

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Título:

Sistema de identificación de personas basado en rasgos
faciales.

Autores:

✚ Amelia Aguilera Reyes.

✚ Rafael Alberto Quiles Velázquez.

Tutor:

Ing. Yainier Labrada Nueva.

Co-Tutor:

Ing. Adrián Rivera Correa.

Frase

"No pretendamos que las cosas cambien, si siempre hacemos lo mismo.

La crisis, es la mejor bendición que puede sucederle a personas y países, porque la crisis trae progresos.

La creatividad nace de la angustia como el día nace de la noche oscura.

Es en la crisis que nace la inventiva, los descubrimientos y las grandes estrategias.

Quien supera la crisis se supera a sí mismo sin quedar superado.

Quien atribuye a la crisis sus fracasos y penurias, violenta su propio talento y respeta más a los problemas que a las soluciones. La verdadera crisis, es la crisis de la incompetencia.

El inconveniente de las personas y los países es la pereza para encontrar las salidas y soluciones. Sin crisis no hay desafíos, sin desafíos la vida es una rutina, una lenta agonía. Sin crisis no hay méritos.

Es en la crisis donde aflora lo mejor de cada uno, porque sin crisis todo viento es caricia. Hablar de crisis es promoverla, y callar en la crisis es exaltar el conformismo. En vez de esto, trabajemos duro. Acabemos de una vez con la única crisis amenazadora, que es la tragedia de no querer luchar por superarla."

Albert Einstein.

1879



Declaración de autoría

Declaramos que somos los únicos autores de este trabajo de diploma y conferimos a la Universidad de las Ciencias Informáticas (UCI) los derechos patrimoniales del mismo, con carácter exclusivo.

Para que así conste firmamos los presentes a los ____ días del mes de ____ del año ____.

Amelia Aguilera Reyes

Autor

Rafael Alberto Quiles Velázquez

Autor

Ing. Yainier Labrada

Tutor

Ing. Adrián Rivera Correa

Co-tutor



Datos del contacto

Tutor: Ing. Yainier Labrada Nueva.

Graduado de Ingeniería en Ciencias Informáticas en la Universidad de Ciencias Informáticas en el año 2008. Profesor del Departamento de Biometría del Centro de Identificación y Seguridad Digital (CISED), adjunto al Departamento de Ciencias Básicas en la Facultad No. 1, de la Universidad de las Ciencias Informáticas (UCI), con 5 años de experiencia.

Correo electrónico: ylabrada@uci.cu.

Co-tutor: Ing. Adrián Rivera Correa.

Ingeniero en Ciencias Informáticas. Profesor asociado al Centro de Identificación y Seguridad Digital (CISED), pertenece al Departamento de Biometría, graduado en el curso 2011-2012. Tiene 1 año de experiencia en el tema procesamiento de imágenes.

Correo electrónico: acorrea@uci.cu.



Resumen

Los humanos a menudo utilizan los rostros para identificar a las personas, con los avances alcanzados en las capacidades de la computación actualmente se pueden realizar reconocimientos automáticos de forma similar, esta nueva tecnología se basa en el reconocimiento de personas utilizando los rasgos conductuales o físicos intrínsecos de los individuos, a la cual se le conoce con el nombre de biometría. Durante la realización de este trabajo se desarrolla un sistema de identificación de personas basado en rasgos faciales que sea capaz de detectar e identificar el rostro de una persona en una imagen digital. Para ello se lleva a cabo una valoración crítica de los métodos y algoritmos existentes que permiten realizar el reconocimiento de rostros en imágenes digitales, se seleccionan las herramientas necesarias para el desarrollo del sistema y además se diseña, implementa y se valida para comprobar su correcto funcionamiento por medio de pruebas de funcionalidad y de efectividad.

Con el resultado de la presente investigación se pretende automatizar el proceso de identificación en el Centro de Identificación y Seguridad Digital (CISED) con el fin de garantizar la confiabilidad al acceso a las instalaciones del centro, conjuntamente se pueden integrar otras soluciones que se desarrollen en el propio centro.

Palabras claves: biometría, rasgos faciales, reconocimiento de rostro.



Índice de contenido

DECLARACIÓN DE AUTORÍA	II
DATOS DEL CONTACTO	IV
RESUMEN	V
ÍNDICE DE CONTENIDO	VI
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	X
INTRODUCCIÓN	1
CAPÍTULO I: “FUNDAMENTACIÓN TEÓRICA”	5
1.1 Introducción	5
1.2 Antecedentes históricos	5
1.3 Conceptos fundamentales	6
1.4 Sistemas biométricos	8
1.5 Sistemas biométricos. Reconocimiento basado en rasgos faciales	10
1.6 Soluciones existentes en el mundo	13
1.7 Propuesta de los algoritmos a utilizar en el sistema de reconocimiento facial	15
1.8 Lenguajes, Metodologías y Herramientas de desarrollo	26
1.9 Conclusiones parciales	36
CAPÍTULO II: “ANÁLISIS Y DISEÑO”	37
2.1 Introducción	37
2.2 Propuesta del sistema	37
2.3 Modelo de dominio	38
2.4 Fase de planificación	39



2.5	Fase de diseño	44
2.6	Conclusiones parciales	52
CAPÍTULO III: “IMPLEMENTACIÓN Y PRUEBAS”		52
3.1	Introducción	52
3.2	Estándar de codificación	53
3.3	Tareas de ingeniería	53
3.4	Diagrama de componentes	54
3.5	Diagrama de despliegue	56
3.6	Pruebas	56
3.7	Conclusiones parciales	68
CONCLUSIONES GENERALES		69
RECOMENDACIONES		69
BIBLIOGRAFÍA REFERENCIADA		70
BIBLIOGRAFÍA CONSULTADA		72
GLOSARIO DE TÉRMINOS		73
ACRÓNIMOS		74
ANEXOS		76
4.1	Descripción de las historias de usuarios del sistema	76
4.2	Descripción de las tarjetas CRC.	79
4.3	Tareas de Ingeniería del sistema	82
4.4	Casos de pruebas de funcionalidad	84
4.5	Pruebas de tiempo al sistema de identificación.....	88
4.6	Interfaces de Usuario.....	90



4.7 Pruebas al sistema de reconocimiento facial para determinar el umbral de aceptación. 96

Índice de figuras

FIGURA 1.1: PROCESO DE IDENTIFICACIÓN GENÉRICO DE LOS SISTEMAS BIOMÉTRICOS.....7

FIGURA 1.2: ORGANIZACIÓN DE UN SISTEMA DE VERIFICACIÓN, COMPUESTO DE DOS FASES, DETECCIÓN DE ROSTROS Y VERIFICACIÓN.....8

FIGURA 1.3: TÉCNICAS BIOMÉTRICAS ACTUALES: (A) ROSTRO, (B) TERMOGRAMA FACIAL, (C) HUELLA DACTILAR, (D) GEOMETRÍA DE LA MANO, (E) VENAS DE LA MANO, (F) IRIS, (G) PATRONES DE LA RETINA, (H) FIRMA, (I) VOZ.8

