	0	41.2	0
Germán Vinicio Zumárraga	2	0	37.8	0
Raúl Gerardo Zapata Aguinaga	1711005031	0	39.7	0
Marcelo Pérez Jurado	1712760518	0	45.3	0
Santiago Alejandro Albán Lituma	1712940830	0	39.9	0
Pablo Quezada Marín	1713160750	1	28.7	10

Fuente: Tesista

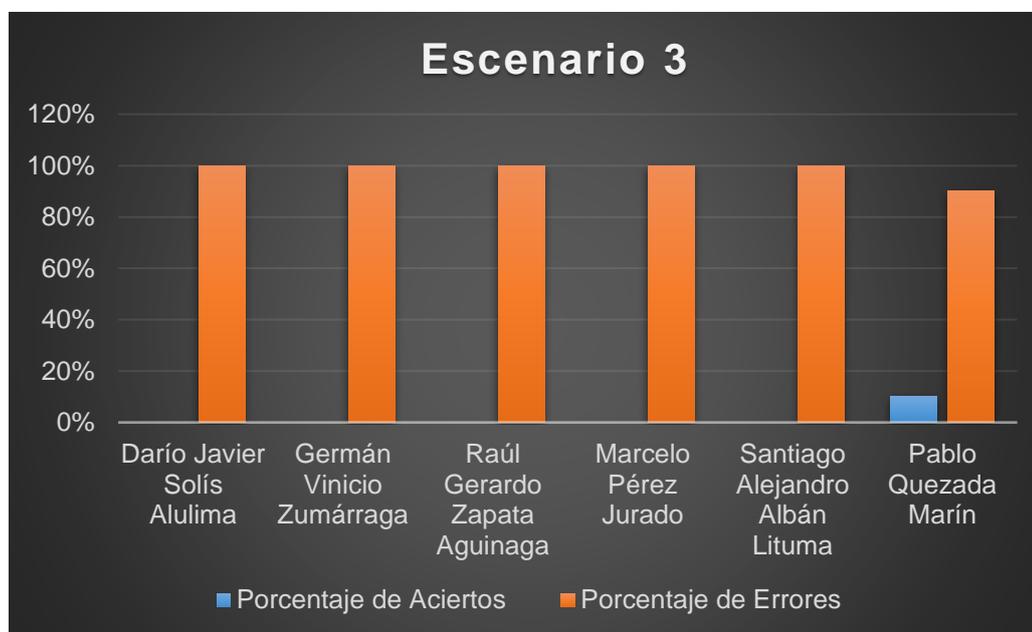


Figura 33. Resultados Escenario 3

Fuente: Tesista

❖ **Escenario número 4.**

A continuación, tabla 25 y figura 34 muestra resultados del escenario número 4.

Tabla 25. Resultados del Escenario 4

Nombres de Usuarios	ID	Días de Aciertos entre los 10 de pruebas	Promedio de Nivel de Confianza	Porcentaje de Aciertos (%)
Darío Javier Solís Alulima	1717186918	2	29.7	20
Germán Vinicio Zumárraga	2	1	28.8	10
Raúl Gerardo Zapata Aguinaga	1711005031	2	29.6	20
Marcelo Pérez Jurado	1712760518	3	28.5	30
Santiago Alejandro Albán Lituma	1712940830	3	27.9	30
Pablo Quezada Marín	1713160750	2	27.7	20

Fuente: Tesista

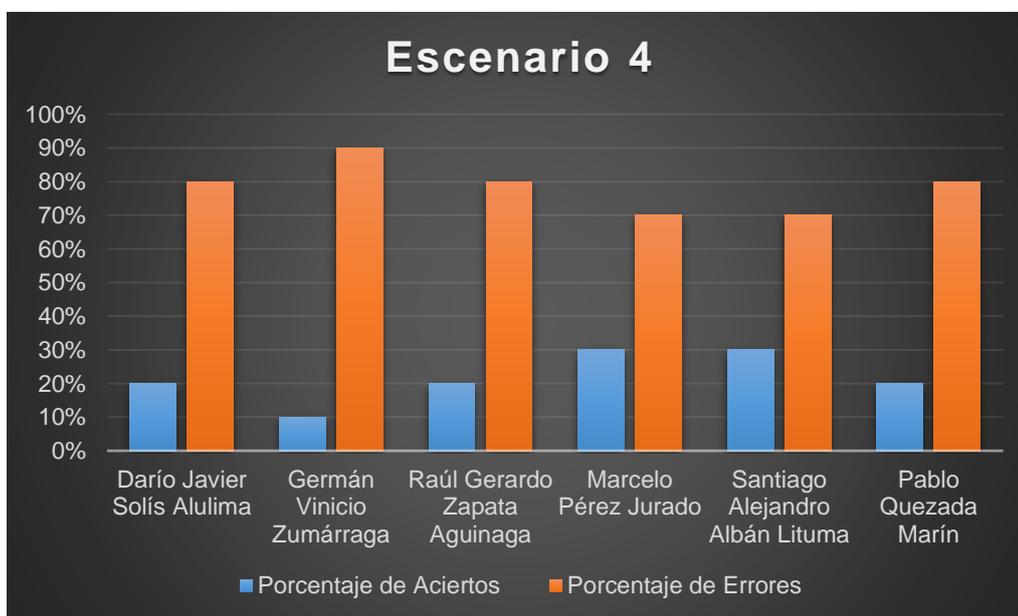


Figura 34. Resultados Escenario 4

Fuente: Tesista

❖ **Escenario número 5.**

A continuación, tabla 26 y figura 35 muestra resultados del escenario número 5.