FIGURA 1.4: DIVISIÓN DE LAS CARACTERÍSTICAS BIOMÉTRICAS..... 9

FIGURA 1.5: COMPARACIÓN DE CARACTERÍSTICAS DE SISTEMAS BIOMÉTRICOS. 10

FIGURA 1.6: CLASIFICACIÓN DE ALGUNOS DE LOS MÉTODOS DE RECONOCIMIENTO DEL ROSTRO.....13

FIGURA 1.7: DETECCIÓN DEL ROSTRO DE LA PERSONA.. 17

FIGURA 1.8: IMAGEN RECORTADA O SEGMENTADA SEGMENTADA..... 17

FIGURA 1.9: ROSTRO INCLINADO..... 18

FIGURA 1.10: DETERMINACIÓN DEL ÁNGULO Σ 19

FIGURA 1.11: IMAGEN ROTADA..... 19

FIGURA 1.12: FORMATO GEOMÉTRICO (IZQUIERDA), CON $W=240$ PÍXELES (DERECHA)..... 20

FIGURA 1.13: FÓRMULA PARA LA ECUALIZACIÓN DEL HISTOGRAMA. 21

FIGURA 1.14: REPRESENTACIÓN DEL HISTOGRAMA DE LA IMAGEN EN SUS TRES COMPONENTES RGB.....21

FIGURA 1.15: REPRESENTACIÓN DEL HISTOGRAMA DE LA IMAGEN EN ESCALA DE GRISES.....21

FIGURA 1.16: REPRESENTACIÓN DEL HISTOGRAMA ECUALIZADO DE LA IMAGEN EN ESCALA DE GRISES.....21

FIGURA 1.17: EXTRACCIÓN DE CARACTERÍSTICAS DE UNA IMAGEN TRAS TENER PREVIAMENTE DETECTADA EL ÁREA DEL ROSTRO..... 22

FIGURA 1.18: PLANTILLA DE ASM. 23

FIGURA 1.19: UN ROSTRO UBICADO POR LA PLANTILLA DE ASM..... 23



FIGURA 1.20: PARA CADA ROSTRO ENCONTRADO SE CREA UNA GRAFO CON LOS PUNTOS DE ASM Y PARA CADA UNO DE ESTOS PUNTOS SE OBTIENE UNA VECINDAD DE 7X7 PÍXELES DE LA CUAL SE EXTRAEN LAS CARACTERÍSTICAS CORRESPONDIENTES.	23
FIGURA 1.21: FÓRMULA PARA CALCULAR LA DCT EN UNA MATRIZ 2D	24
FIGURA 1.22: VECINDAD EXTRAÍDA PARA CADA UNO DE LOS PUNTOS ANTES LOCALIZADOS CON UN TAMAÑO DE 7X7 PÍXELES A LA CUAL SE LE APLICA EL RECORRIDO EN ZIG-ZAG DANDO COMO SALIDA EL VECTOR CARACTERÍSTICO CORRESPONDIENTE.	24
FIGURA 1.23: PROCESO DE COMPARACIÓN CON LA BASE DE DATOS.....	24
FIGURA 1.24: ECUACIÓN PARA EL CÁLCULO DE LA DISTANCIA ENTRE DOS VECTORES.....	25
FIGURA 1.25: TRES USUARIOS, A MODO DE EJEMPLO, DE LA BASE DE DATOS BIOID.....	26
FIGURA 1.26: FASES DE LA METODOLOGÍA XP.....	29
FIGURA 2.1: ETAPAS DEL SISTEMA DE IDENTIFICACIÓN DE PERSONAS BASADO EN RASGOS FACIALES.....	38
FIGURA 2.2: DIAGRAMA DEL MODELO DE DOMINIO.	38
FIGURA 2.3: DIAGRAMA DE PAQUETES DEL SISTEMA.	45
FIGURA 2.4: DIAGRAMA DE CLASES PERTENECIENTE AL SUB-PAQUETE DE GESTIÓN DE USUARIOS.....	45
FIGURA 2.5: DIAGRAMA DE CLASES PERTENECIENTE AL SUB-PAQUETE DE ALGORITMO DE RECONOCIMIENTO FACIAL.....	46
FIGURA 2.6: DIAGRAMA ENTIDAD-RELACIÓN.	47
FIGURA 2.7: ARQUITECTURA DEL SISTEMA DE IDENTIFICACIÓN DE PERSONAS.....	48
FIGURA 2.8: ESTILO ARQUITECTÓNICO TUBERÍAS Y FILTROS DEL SISTEMA.....	50
FIGURA 3.1: DIAGRAMA DE COMPONENTES DEL SISTEMA.	55
FIGURA 3.2: DIAGRAMA DE DESPLIEGUE DEL SISTEMA.....	56
FIGURA 3.3: DETERMINACIÓN DEL UMBRAL EQUAL ERROR RATE.....	57
FIGURA 3.4: DETERMINACIÓN DEL UMBRAL DE ACEPTACIÓN PARA EL SISTEMA DE IDENTIFICACIÓN DE PERSONAS.	58
FIGURA 3.5: REPRESENTACIÓN DE TIEMPO DE COMPARACIÓN.....	59



FIGURA 3.6: REPRESENTACIÓN DE LOS TIEMPOS DE IDENTIFICACIÓN DE UNA PERSONA. 61

FIGURA 3.7: REPRESENTACIÓN DE LOS TIEMPOS DE EXTRACCIÓN DE LOS VECTORES CARACTERÍSTICOS. 62

FIGURA 3.8: GRAFO DE FLUJO: FUNCIONALIDAD OBTENER VECTOR CARACTERÍSTICO. 66

FIGURA 3.9: GRAFO DE FLUJO: FUNCIONALIDAD COMPARAR VECTOR CARACTERÍSTICO..... 67

Índice de tablas

TABLA 1.1: COMPARACIÓN DE LOS MÉTODOS DE DETECCIÓN DE ROSTROS. 12

TABLA 1.2: CARACTERÍSTICAS GEOMÉTRICAS DE LAS IMÁGENES. 20

TABLA 2.1: ACTORES DEL SISTEMA. 39

TABLA 2.2: DESCRIPCIÓN DE LA HU. PROCESAR IMAGEN. 42

TABLA 2.3: ESTIMACIÓN DE ESFUERZO POR HISTORIAS DE USUARIO..... 43

TABLA 2.4: PLAN DE DURACIÓN DE LAS ITERACIONES. 43

TABLA 2.5: PLAN DE ENTREGA. 44

TABLA 2.6: TARJETA CRC CORRESPONDIENTE A LA CLASE RECONOCIMIENTO FACIAL ASM. 44

TABLA 3.1: CONVENCIONES DE NOMBRE. 53

TABLA 3.2: TAREAS DE INGENIERÍA EN LA PRIMERA ITERACIÓN..... 54

TABLA 3.3: DESCRIPCIÓN DE LOS COMPONENTES DEL SISTEMA DE IDENTIFICACIÓN. 56

TABLA 3.4: TIEMPO DE COMPARACIONES DE IMÁGENES FACIALES SIN EL TIEMPO DE EXTRACCIÓN DE CARACTERÍSTICAS. 59

TABLA 3.5: TIEMPOS DE EXTRACCIÓN DE CARACTERÍSTICAS CONJUNTAMENTE CON LA COMPARACIÓN CON LA BASE DE DATOS. 61

TABLA 3.6: TIEMPO DE EXTRACCIÓN DE VECTORES. 62

TABLA 3.7: CASO DE PRUEBA DE LA FUNCIONALIDAD: PROCESAR IMAGEN. 63



Introducción

El reconocimiento de rostros y la visión con el empleo de una computadora es un tema que ha tomado mucho auge en los últimos años, debido a la importancia que presenta. En este trabajo investigativo el reconocimiento de rostros es el tema central, en el cual se describen las fases a desempeñar para reconocer rostros en una imagen digital bidimensional.

La biometría proviene de las palabras bio (vida) y metría (medida), esta se encarga del reconocimiento automático de personas mediante sus rasgos físicos (cara, retina, iris, huellas dactilares, entre otras modalidades) o rasgos de conducta (forma de andar, de escribir, la voz, entre otros), para verificar identidades o para identificar individuos (1). Una de las características de la biometría son los patrones faciales, estos se obtienen a partir de los algoritmos de reconocimiento facial que realizan los procesos de segmentación, extracción de características y descripción, en el cual cada objeto queda representado por una colección de descriptores (2).

En el presente, muchas organizaciones acuden al uso de soluciones de seguridad biométrica, tales como algunos tipos de lectores de huella digital, dispositivos de análisis del iris, sistemas de reconocimiento facial, entre otros. La tecnología de reconocimiento facial se basa en el concepto de biometría, por lo que se habla de una tecnología basada en computadoras que determina la ubicación y el tamaño del rostro de las personas en imágenes digitales (2). En un transcurso de diez a quince años, se han logrado importantes avances e iniciativas en la tecnología de reconocimiento facial, siendo así el centro de atención en el tema de identificación de personas.