Tabla 26. Resultados del Escenario 5

Nombres de Usuarios	ID	Días de Aciertos entre los 10 de pruebas	Promedio de Nivel de Confianza	Porcentaje de Aciertos (%)
Darío Javier Solís Alulima	1717186918	0	38.9	0
Germán Vinicio Zumárraga	2	1	29.7	10
Raúl Gerardo Zapata Aguinaga	1711005031	0	33.7	0
Marcelo Pérez Jurado	1712760518	0	37.0	0
Santiago Alejandro Albán Lituma	1712940830	0	38.6	0
Pablo Quezada Marín	1713160750	0	34.9	0

Fuente: Tesista



Figura 35. Resultados Escenario 5

Fuente: Tesista

❖ **Escenario número 6.**

A continuación, tabla 27 y figura 36 muestra resultados del escenario número 6.

Tabla 27. Resultados del Escenario 6

Nombres de Usuarios	ID	Días de Aciertos entre los 10 de pruebas	Promedio de Nivel de Confianza	Porcentaje de Aciertos (%)
Darío Javier Solís Alulima	1717186918	4	27.8	40
Germán Vinicio Zumárraga	2	3	29.1	30
Raúl Gerardo Zapata Aguinaga	1711005031	4	28.0	40
Marcelo Pérez Jurado	1712760518	5	26.8	50
Santiago Alejandro Albán Lituma	1712940830	2	28.9	20
Pablo Quezada Marín	1713160750	5	27.7	50

Fuente: Tesista

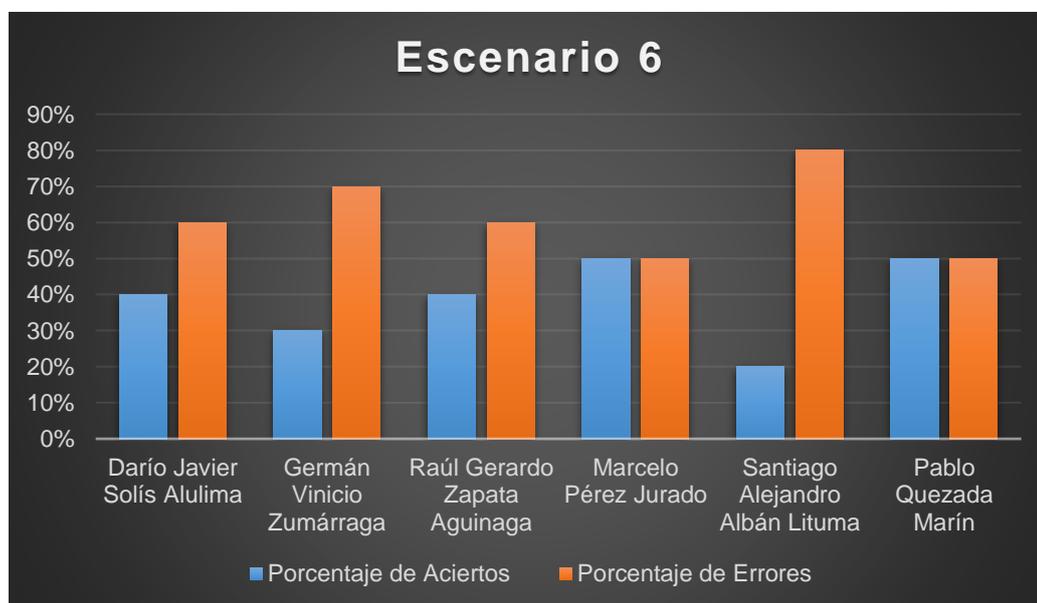


Figura 36. Resultados Escenario 6

Fuente: Tesista

❖ **Escenario número 7.**

A continuación, tabla 28 y figura 37 muestra resultados del escenario número 7.

Tabla 28. Resultados del Escenario 7

Nombres de Usuarios	ID	Días de Aciertos entre los 10 de pruebas	Promedio de Nivel de Confianza	Porcentaje de Aciertos (%)
Darío Javier Solís Alulima	1717186918	2	28.9	20
Germán Vinicio Zumárraga	2	3	28.2	30
Raúl Gerardo Zapata Aguinaga	1711005031	2	27.1	20
Marcelo Pérez Jurado	1712760518	3	29.1	30
Santiago Alejandro Albán Lituma	1712940830	3	27.7	30
Pablo Quezada Marín	1713160750	2	28.0	20

Fuente: Tesista

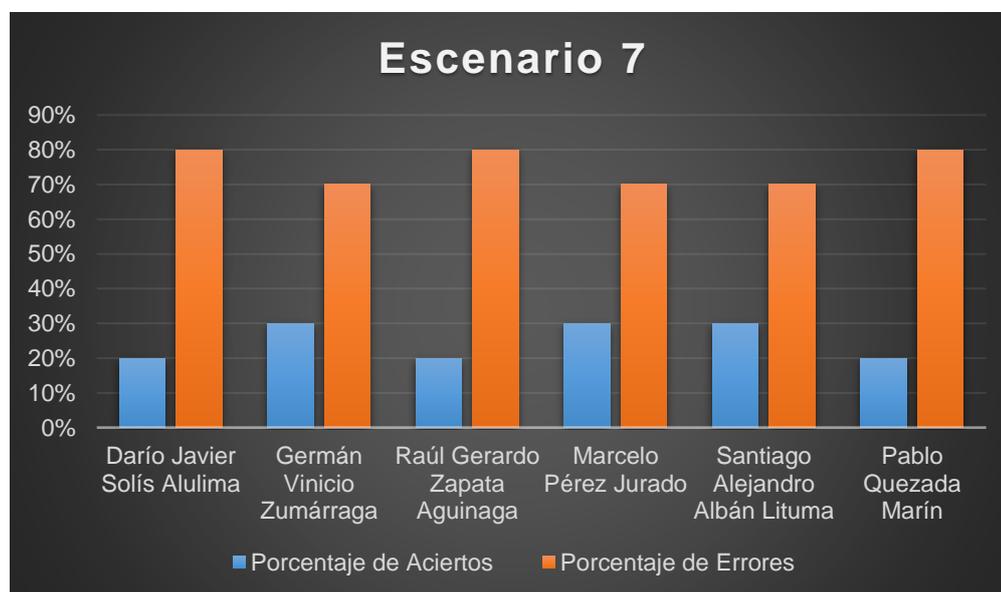


Figura 37. Resultados Escenario 7

Fuente: Tesista

❖ **Escenario número 8.**

A continuación, tabla 29 y figura 38 muestra resultados del escenario número 8.

Tabla 29. Resultados del Escenario 8

Nombres de Usuarios	ID	Días de Aciertos entre los 10 de pruebas	Promedio de Nivel de Confianza	Porcentaje de Aciertos (%)
Darío Javier Solís Alulima	1717186918	10	16.9	100
Germán Vinicio Zumárraga	2	9	17.1	90
Raúl Gerardo Zapata Aguinaga	1711005031	10	18.2	100
Marcelo Pérez Jurado	1712760518	10	17.4	100
Santiago Alejandro Albán Lituma	1712940830	9	17.1	90
Pablo Quezada Marín	1713160750	10	12.9	100

Fuente: Tesista

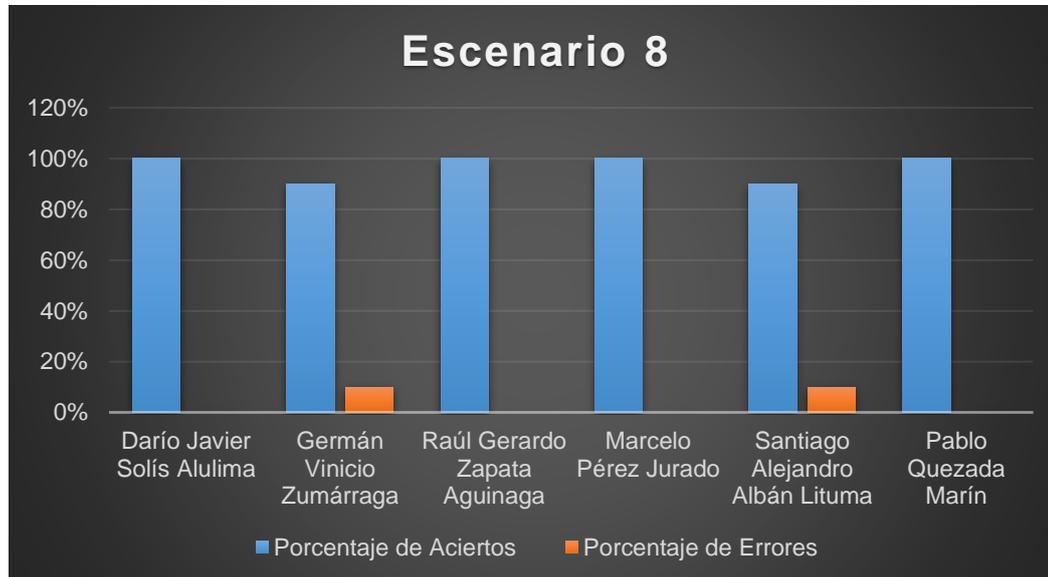


Figura 38. Resultados Escenario 8

Fuente: Tesista

3.3.4 Exactitud del sistema ante los diferentes escenarios.

Una vez concluidos las pruebas de funcionamiento en los diferentes escenarios posibles se puede observar en la figura 39 como resultado final que el mejor escenario es el número 8 debido a que cumple con los valores determinados de cada uno de los tres parámetros (resultados obtenidos en las pruebas del algoritmo, Capítulo 3, sección 3.1), mientras que en el escenario 4 y 6 tiende a mejorar el porcentaje de reconocimiento debido al parámetro del nivel de iluminación es Alto y esto es un factor muy importante para el funcionamiento del sistema.