El reconocimiento facial automatizado es un concepto novedoso, el mismo comienza a partir de la década del 60, más específico durante los años 1964 y 1965 cuando Woodrow Wilson Bledsoe, Helen Chan Wolf y Charles Bisson desarrollan el primer sistema semiautomático, que para efectuar sus operaciones requería de un administrador que localizara los rasgos, como ojos, orejas, nariz y boca en las fotografías, antes de que este calculara distancias entre puntos en común, los cuales eran comparados luego con datos de referencia (3).

En la actualidad la tecnología de reconocimiento facial se utiliza para combatir las falsificaciones de pasaportes, en la identificación de niños extraviados y minimizar el fraude en las identificaciones (4). En Cuba existe el Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV), donde se hacen estudios de los patrones biométricos para el reconocimiento de personas según sus características



faciales (5). Este es un centro de investigaciones teóricas y aplicadas que tiene como misión fundamental asimilar, desarrollar e introducir en la práctica social los aspectos más novedosos de la teoría y la práctica del reconocimiento de patrones (RP) en su concepción más general, y de la minería de datos (MD), que permitan responder a las necesidades del progreso científico-técnico y socio-económico, así como incrementar el patrimonio científico nacional (6).

En la Universidad de las Ciencias Informáticas (UCI) se encuentra el Centro de Identificación y Seguridad Digital (CISED) y en el mismo el departamento de Biometría, que estudia los patrones para el reconocimiento de personas, como huellas dactilares, firma digital y reconocimiento facial. Actualmente el centro dispone de un control de acceso para la autenticación de personas a través de su credencial, el cual no garantiza la autenticidad de las personas ya que puede ser burlado fácilmente mediante la suplantación de identidad.

Con lo mencionado anteriormente se da a conocer como **problema científico**: el sistema de control de acceso llevado a cabo actualmente en el CISED no garantiza la confiabilidad en la identificación de personas para el acceso a las instalaciones del centro.

Con el objetivo de lograr una alternativa de solución, el presente trabajo propone dar respuesta al problema existente, razón por la cual el **objeto de estudio** queda enmarcado en los sistemas de identificación de personas basado en rasgos faciales.

El **objetivo** del trabajo es desarrollar un sistema de identificación de personas basado en rasgos faciales para automatizar el proceso de identificación en el CISED.

Los **objetivos específicos** que se derivan del objetivo del trabajo son:

- ✓ Analizar los sistemas de identificación existentes en el mundo basado en rasgos faciales.
- ✓ Identificar las principales características de los sistemas de identificación de personas basado en rasgos faciales.
- ✓ Analizar los diferentes algoritmos para el reconocimiento de personas basado en rasgos faciales.
- ✓ Diseñar el sistema de identificación de persona basado en rasgos faciales.
- ✓ Implementar el sistema de identificación de persona basado en rasgos faciales.
- ✓ Realizar pruebas al sistema de identificación de persona basado en rasgos faciales.

Las **tareas de investigación** para dar respuesta al objetivo del trabajo son:

- ✓ Análisis de los sistemas existentes a nivel mundial y nacional sobre la identificación de personas basado en rasgos faciales. (Amelia Aguilera Reyes)



- ✓ Análisis de las normas y estándares internacionales de los sistemas de identificación de personas basados en rasgos faciales. (Amelia Aguilera Reyes)
- ✓ Identificación de las herramientas y tecnologías a utilizar en el desarrollo del sistema de identificación de personas basado en rasgos faciales. (Amelia Aguilera Reyes y Rafael Quiles Velázquez)
- ✓ Especificación de los requisitos del software. (Amelia Aguilera Reyes y Rafael Quiles Velázquez)
- ✓ Definición de una arquitectura de software para el sistema de identificación de personas basado en rasgos faciales. (Amelia Aguilera Reyes y Rafael Quiles Velázquez)
- ✓ Elaboración del diseño de la interfaz de usuario. (Rafael Quiles Velázquez)
- ✓ Modelación del diseño de clases. (Amelia Aguilera Reyes y Rafael Quiles Velázquez)
- ✓ Implementación de las funcionalidades del sistema de identificación de personas basado en rasgos faciales. (Amelia Aguilera Reyes y Rafael Quiles Velázquez)
- ✓ Ejecución de las pruebas de calidad al sistema de identificación de personas basado en rasgos faciales. (Amelia Aguilera Reyes y Rafael Quiles Velázquez)

Métodos científicos

Para el desarrollo de la investigación se utilizaron diferentes métodos científicos como son: los teóricos que permiten estudiar las características del objeto de investigación que no son observables directamente y los empíricos que describen y explican las características del objeto.

Métodos teóricos:

- ✓ **Análítico-Sintético:** Se consulta la bibliografía referente al tema de identificación de personas basado en rasgos faciales, así como las normas y estándares internacionales para su identificación de forma segura y se hace uso de la información más adecuada al presente trabajo.
- ✓ **Análisis Histórico-Lógico:** Se realiza un estudio de la evolución de los sistemas de identificación de persona basado en rasgos faciales desarrollados hasta el momento y sus principales aportes.

Método empírico

Como parte de los empíricos se tiene la **entrevista** para interactuar con especialistas que poseen amplio conocimiento en el tema de sistema de identificación de personas.

La justificación de la investigación del presente trabajo se basa en el desarrollo de un sistema que permita realizar la identificación de personas basado en rasgos faciales y así utilizarlo como un control de acceso en el propio centro, del cual él sea propietario y pueda así adaptarlo y configurarlo para los diversos



ambientes de proyectos. Además forma parte de los esfuerzos del CISED para crear sistemas propios de identificación, representando un paso importante para encontrar soluciones alternativas que garanticen la seguridad en el control de acceso.

El contenido se encuentra estructurado en tres capítulos los cuales se agrupan de la manera siguiente:

- ✓ **Capítulo 1:** Fundamentación teórica que incluye un estudio del estado del arte del tema tratado; se abordan elementos teóricos de la investigación tales como biometría, proceso de reconocimiento facial, metodología, lenguajes y herramientas de desarrollo que se utilizan para implementar la solución.
- ✓ **Capítulo 2:** Análisis y diseño donde se presentan las fases de Planificación y Diseño definidas por la metodología Programación Extrema (XP), para dar solución al problema científico. Se define la arquitectura para desarrollar la solución. Se identifican las historias de usuarios y los requisitos no funcionales, se realiza el plan de iteraciones y plan de entregas.
- ✓ **Capítulo 3:** En el tercer capítulo se muestran algunos de los artefactos relacionados con la implementación del sistema, como los diagramas de componentes y despliegue. Se valida el sistema implementado mediante pruebas de efectividad y funcionalidad.



Capítulo I: “Fundamentación Teórica”

1.1 Introducción

Este capítulo aborda los elementos teóricos que sustentan el objeto de estudio y el objetivo de la investigación. Se relacionan todos los conceptos que desde el punto de vista teórico permiten un mejor entendimiento de lo que se plantea en la situación problemática y en el marco del problema en sentido general.

También se realiza un estudio sobre las soluciones que existen en el mercado, con el fin de facilitar la comprensión de la importancia de la investigación, su alcance y su aporte científico.

Por otra parte, se realiza un análisis comparativo de las principales metodologías existentes, además de hacer un estudio detallado de las tecnologías y herramientas que se utilizan a lo largo de la presente investigación, conjuntamente con los algoritmos adecuados para la realización del sistema.

1.2 Antecedentes históricos

En realidad, la biometría se entiende en términos muy amplios. Las personas la ejercen muchas veces a lo largo del día sin casi darse cuenta. Por ejemplo, cuando una persona descuelga el teléfono y escucha la voz del interlocutor, el cerebro trata de comprobar si esa voz se parece a cualquiera de las muestras que tiene almacenadas en su memoria y que ha ido recopilando a lo largo de la vida. El cerebro encuentra similitudes suficientes entre alguno de sus recuerdos y lo que escucha en ese momento, entonces se reconoce a la persona que ha llamado. Si no, se asume que es alguien que no conoce. Del mismo modo, los animales reconocen a otros animales, incluidos los seres humanos, por características biométricas tales como el olor, el tacto o el timbre de la voz. Por tanto, aunque se podría pensar en la biometría como una ciencia-ficción futurista, los principios básicos de la misma eran comprendidos y utilizados antiguamente (7).