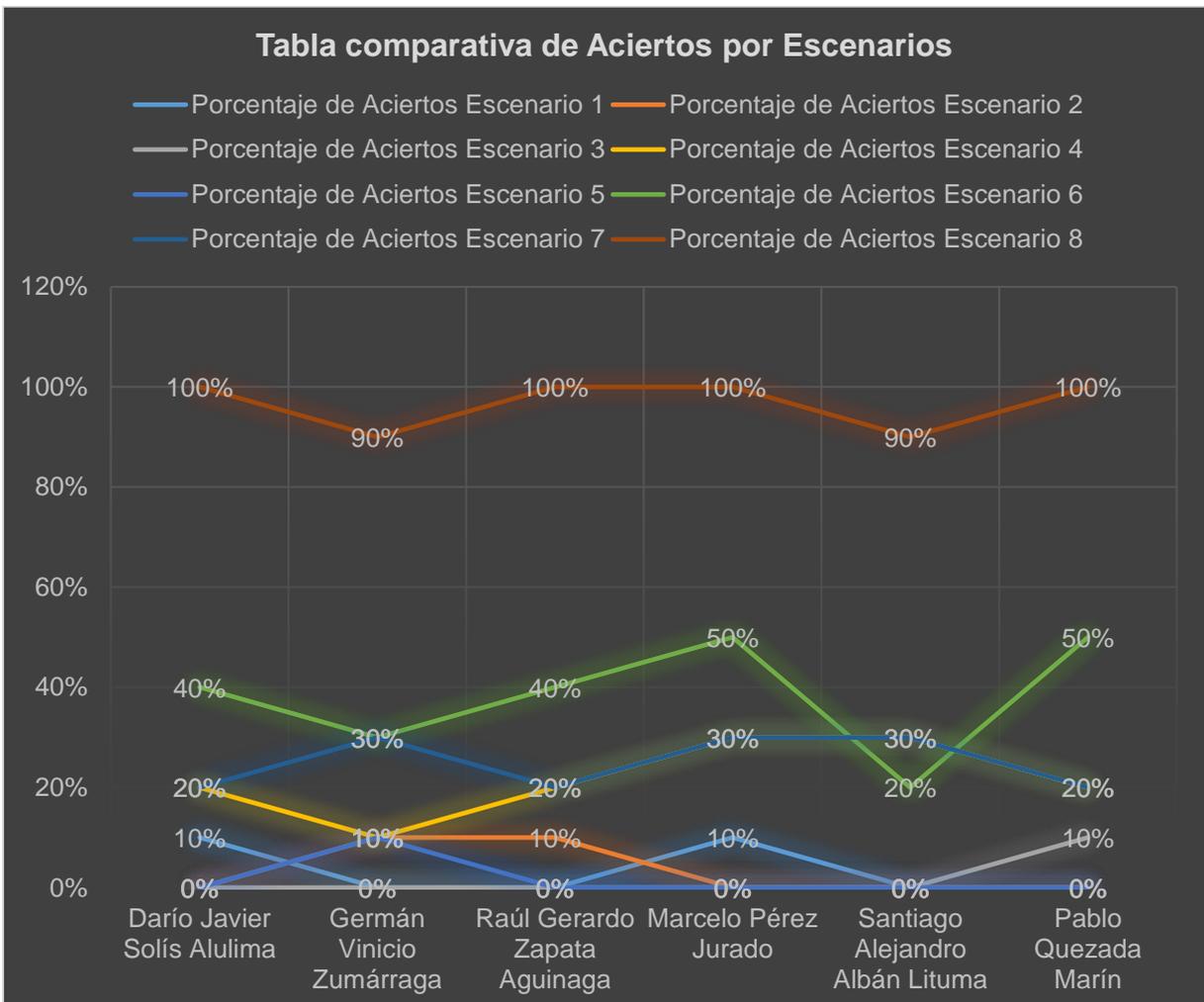


Figura 39. Tabla comparativa en los escenarios propuestos
Fuente: Tesista

CONCLUSIONES

- Para la realización del presente tema de investigación se hizo el estudio de varios algoritmos que se utilizan para el reconocimiento facial, además de sistemas que se usan para la restricción de acceso a un lugar determinado a nivel mundial.
- El sistema de reconocimiento facial está basado en el algoritmo LBPH (Histogramas de Patrones Binarios Locales) la cual se validó y se la puede ver en la tabla 2 de la sección 1.4 porque facilita el entrenamiento a partir de pocas imágenes y no es necesario tener una gran base de datos como lo requieren otros algoritmos como: EigenFaces, FisherFaces que de igual forma son propios de OpenCv.
- Los parámetros utilizados en la investigación son: Nivel de Confianza, Distancia de reconocimiento y Nivel de Iluminación, mismos que fueron determinados mediante las pruebas de funcionamiento las cuales se puede revisar en el capítulo 3, sección 3.1.
- Se determinó que el Nivel de Confianza más óptimo para el presente trabajo en base a las pruebas realizadas de reconocimiento se encuentra entre el rango de 25 a 30. (Ver en Capítulo 3, sección 3.1.1.).
- La distancia para garantizar un eficiente reconocimiento se estableció mediante las pruebas realizadas y se llegó a determinar que la más óptima para el sistema realizado es entre 20-30 cm, porque es donde mejor se detecta el rostro, permitiendo tener una captura con mejores detalles para el procesamiento de la imagen (Ver en Capítulo 3, sección 3.1.2.).
- Se realizó pruebas en diferentes escenarios para obtener los resultados de funcionamiento en la cual se determinó que los tres parámetros antes mencionados deben ser fijos y deben cumplir con los valores determinados por las pruebas, esto se lo puede revisar en el capítulo 3, sección 3.3.3.
- Se presentó problemas de luminosidad debido a que el sitio físico de trabajo que no tuvo las condiciones necesarias para ejecutar el algoritmo por lo se recurrió a luz artificial para obtener un nivel de iluminación alta, la misma que fue determinado en las pruebas de funcionamiento. (Ver en Capítulo 3, sección 3.1.3.).
- Se evidencio que el número de imágenes es importante para que el algoritmo tenga un mejor rendimiento por lo que para este sistema se tomó 40 imágenes por usuarios
- Se realizó el diseño de cada uno de los procesos ejemplificando el flujo con cada uno de los archivos que intervienen en: el entrenamiento del motor, reconocimiento facial, adicionar y eliminar usuario.
- El presente trabajo tiene gran importancia ya que propicia su aplicación en cualquier área que tenga acceso restringido como medida de seguridad.
- Poner en práctica la utilización del sistema para grandes volúmenes de usuarios.

RECOMENDACIONES

- Verificar que todos los paquetes instalados para la ejecución del sistema sean versiones compatibles, en este caso la librería OpenCV y Python, una manera muy práctica es realizando programas sencillos como carga de imágenes, videos, etc.
- Realizar pruebas de funcionamiento en diferentes ambientes o escenarios para poder encontrar los valores y parámetros adecuados como es la iluminación, la distancia de la cámara al usuario y así poder determinar el mejor lugar físico de trabajo para la implementación del sistema.
- Crear una base de datos con varias imágenes por usuario, en la investigación se realizó con 40 tomas tratando de captar todos los rasgos posibles para que el sistema tenga un procesamiento adecuado que garantice su funcionamiento.
- Se sugiere que los usuarios que van hacer registrados en el sistema deben únicamente utilizar la función de adicionar usuarios que fue creado en el mismo sistema donde se debe registrar el nombre y número de ID y automáticamente la cámara del sistema se activa capturando las imágenes del rostro por lo que no es necesario utilizar otros dispositivos externos de capturas de imágenes ya que esto provoca un error en el sistema.

BIBLIOGRAFÍA.

- Barranco Gutiérrez, A., Martínez Díaz, S., & Gómez Torres, J. (01 de 2016). *Visión estereoscópica por computadora con Matlab y OpenCV*. Obtenido de https://www.researchgate.net/profile/Alejandro_Barranco_Gutierrez5/publication/309717602_VISION_ESTEREOSCOPICA_POR_COMPUTADORA_CON_MATLAB_Y_OPENCV/links/581e1b7e08aea429b295bf32.pdf
- Cerón Correa , A., Salazar, A., & Prieto Ortiz, F. (2013). RECONOCIMIENTO DE ROSTROS Y GESTOS FACIALES MEDIANTE UN ANÁLISIS DE RELEVANCIA CON IMÁGENES 3D. *RIDI*, 9. Obtenido de http://revistas.uptc.edu.co/revistas/index.php/investigacion_uitama/article/view/2563/2420
- Espinosa Duró, V. (08 de 2014). *Evaluación de Sistemas de Reconocimiento Biométrico*. Obtenido de <http://www.jcee.upc.es/JCEE2001/PDFs%202000/13ESPINOSA.pdf>
- Ferruz Melero, J. (4 de 02 de 2011). Obtenido de <https://www.xatakaciencia.com/tecnologia/sensores-infrarrojos>
- Gracia, L. (9 de septiembre de 2013). *Qué es OpenCV*. Obtenido de unpocodejava: <https://unpocodejava.wordpress.com/2013/10/09/que-es-opencv/>
- Mendes, C., Fco, J., & Rodriguez Cotorrelo, E. (12 de Marzo de 2013). *Metodología XP*. Recuperado el 06 de 03 de 2017, de <http://www.slideshare.net/Piskamen/metodologia-xp>
- Vegas, J. (04 de 1998). *ORACLE: Arquitectura*. Obtenido de <https://www.infor.uva.es/~jvegas/cursos/bd/orarq/orarq.html>
- Alvarez, R., & Alvarez, M. (26 de 10 de 2016). *desarrollo web*. Obtenido de <https://desarrolloweb.com/articulos/303.php>
- Arbelaez, R. (24 de agosto de 2016). *Las 7 claves fundamentales para la seguridad tecnológica de una empresa dichas por experto de Microsoft*. Obtenido de Microsoft.com: <https://news.microsoft.com/es-xl/las-7-claves-fundamentales-para-la-seguridad-tecnologica-de-una-empresa-dichas-por-experto-de-microsoft/#sm.00000paf110I98fbgvhiumetx30p8>