En la época de los faraones, en el Valle del Nilo (Egipto) se utilizaban los principios básicos de la biometría para verificar a las personas que participaban en diferentes operaciones comerciales y judiciales (3). Ya en las últimas décadas siendo más específico en los años 70, Goldstein, Harmon, Lesk, usaron 21 marcadores subjetivos específicos tales como el color del cabello y el grosor de los labios para automatizar el reconocimiento facial (2). La limitación de estas soluciones previas era que se computaban manualmente.



En 1988, Kirby y Sirobich aplicaron al problema del reconocimiento facial una técnica estándar del álgebra lineal, o sea el análisis de componentes principales. Esto fue considerado algo así como un hito al mostrar que eran requeridos menos de 100 valores para cifrar acertadamente la imagen de un rostro convenientemente alineado y normalizado (3).

En 1991, Matthew Turk y Alex Pentland publican "Eigenfaces for recognition" en "Journal Cognitive Neuroscience", donde se planteaba que el reconocimiento facial en tiempo real era posible; utilizando las técnicas Eigenfaces, el error residual podía ser utilizado para detectar rostros en las imágenes, un descubrimiento que permitió desarrollar sistemas automatizados de reconocimiento facial en tiempo real seguros (8).

De 1993 a 1997 corrió el programa FERET (Face Recognition Technology, tecnología de reconocimiento facial) patrocinado por el departamento de Defensa la Agencia de Investigación de Productos de Avance de Defensa (DARPA) de Estados Unidos (9), su misión principal fue el desarrollo de capacidades de reconocimiento facial automático que pudiera ser empleado por personal de seguridad, inteligencia y justicia en el desarrollo de sus labores.

Un poco más en la actualidad, en mayo de 2004 comienza el gran reto del reconocimiento facial (The Face Recognition Grand Challenge FRGC) que consistió en una serie de problemas que son progresivamente más difíciles, el objetivo principal de FRGC es mejorar la calidad de los sistemas de reconocimiento facial sobre la prueba de reconocimiento facial del vendedor (The Face Recognition Vendor Test FRVT). Estas pruebas consistían en evaluar las capacidades de los sistemas de reconocimiento facial comercialmente disponibles y educar la comunidad de biometría y el público general sobre como presentar y analizar resultados apropiadamente. FRVT fue diseñado para medir progresos técnicos, para evaluar el rendimiento en bases de datos a gran escala de la vida real, y para introducir nuevos experimentos y ayudar a entender mejor el rendimiento del reconocimiento facial (10).

1.3 Conceptos fundamentales

- **Biometría:** Se refiere a todas aquellas técnicas que permiten identificar y autenticar a las personas a través de sus características fisiológicas y de comportamiento (2). Recientemente, el término "biometría" también ha sido utilizado para referirse al campo emergente de tecnología fiel a la identificación de individuos que usan rasgos biológicos, como aquellos basados en la exploración de huellas digitales, iris o reconocimiento facial (4).



- **Sistema biométrico:** Sistema automatizado que realiza labores de biometría, es decir, un sistema que fundamenta sus decisiones de reconocimiento mediante una característica personal que puede ser reconocida o verificada de manera automatizada. A continuación se muestra el funcionamiento de forma general de los sistemas biométricos (**figura 1.1**) (11).

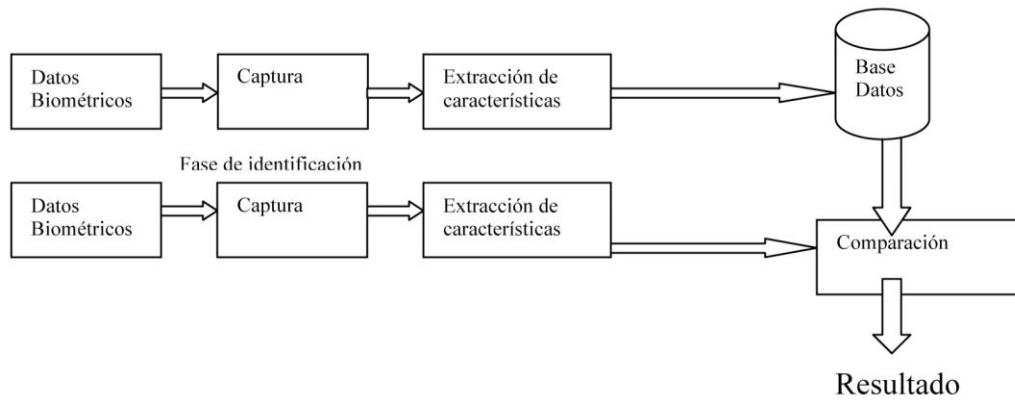


Figura 1.1: Proceso de identificación genérico de los Sistemas Biométricos.

- **Reconocimiento:** Es un término genérico que no implica por defecto una verificación o identificación de un individuo. Todos los sistemas biométricos realizan reconocimiento para "distinguir de nuevo" una persona que se ha ingresado previamente a un sistema (7).
- **Sistema de reconocimiento facial:** Un sistema de reconocimiento facial es una aplicación dirigida por ordenador que identifica automáticamente a una persona en una imagen digital. Esto es posible mediante un análisis de las características faciales de la persona, extraídas de la imagen o de un fotograma clave de una fuente de video, y comparándolas con una base de datos (7). Es utilizado principalmente en Sistemas de Seguridad para el reconocimiento de los usuarios. Consiste en un lector que localiza el rostro definiendo las características de la persona, y al solicitar acceso se verifica que coincidan las características del usuario con la base de datos (12). Este proceso se muestra en la figura 1.2.

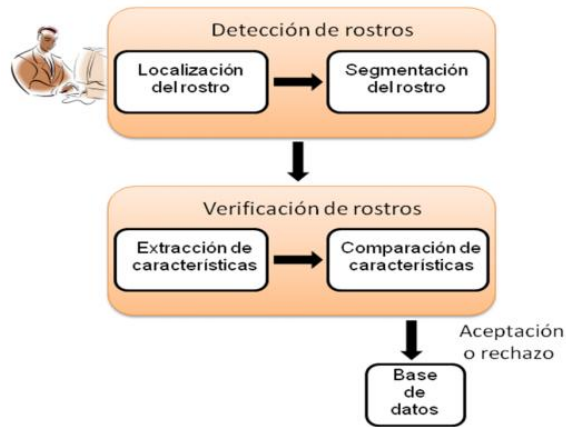


Figura 1.2: Organización de un sistema de verificación, compuesto de dos fases, detección de rostros y verificación (12).

1.4 Sistemas biométricos. Tipos de sistemas biométricos y sus aplicaciones

En la actualidad existen sistemas biométricos que basan su acción en el reconocimiento de diversas características, como puede apreciarse en la siguiente figura:

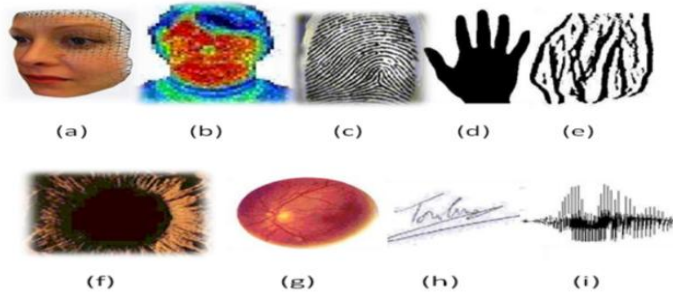


Figura 1.3: Técnicas biométricas actuales: (a) rostro, (b) termograma facial, (c) huella dactilar, (d) geometría de la mano, (e) venas de la mano, (f) iris, (g) patrones de la retina, (h) firma, (i) voz (14).