- Armengot Iborra, M. (09 de 2006). *Análisis comparativo de métodos basados en subespacios aplicados al reconocimiento de caras*. Obtenido de <http://www.uv.es/marjoari/pdf/definitivo.pdf>
- Batra, R. (2015). *A Primer on SQL(3rd Edition)*. En R. Batra. Leanpub.
- Bronte Palacios, S. (2008). *Sistema de Detección y Reconocimiento Facial de Conductores Mediante Sistemas de Visión Computacional*. Obtenido de http://www.robosafe.com/personal/sebastian.bronte/pub/S_Bronte_TFC08.pdf
- Camshift*. (2016). Obtenido de Vision artificielle: <http://dictionnaire.sensagent.leparisien.fr/Camshift/fr-fr/>
- CASTRILLON , W., ALVAREZ , D., & LÓPEZ , A. (2008). TÉCNICAS DE EXTRACCIÓN DE CARACTERÍSTICAS EN IMÁGENES PARA EL RECONOCIMIENTO DE EXPRESIONES FACIALES. *UTP*, 9.
- (2014). *Computer Vision*. En R. Klette, *Computer Vision*. Durham, UK: Springer London Heidelberg New York Dordrecht.
- dezyre. (2015). *dezyre*. Obtenido de Principal Component Analysis Tutorial: <https://www.dezyre.com/data-science-in-python-tutorial/principal-component-analysis-tutorial>
- Eslava Muñoz, V. J. (2012). HTML, presente y futuro de la web. En V. J. Eslava Muñoz, *HTML, presente y futuro de la web* (pág. 188).
- fergarcia. (23 de 01 de 2013). *Entorno de Desarrollo Integrado (IDE)*. Obtenido de <https://fergarcia.wordpress.com/2013/01/25/entorno-de-desarrollo-integrado-ide/>
- Florian, L., & Carranza, F. (2006). *Reconocimiento del Iris*. Obtenido de <http://www.advancedsourcecode.com/reconocimientodeliris.pdf>
- Furfaro, A. (2010). *Manejo de Bibliotecas Opencv*. Obtenido de <http://www2.electron.frba.utn.edu.ar/~afurfaro/Info1/Opencv/opencv.pdf>
- Garage, W. (2010). *OpenCV*. Obtenido de ICRA 2010 Tutorial: [http://mirror.sysu.edu.cn/wiki.ros.org/attachments/Events\(2f\)ICRA2010Tutorial/ICRA_2010_OpenCV_Tutorial.pdf](http://mirror.sysu.edu.cn/wiki.ros.org/attachments/Events(2f)ICRA2010Tutorial/ICRA_2010_OpenCV_Tutorial.pdf)
- Gimeno Hernández, R. (2010). *ESTUDIO DE TÉCNICAS DE RECONOCIMIENTO FACIAL*. Barcelona.

- González Urbano, M. (2013). *Reconocimiento de iris*. Obtenido de <http://www.maia.ub.es/~sergio/linked/marcosgonzalezmemoria.pdf>
- Hilyard, J., & Stephen, T. (2015). *C# 6.0 Cookbook*. Microsoft Press.
- Identificación de iris para soluciones stand-alone o web*. (2015). Obtenido de <http://www.goit.cl/verieye.html>
- Lopez Acevedo, L. (2013). *Tutorial de PostgreSQL*. Obtenido de Tutorial de PostgreSQL. Obtenid
- López, A. F., Alvarez, D. A., & Castrillon, W. A. (2008). Técnicas de extracción de características en imágenes para el reconocimiento de expresiones faciales. *UTP*, 9.
- maneja* *datos*. (02 de 2014). Obtenido de <http://www.manejandodatos.es/2014/02/probando-anaconda-una-version-muy-completa-de-python/>
- Meléndez Valladarez, S. M., Gaitan, M. E., & Pérez Reyes, N. N. (28 de 01 de 2016). *METODOLOGIA ÁGIL DE DESARROLLO DE SOFTWARE PROGRAMACION*. Recuperado el 06 de 03 de 2017, de <http://repositorio.unan.edu.ni/1365/1/62161.pdf>
- Mery, D. (17 de 08 de 2004). *Visión por Computador*. Obtenido de <http://dmery.sitios.ing.uc.cl/Prints/Books/2004-ApuntesVision.pdf>
- Microsoft. (24 de agosto de 2016). *Microsoft*. Recuperado el 08 de 03 de 2017, de Las 7 claves fundamentales para la seguridad tecnológica de una empresa dichas por experto de Microsoft: <https://news.microsoft.com/es-xl/las-7-claves-fundamentales-para-la-seguridad-tecnologica-de-una-empresa-dichas-por-experto-de-microsoft/#sm.0000qkt2by158febirs8qlerosswg#o6x44if2gaZEmlkC.97>
- Ming-Shi Wang ; Wei-Che Chen. (30 de 04 de 2008). *SPIE*. Obtenido de <http://electronicimaging.spiedigitallibrary.org/article.aspx?articleid=1099994>
- Moreno Díaz , A. (2004). *Reconocimiento Facial Automático mediante Técnicas de Visión Tridimensional*. Obtenido de <http://oa.upm.es/625/1/10200408.pdf>
- Peralta, G. (07 de 06 de 2012). *Maestros de la Web*. Obtenido de <http://www.maestrosdelweb.com/ventajas-python/>
- Pietikäinen, M. (12 de 12 de 2010). *Local Binary Patterns*. Obtenido de http://www.scholarpedia.org/article/Local_Binary_Patterns

- Proyectos ágiles*. (2017). Recuperado el 04 de 03 de 2017, de <https://proyectosagiles.org/que-es-scrum/>
- ROLLET, O. (7 de 10 de 2014). *Ingeniería Systems*. Obtenido de PHP y MySQL - Manual de aprendizaje: <http://www.ingenieriasystems.com/2014/07/PHP-y-MySQL-Manual-de-aprendizaje-para-crear-un-sitio-web-Autor--Olivier-ROLLET.html>
- Vega Aquino, E. A. (2011). *Detección y seguimiento de rostro*. Madrid.
- Villegas Quezada, C. (2005). *Reconocimiento de rostros utilizando análisis de componentes principales: limitaciones del algoritmo*. Obtenido de <http://www.bib.uia.mx/tesis/pdf/014620/014620.pdf>
- Wagner, P. (8 de 11 de 2011). *Local Binary Patterns*. Obtenido de http://bytefish.de/blog/local_binary_patterns/
- Zamorano Flores, J. L., & Serrano Moya, G. F. (8 de 11 de 2002). *Sistemas infrarrojos de comunicaciones inalámbricas*. Obtenido de <http://www.izt.uam.mx/newpage/contactos/anterior/n47ne/infra.pdf>

ANEXOS

Manual para la realización de las pruebas para usuarios.

Para la realización de la aplicación en Java, se hicieron dos paquetes, unos donde se encuentra el código fuente con los objetos sobre los cuales se van a trabajar, y el otro donde se encuentran los componentes gráficos uno para el administrador del sistema y el otro para el usuario final.

A continuación, se muestra la interfaz principal que posee la aplicación para el administrador



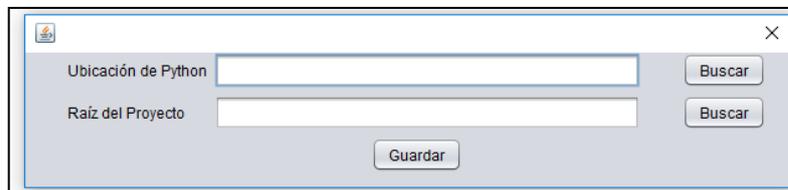
En la interfaz siguiente se muestra el menú de la aplicación, la cual contiene las opciones de:

- **Salir**
- **Adicionar usuario**
- **Eliminar usuario**
- **Reconocimiento**
- **Ver historial**
- **Configurar programa**



En el programa se tiene una clase Configuración que va a mapear contra el archivo `pythonProperties.properties`, que contiene las propiedades de Python, en las que se ejemplifica los `get` y los `set` para acceder a cada una de las propiedades.

Al ejecutar **Configuración del programa** se muestra la siguiente pantalla donde el usuario debe escoger donde se encuentra la ubicación de Python y la raíz del proyecto:



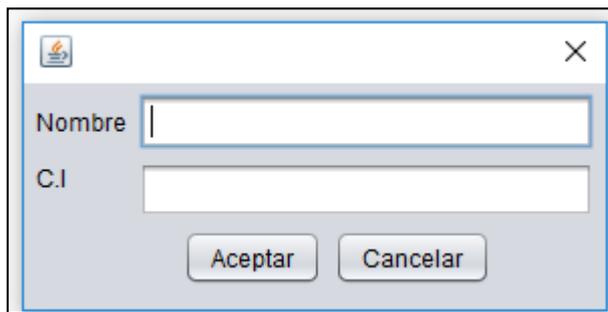
Al ejecutar **Ver Historial** se muestra la siguiente pantalla con todos los usuarios:

ID	Nombre	Autenticado	Confianza	Hora
1717186918	DarioSolis	<input checked="" type="checkbox"/>	21.0322193...	2017-07-18 ...
1717186918	DarioSolis	<input checked="" type="checkbox"/>	24.0631644...	2017-07-18 ...
-1	Unknow	<input type="checkbox"/>	29.7070481...	2017-07-18 ...
1712760618	Marcelo	<input checked="" type="checkbox"/>	24.5586227...	2017-07-18 ...
1711005031	Raul	<input checked="" type="checkbox"/>	19.2745059...	2017-07-18 ...
1713160750	Pablo	<input checked="" type="checkbox"/>	19.9985411...	2017-07-18 ...
-1	Unknow	<input type="checkbox"/>	26.6559941...	2017-07-18 ...
1717186918	DarioSolis	<input checked="" type="checkbox"/>	20.2806251...	2017-07-18 ...
-1	Unknow	<input type="checkbox"/>	25.6132199...	2017-07-18 ...
-1	Unknow	<input type="checkbox"/>	28.1314280...	2017-07-18 ...
-1	Unknow	<input type="checkbox"/>	25.0228020...	2017-07-18 ...
1717186918	Dario	<input checked="" type="checkbox"/>	24.5906837...	2017-07-18 ...
1717186918	Dario	<input checked="" type="checkbox"/>	23.7314549...	2017-07-18 ...
-1	Unknow	<input type="checkbox"/>	32.3258741...	2017-07-18 ...
-1	Unknow	<input type="checkbox"/>	29.1198706...	2017-07-18 ...
-1	Unknow	<input type="checkbox"/>	29.9558884...	2017-07-18 ...