Este tipo de tecnología evoluciona rápidamente y tiene un fuerte potencial que hace que sea especialmente interesante en determinadas áreas, de entre las cuales la seguridad informática es una de las más recientes. La biometría forma parte del futuro de los sistemas de seguridad, su desarrollo en los últimos años ha experimentado un gran crecimiento respecto a otras tecnologías. Su eficacia potencial la hacen especialmente interesante en determinadas áreas, en las que ya se empiezan a emplear los sistemas biométricos (13). A continuación se muestran algunas de estas áreas que utilizan los sistemas biométricos para potenciar la seguridad:

- Entidades financieras: Uno de los sectores más preocupados históricamente por la seguridad, para evitar fraudes y pérdidas de dinero.



- Comercio electrónico y banca electrónica: Es una de las áreas que más ha crecido en los últimos años, y la que más ha influido en el desarrollo de nuevos sistemas de seguridad.
- Acceso a sistemas: A nivel local los sistemas puede resultar muy beneficiosos para la seguridad de las empresas, sus posibilidades serían enormes si se crease una base de datos con datos biométricos global que permitiese identificar también a los clientes, con el fin de que nadie ajeno a las actividades de la empresa pudiera franquear la entrada.

1.4.1 Clasificación de las características biométricas

En la **figura 1.4** se presenta un esquema de división de las características biométricas donde se muestran los sistemas biométricos que se basan en rasgos físicos (anatómicas) y aquellos basados en patrones de comportamiento del usuario.

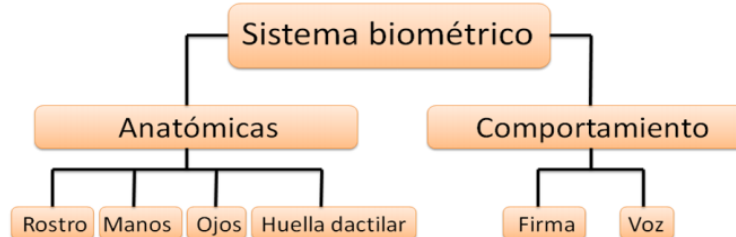


Figura 1.4: División de las características biométricas. (14)

Como se puede observar, las características biométricas de tipo anatómico están basadas en las propiedades estables del cuerpo y las de comportamiento incluyen particularidades aprendidas por el individuo durante su vida. (14)

Sin embargo, la clasificación tiende a crear confusión ya que patrones conductuales como la voz y firma, pueden variar de acuerdo con el estado de ánimo de la persona.

Para evitar este problema es mejor clasificar a las señales biométricas en:

- Señal biométrica estable: Tiene la propiedad de ser relativamente constante en el tiempo excepto por pequeñas perturbaciones debidas al ruido, además, por cambios físicos realmente grandes como cirugías o accidentes (la huella digital, el rostro, los ojos y las manos).
- Señal biométrica alterable: Por ejemplo la voz es el resultado de la vocalización de una frase (variable), a través de las cuerdas vocales (fijas), dando como resultado una plantilla que depende tanto de la frase pronunciada como de las cuerdas vocales.

La huella digital, el rostro, los ojos y las manos no siempre se considerarán señales biométricas estables, como tampoco la voz o la escritura serán variables. Por ejemplo puede existir una señal biométrica



extraída de una imagen del rostro que mida la forma y la expresión cuando se pronuncia una palabra o se demuestra una emoción, o puede existir una señal biométrica estable derivada de la pronunciación de una palabra fija, en un tono fijo, en la cual la señal dependería únicamente de las cuerdas vocales del individuo.

1.4.2 Modalidades biométricas

Según estudios realizados por International Biometric Group (12) el porcentaje del Mercado Biométrico por tecnología en el año 2006, demuestra que las tecnologías de mayor uso hasta ese momento y con mayor apoyo por las industrias comerciales son: los sistemas biométricos basados en la huella digital, y el reconocimiento facial. Esto se debe a la presencia de ciertas características (ver **figura 1.5**) que le han dado una gran aceptación, como su gran fiabilidad y facilidad de uso, debido a que no genera rechazo en las personas que lo utilizan como es el caso de la retina, y en el reconocimiento del rostro por ejemplo, no se necesita de la autorización de la persona a identificar para utilizarlo.

	<i>Ojo (iris)</i>	<i>Ojo (retina)</i>	<i>Huellas dactilares</i>	<i>Geometría de la mano</i>	<i>Escritura y firma</i>	<i>Voz</i>	<i>Cara</i>
<i>Fiabilidad</i>	Muy alta	Muy alta	Alta	Alta	Media	Alta	Alta
<i>Facilidad de uso</i>	Media	Baja	Alta	Alta	Alta	Alta	Alta
<i>Prevención de ataques</i>	Muy alta	Muy alta	Alta	Alta	Media	Media	Media
<i>Aceptación</i>	Media	Media	Alta	Alta	Muy alta	Alta	Muy Alta
<i>Estabilidad</i>	Alta	Alta	Alta	Media	Baja	Media	Media

Figura 1.5: Comparación de características de sistemas biométricos (12).

1.5 Sistemas biométricos. Reconocimiento basado en rasgos faciales

1.5.1 Algoritmos de reconocimiento facial

Los algoritmos de reconocimiento facial automático constan de un conjunto de pasos que se ejecutan a partir de que se tiene una imagen digital en la que se encuentra el rostro de la persona que se quiere reconocer. El primero de estos pasos consiste en la detección facial dentro de la imagen y el segundo es el reconocimiento del rostro detectado (15).



1.5.1.1 Algoritmo para la detección de rostros

Es imprescindible para el correcto funcionamiento de un algoritmo de reconocimiento que el rostro sea localizado de manera precisa. En la actualidad existe una gran cantidad de algoritmos de detección de rostros con desempeño variado y dependiente de los escenarios a considerar (16). Una posible clasificación de los algoritmos de detección de rostros es:

- Métodos basados en rasgos faciales: buscan encontrar aquellas características presentes en cualquier rostro: ojos, cejas, labios, boca, mentón, líneas de contorno, etc.

Dentro de las técnicas basadas en rasgos faciales se pueden hacer diferentes tipos de análisis, como puede ser el análisis de bajo nivel, este no es más que técnicas que trabajan a nivel de píxel, ejemplo de este son los bordes, color y el video.

- Métodos basados en la imagen: aplican herramientas generales de reconocimiento de patrones para sintetizar un modelo a partir de un conjunto de imágenes de entrenamiento. Trabajan con la imagen completa o una región, sin buscar rasgos faciales de forma localizada.

El objeto de estudio de las técnicas basadas en la imagen es la propia imagen. El conocimiento previo se incorpora implícitamente en esquemas de entrenamiento. Se trabaja directamente con representación de la imagen a la que se le aplican algoritmos de entrenamiento y análisis.

Dentro de esta técnica se encuentran los siguientes métodos:

- Redes neuronales: Es uno de los métodos más utilizado en la detección de rostros en una imagen, dado el alto porcentaje de aciertos que produce, siendo en algunos casos superiores al 95% (16). La red neuronal se entrena usando un conjunto de imágenes que representan rostros de todo tipo (de varias razas, tonos de piel, con y sin gafas, pendientes, posiciones de los labios y ojos, ligeras rotaciones, e incluso caras humanoides) y otro conjunto de imágenes que no representan rostros, de forma que la red neuronal pueda establecer el criterio adecuado acerca de lo que es un rostro y lo que no lo es. La respuesta de la red neuronal ante una imagen de entrada es la de decidir si dicha imagen corresponde o no a un rostro, es decir, una respuesta binaria.
- Métodos estadísticos: Este tipo de algoritmo no asume ningún tipo de información previa de la tipología de un rostro; sino, que a partir de un conjunto de muestras (imágenes con y sin rostros) de entrenamiento extraen la información relevante que diferencia un objeto rostro de un objeto no rostro. Uno de los trabajos de detección de rostros en tiempo real es el esquema algorítmico



desarrollado por Viola y Jones que se basa en el aprendizaje de AdaBoost (17) para construir un clasificador no lineal.

En la tabla 1.1 se realiza una pequeña comparación de las técnicas más importantes de la fase de detección.

		Ventajas	% Efectividad	Desventajas
Técnicas basadas en rasgos faciales	Análisis de bajo nivel en el color	Minimizan los efectos que tienen las variaciones de iluminación o pose.	90%	Dificultad al localizar ciertos rasgos faciales por oclusión.
	Red Neuronal	Alto porcentaje de aciertos que produce, superior al 95%.	95%	Se caracteriza por ser lento.
Técnicas basadas en imágenes	Métodos Estadísticos: AdaBoost desarrollado por Viola y Jones	Procesa imágenes extremadamente rápido. Eficaz y rápido en término de detección de rostro.	99%	Genera falsos positivos, al reconocer objetos en la imagen que no son rostros.

Tabla 1.1: Comparación de los métodos de detección de rostros (17).

1.5.1.2 Algoritmos para el reconocimiento de rostros

Los algoritmos para el reconocimiento del rostro se clasifican en: métodos holísticos y métodos basados en características locales (16). El primero consiste en utilizar toda la imagen de la cara como entrada al sistema de reconocimiento siendo la unidad básica de procesamiento y el segundo se basa en extraer



características locales, como los ojos, nariz, boca, entre otras. Sus posiciones y estadísticas locales constituyen la entrada al sistema de reconocimiento. En la **figura 1.6** se muestra los algoritmos a conocer según el método que se seleccione para el reconocimiento del rostro.

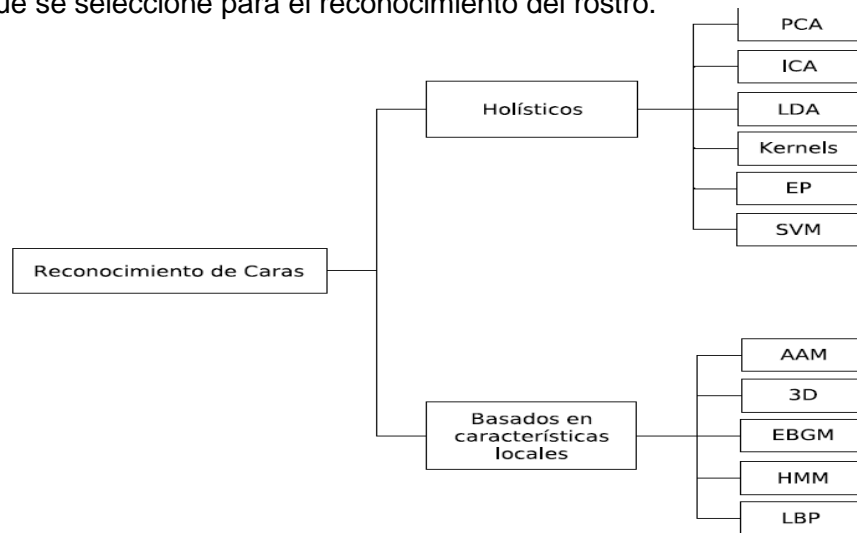


Figura 1.6: Clasificación de algunos de los métodos de reconocimiento del rostro (16).

1.6 Soluciones existentes en el mundo

1.6.1 Empresas que utilizan el reconocimiento del rostro en el mundo

- La empresa Google desarrolló un producto llamado Picasa y una herramienta de búsqueda de reconocimiento de rostros (18). Cuenta con un software que permite subir álbumes de fotos en Internet para incorporarla a un banco de imágenes. La herramienta busca e identifica automáticamente el rostro de cualquier persona que se le indique y muestra todas las imágenes donde crea que se encuentre. Es multiplataforma.
- Facebook ha creado una herramienta de reconocimiento facial, que permite reconocer al usuario mediante una foto (19). Sin embargo, en este caso, han tenido que desactivar su tecnología de reconocimiento facial y borrar los datos incluidos en los perfiles de sus usuarios, debido a la Comisión de Protección de Datos (DPC), ya que exigen una mayor transparencia para los usuarios sobre la manera en que se tratan sus datos.
- ZKSoftware Inc: Es una compañía china de alta tecnología especializada en inversiones de riesgo, finanzas, valores, bienes y la producción de tecnología biométrica. Dentro de las tecnologías que ha desarrollado se encuentra el iFace 302 (20), es un equipo de identificación multi-biométrico,



integra la última plataforma de tecnología Zk Face 5.0 y Zk Finger 10.0, contiene una cámara infrarroja de alta definición que permite identificar personas en ambientes oscuros. Otras de las características que presenta es que dispone de diferentes métodos de identificación como, lector facial, lector de huellas dactilares, y contraseña.

- **Identix:** Es una de las compañías que fabrican tarjetas de identidad con datos biométricos como el iris o huellas dactilares. Identix provee la tecnología que verifica la captura facial y la firma digital. Ofrece soluciones para una amplia gama de aplicaciones para mercados que incluyen seguridad de las empresas corporativa, intranet, extranet, internet, acceso celular a Internet y la seguridad, comercio electrónico. Identix ofrece soluciones como el Facelt, Mobile-Eyes, ABIS System para el trabajo con datos biométricos (21). ABIS System brinda una plataforma para el reconocimiento multi-biométrico y provee una exactitud nunca antes alcanzada mediante el uso de un sistema multi-modular. Este sistema incluye reconocimiento facial, huella dactilar y de iris.
- **DATYS:** Empresa de Desarrollo de Aplicaciones, Tecnologías y Sistemas creada en el año 2006 de capital 100 % cubano, dedicada al desarrollo de aplicaciones informáticas en diversas líneas de negocios, a las que aporta soluciones con sus productos en las áreas de la identificación biométrica, la gestión de contenidos, la minería en textos, la seguridad técnica integral, la gestión de sistemas empresariales, hoteleros y de agencias de viajes (22). La empresa creó un producto llamado Biomesys Face, es un sistema de identificación biométrica por rostro, diseñado para reducir los tiempos y aumentar la efectividad en el reconocimiento de las personas a partir de las características faciales, sustentado en estándares biométricos internacionales y sobre una arquitectura orientada a servicios.
- **CENATAV:** Es un centro orientado a las investigaciones teóricas y aplicadas en el área del Reconocimiento de Patrones y la Minería de Datos, creado en Cuba en el año 2004 y se encuentra en proceso de consolidación (5). Las investigaciones incluyen en la actualidad el procesamiento digital de imágenes y señales, la teledetección, el reconocimiento lógico combinatorio de patrones, el reconocimiento sintáctico estructural, la teoría de testores, algoritmos conceptuales, análisis de texturas, la interpretación conceptual de datos espaciales, la minería de texto, la minería de datos mezclados, entre otras. Las aplicaciones están dirigidas a áreas tales como la biometría, la recuperación de información, la prospección geológica, procesamiento de información de texto, entre otras.



- CENATAV-DATYS: Fruto de esta unión es el módulo Analytics (23), el cual permite el análisis de la señal de forma automática para el reconocimiento de matrículas de vehículos y de rostros. Este sistema fue obtenido a partir de la integración de los algoritmos para la identificación de chapas creado por Cenatav y el producto antes creado Biomesys por la empresa Datys. Actualmente se utiliza en todos los puntos de control del país con una aplicación operativa desarrollada por especialistas de la Policía Nacional Revolucionaria (PNR) y también es de gran utilidad para el Minint con vistas al control de carros circulados.

Se estudiaron soluciones de las distintas empresas mencionadas anteriormente llegando a la conclusión, de que las mismas tienen un alto costo en el mercado dificultando la adquisición de alguno de estos sistemas, además las herramientas necesarias para su utilización son privativas por lo que se propone desarrollar un sistema de identificación de personas basado en rasgos faciales en el CISED, que permita llevar el control de las personas en el centro a través de sus datos identificativos y características faciales. Este consta de una aplicación dirigida por un ordenador que identifica automáticamente a una persona mediante una imagen digital. Esto es posible mediante un análisis de las características faciales del sujeto extraídas de la imagen y comparadas con una base de datos.

1.7 Propuesta de los algoritmos a utilizar en el sistema de reconocimiento facial

1.7.1 Procesos para el reconocimiento facial

Los Sistemas de Reconocimiento Facial operan de dos formas diferentes: verificación (autenticación) e identificación. Los sistemas de verificación, realizan una solicitud de identidad al usuario por medio de un identificador artificial (contraseña, tarjeta, etc), luego la identidad es verificada capturando la muestra biométrica del usuario y comparándola con la muestra almacenada que corresponde a dicho identificador (24). Por su parte, los sistemas de identificación capturan la muestra biométrica del usuario y determinan, mediante una búsqueda en una base de datos de una muestra biométrica aceptablemente aproximada, la identidad del usuario.

El sistema a desarrollar presenta varias fases, estas consisten en la adquisición o captura del rostro del usuario, detección del rostro y normalización de la imagen, localización de los puntos principales, extracción de características y la confrontación con la base de datos.



1.7.1.1 Algoritmo a utilizar para la detección de rostros de la persona

La primera etapa en el proceso de identificación de personas basado en rasgos faciales es la captura del rostro del usuario o la adquisición de la imagen de la persona a identificar, realizándose el proceso de identificación de la misma manera sin interesar la vía seleccionada.

Una vez que haya sido capturada la imagen se procede a la detección del rostro y la normalización de la imagen.

La detección de rostros puede ser considerada como un caso específico de la detección de objetos. Lo que se pretende es localizar el rostro y en caso de su existencia devuelve su ubicación en la imagen y sus dimensiones. La localización facial se realiza mediante la detección de características faciales, como los ojos, la boca, etc (15).

Existen varios factores que impiden la localización del rostro (25). Uno de estos factores es la posición de este respecto a la cámara; otro factor sería la presencia o ausencia de componentes estructurales como la barba, el bigote o anteojos, de estos hay una gran cantidad con una gran variabilidad (color, tamaño, forma). La expresión facial, aunque afecta más al reconocimiento facial, también puede impedir la localización facial. Otros factores como la oclusión parcial del rostro por objetos o por otros rostros, o simplemente condiciones de iluminación y características de la cámara, como puede ser el sensor, afectan a la localización de un rostro.

Para el presente trabajo se utiliza en la fase de detección las **técnicas basadas en la imagen**, ya mencionado anteriormente como una de las metodologías a utilizar en el proceso de detección, las cuales establecen dos métodos principales, los basados en redes neuronales y los estadísticos siendo este último el más aceptable en rendimiento, por lo que se elige este método para la implementación de la fase de detección de rostro. El algoritmo que se utiliza basado en este método es AdaBoost, también conocido como Viola y Jones (17).

AdaBoost se basa en la detección de objetos y posee además una buena efectividad (17). Reduce 15 veces el tiempo necesario para la detección de objetos en general, logrando realizar este proceso en tiempo real. El algoritmo clasificador consiste en un árbol de decisión compuesto por una serie de clasificadores débiles, los cuales están dispuestos en forma de cascada formando una sucesión, lo cual trae como resultado un clasificador fuerte con alto grado de efectividad. Luego del estudio se considera que AdaBoost, es aplicado en la etapa de detección de rostros en el sistema de identificación de persona basado en rasgos faciales.



El algoritmo utiliza una ventana que recorre la imagen comprobando en cada posición la existencia del objeto con el cual fue entrenado, en este caso es un rostro. El tamaño de la ventana está dado por el tamaño mínimo de los rostros que se desean detectar y una vez finalizado el recorrido de la imagen es aumentado sucesivamente hasta cubrir el total de la imagen. Finalmente, el clasificador devuelve una secuencia de rectángulos que indican las posiciones de los rostros detectados en la imagen (ver **figura 1.7**). Luego de ser detectado el rostro lo próximo sería segmentar el área donde se encuentra el rostro en la imagen (ver **figura 1.8**).

La biblioteca Emgu cv, que posteriormente se aborda información acerca de la misma, cuenta con un clasificador de objetos basado en el método AdaBoost, el cual se utiliza para llevar a cabo el proceso de detección de rostro del sistema de identificación, siendo así la función que permite desarrollar dicho paso es: `cvDetectHaarCascade(cascade, scale factor, min neighbors, flags, min size)`, donde el parámetro cascade es el entrenamiento cargado a partir de un fichero XML, scale factor es el factor con que aumenta el tamaño de la ventana de búsqueda, por defecto es 1.1 que representa aumento del 10%, min neighbors es el número mínimo de rectángulos vecinos que conforman un objeto y min size es el tamaño mínimo de la ventana de búsqueda.

Luego de ejecutada, la función retorna una secuencia de rectángulos que representan la posición y el área que ocupan los rostros detectados dentro de la imagen. A partir de esta secuencia y la imagen original, cada rostro es extraído y almacenado como una imagen para continuar a la siguiente etapa de reconocimiento.



Figura 1.7: Detección del rostro de la persona.



Figura 1.8: Imagen recortada o segmentada.



1.7.1.2 Procesos para la normalización de la imagen

Para la realización del sistema propuesto primeramente se realiza un pre-procesado de las pupilas de los usuarios (27) disponiendo de las coordenadas de las mismas. Este método se efectúa para poder realizar de forma efectiva la normalización, alineación y rotación de las imágenes.

Primeramente se obtiene el punto medio del ojo, este punto se va a obtener aplicando el algoritmo Active Shape Model (ASM) que en próximos epígrafes se explica con más detalles de su funcionalidad. Una vez que son localizados estos puntos se espera que los usuarios van a estar mirando hacia el frente, mejorando, además, la colocación de las pupilas de los usuarios. Esto se consigue calculando la distancia interpupilar (por sus siglas en inglés IPD), es decir, la distancia entre los centros de los ojos (27). El objetivo de calcular el IPD en el eje “x” es para normalizar la imagen con el fin de que las pupilas se encuentren situadas siempre en la misma posición, dentro de la imagen. El objetivo de calcular el IPD en el eje “y” es para verificar si la imagen necesita algún tipo de rotación, con el fin de que las pupilas siempre se encuentren al mismo nivel.

En la **figura 1.9** se muestra un ejemplo de cuando el valor de IPD en el eje “y” es diferente de 0, es decir el rostro se encuentra inclinado.



Figura 1.9: Rostro inclinado (27).

En el caso que la distancia inter-pupilar en el eje “y” sea mayor que 0, indica que se necesita realizar la operación de rotación de la imagen con el objetivo de que la recta que une las pupilas en el eje “x” sea paralela a los bordes superior e inferior de la imagen.

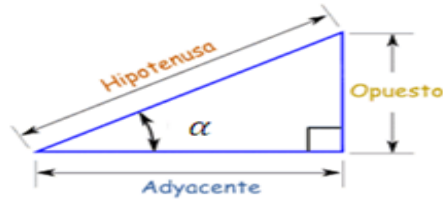
La rotación se utiliza para enderezar una imagen en el plano frontal, tomando como eje de rotación la recta que pasa por los centros de los ojos de los usuarios.

La transformación espacial, llamada transformación o aplicación afín, consiste en una transformación lineal seguida de una traslación o desplazamiento: $x \rightarrow Ax + b$



Observando la fórmula anterior, se aprecia que la transformación se encuentra compuesta por una matriz de rotación A , y un vector de traslación b que toma un valor de 0 porque no se necesita trasladar las pupilas en los ejes.

Para calcular la matriz de rotación se determina el ángulo σ (ver **figura 1.10**) entre la línea que une el centro de las pupilas de los ojos de las imágenes y la línea que une las coordenadas donde se desean que estén situados los centros de las pupilas.



$$\alpha = \pm \arctan \left(\frac{\text{Cateto Opuesto}}{\text{Cateto Adyacente}} \right)$$

Figura 1.10: Determinación del ángulo σ (27).

Con este ángulo, la transformada de rotación se define como:
$$A = \begin{pmatrix} \cos(\alpha) & \sin(\alpha) & 0 \\ -\sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Una vez calculado el ángulo σ se obtiene la imagen rotada (ver **figura 1.11**).



Figura 1.11: Imagen rotada (27).

Luego de haber realizado el pre-procesado de las pupilas, se llega a un segundo proceso de normalización, es decir la normalización de la imagen, etapa clave con el objetivo de poder universalizar los algoritmos desarrollados.

- Normalización de la imagen.

La normalización se basa en el estándar ISO/IEC 19794-5 del 2005 para el procesamiento de imágenes faciales (27). En la tabla 1.2, se muestran las características geométricas que deben tener el tipo de imágenes utilizadas, así como un ejemplo gráfico (**Figura 1.12**).



Características o parámetros	Valor
Ancho de la imagen	W
Alto de la imagen	$W/0.75$
Coordenada "Y" de los ojos	$0.6*W$
Coordenada "X" del primer ojo (derecho)	$(0.375*W)-1$
Coordenada "X" del segundo ojo (izquierdo)	$(0.625*W)-1$
Ancho desde un ojo a otro	$(0.25*W)-1$

Tabla 1.2: Características geométricas de las imágenes.

Es importante mencionar que el ancho de la imagen (W) depende del valor que toma la distancia interpupilar (IPD). En el caso de que el rostro se encuentre inclinado, sería la distancia interpupilar una vez que la imagen haya sido rotada. En la figura 1.12 el ancho de la imagen (W) es igual a un valor de 240 píxeles.

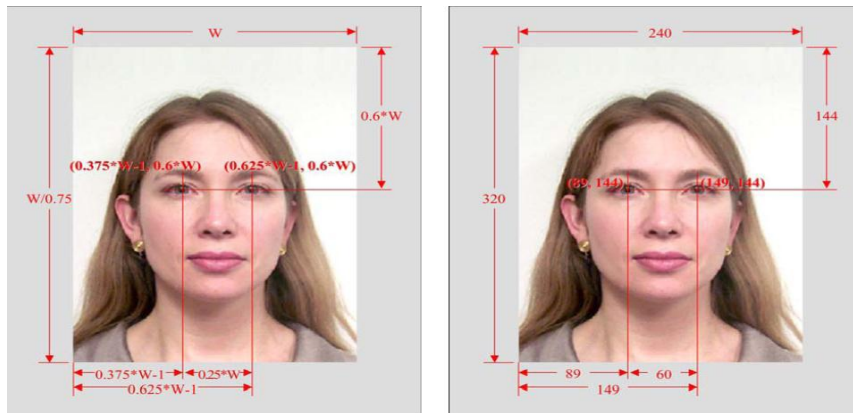


Figura 1.12: Formato geométrico (izquierda), con $W=240$ píxeles (derecha) (27).

- Normalización de la imagen basado en el color.

Para la normalización por color, las imágenes son convertidas a escala de grises (26). Mediante esta operación una imagen es representada mediante 256 tonalidades de gris, lo cual minimiza las variaciones de color. Pero esto no resulta suficiente si se tiene en cuenta que, en muchas ocasiones, las imágenes no son tomadas por profesionales; lo cual ocasiona significativas variaciones de iluminación.



Con el objetivo de estandarizar las condiciones de iluminación de los rostros en el sistema a desarrollar, se utiliza la técnica de ecualización del histograma (26). El histograma de una imagen representa la frecuencia relativa de ocurrencia de varios niveles de gris en la imagen. La ecualización del histograma de una imagen es una transformación que pretende obtener para una imagen un histograma con una distribución uniforme. Es decir, que exista el mismo número de píxeles para cada nivel de gris del histograma de una imagen monocroma.

La función de la ecualización es:

$$v(i, j) = \frac{Fu(u(i, j) - Fu(a))}{1 - Fu(a)}(L - 1) + 0.5$$

Figura 1.13: Fórmula para la ecualización del histograma (26).

En teoría, la operación debería transformar el histograma en otro con una forma perfectamente uniforme sobre todos los niveles de gris. Sin embargo, en la práctica esto no se va a poder conseguir pues se trabaja con funciones de distribución discretas en lugar de continuas. En la transformación, todos los píxeles de un mismo nivel de gris se transforman a otro nivel de gris (ver **figura 1.15**) y el histograma se distribuye en todo el rango disponible separando en lo posible las ocupaciones de cada nivel. El resultado de la ecualización maximiza el contraste de una imagen sin perder información de tipo estructural, esto es, conservando su entropía (ver **figura 1.16**).



Figura 1.14: Representación del histograma de la imagen en sus tres componentes RGB.

Figura 1.15: Representación del histograma de la imagen en escala de grises.

Figura 1.16: Representación del histograma ecualizado de la imagen en escala de grises.

La biblioteca Emgu cv posee la función `cvEqualizeHist(src,dst)` encargada de ecualizar el histograma de una imagen en escala de grises, la cual resulta de mucha utilidad para la normalización de los rostros por color, donde `src` es la imagen de entrada y `dst` la imagen ecualizada.



1.7.1.3 Localización de los puntos característicos

La utilización de las características faciales se ha convertido en un reto en la actualidad, especialmente en aplicaciones donde existen ambientes no controlados que inciden negativamente factores sobre las potencialidades biométricas de los rostros humanos, como son: la pose, la iluminación, la expresión, los accesorios utilizados, entre otros. Debido a esto se ha analizado la influencia de cada uno de ellos en el rendimiento de la aplicación a desarrollar y se decide trabajar con un algoritmo capaz de solucionar dichas inconvenientes, este es conocido como **Active Shape Model (ASM)** (28).

Para desarrollar dicha fase se utiliza el modelo estadístico, Active Shape Model (ASM) y una vez localizados los puntos principales se procede a la extracción de características, para lo cual se acude a un **banco de filtro**.

- **Active Shape Model**

Para la ubicación de los puntos característicos en el rostro del sistema de identificación se aplica el algoritmo Active Shape Model (ASM) (28) que crea un modelo de las características que se desee encontrar. Desarrollado por Tim Cootes Taylor y Chris en 1995. Este modelo se define por una serie de puntos de modelo, así como la conexión entre ellos. Examina la región alrededor de cada punto encontrado en la imagen para encontrar la mejor opción en la imagen que se ajuste a los demás puntos. Mueve el modelo de punto a punto para ubicar de una forma adecuada los demás puntos sobre la imagen. La adecuación se realiza utilizando los bordes de imagen.

En la siguiente figura se muestra como se extraen algunas características de la imagen a tratar, posteriormente dichas características se analizan por el modelo ASM para determinar si dicho objeto es un rostro.



Figura 1.17: Extracción de características de una imagen tras tener previamente detectada el área del rostro (28).

En la **figura 1.18** se muestran los puntos extraídos por ASM, estos puntos se ubican según unas estadísticas que provee el algoritmo, se procede ubicando una plantilla sobre la parte de la imagen que contiene un rostro.



Una vez obtenidos los puntos, se ubican los ojos para realizar el escalamiento para que todos los grafos tengan la misma distancia y luego obtener una imagen a la que solo se extrae el rostro (ver **figura 1.19**).

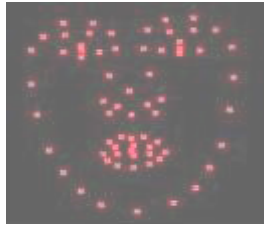


Figura 1.18: Plantilla de ASM (28).

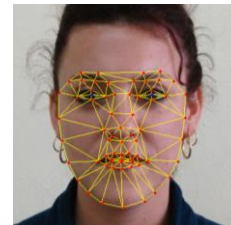


Figura 1.19: Un rostro ubicado por la plantilla de ASM.

1.7.1.4 Extracción de características faciales

La extracción de características es una parte fundamental de un sistema de reconocimiento de personas. En el presente trabajo se utilizan los 68 puntos extraídos del rostro que ubican el contorno de las cejas, ojos, nariz y boca. De cada uno