Otros de los archivos que se desarrollo fue person, el cual vendría representando una persona, el cual va a tener en conjunto de anotaciones de Jsom.

Para adicionar se crea un objeto del tipo AddUser que contiene los atributos que se le van adicionar al usuario.

Al ejecutar **Adicionar Usuario** se muestra la siguiente pantalla para que se ingresen los datos correspondientes, para luego comenzar la toma de las fotos:



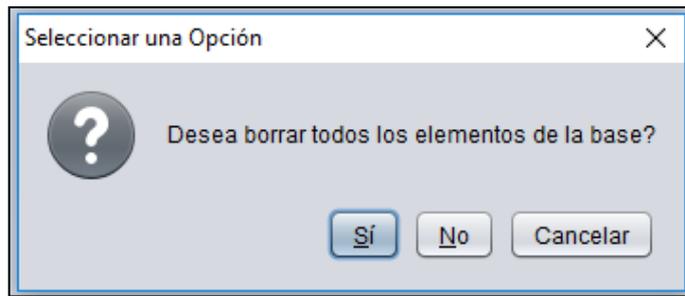
A screenshot of a Windows-style dialog box titled "Adicionar Usuario". It contains two text input fields: "Nombre" and "C.I.". Below the fields are two buttons: "Aceptar" and "Cancelar". The dialog box has a standard Windows window border with a close button (X) in the top right corner.

Al presionar el botón aceptar, se van a validar cada uno de los atributos ingresados, si los mismos están correctos y cumplen con la validación especificada entonces se guarda en la base de datos y se ejecuta el trainer, sino no se muestra un mensaje de error. Si se presiona Cancelar no se ejecuta la funcionalidad.

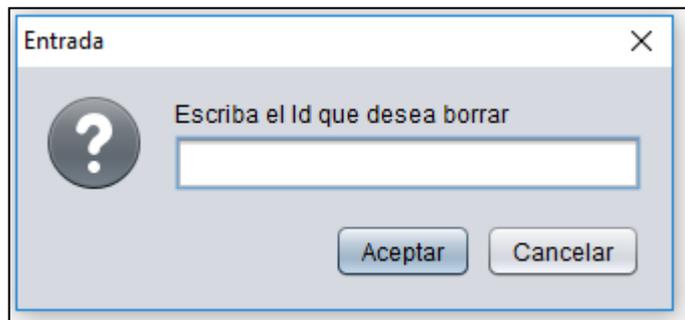
Luego la aplicación comienza la toma de las fotos:



Al ejecutar **Eliminar usuario** se muestra la siguiente pantalla con dos opciones, una para que se eliminen todos los elementos que se encuentran en la base de datos si se presiona **Si**:



Y la otra opción que es **No** para eliminar usuarios específicos la cual solicita el ID a ser eliminado y se muestra la interfaz para que ingresen el identificador de lo que se desea borrar:



En cuanto el interfaz para los usuarios a ser reconocidos o no dentro del sistema se ha creado la misma interfaz del administrador, pero con la diferencia que únicamente se encuentra habilitado la opción **Reconocer**, lo cual garantiza que el sistema no sea manipulado o modificado por personas no autorizadas como se muestra en la imagen siguiente:



Para el reconocimiento se trabajará en JavaFrontEnd, y se hará uso de una biblioteca para trabajar las salidas desde un objeto Json a un objeto java, a través de las anotaciones para

determinar contra qué valor se debe mapear dentro del objeto Json. Se utilizará el PythonprocessBuilder el cual se encarga de generar un proceso aparte de la máquina virtual de Java, por lo que va a propiciar que se ejecute en cualquier plataforma. Fuera de la máquina virtual se tiene que mandar a ejecutar un proceso, y esta a su vez no ejecuta ningún proceso hasta que el proceso que se mandó a ejecutar no termine, para que una vez que haya terminado mandar hacer en entrenamiento, ejemplo se ejecuta adicionar y luego se realiza el entrenamiento, y así con cada una de las funcionalidades.

Para el reconocimiento se trabaja con cada una de las propiedades que se encuentran en la raíz, donde se va a presentar los datos recogidos en el result como el nombre, el ID para cada imagen que se toma del sistema.

Al ejecutar **Reconocer**, si el usuario está en base de datos se confirma a través de la siguiente imagen:



Si el usuario no está en base de datos se muestra una advertencia ejemplificando que el usuario no está autorizado:

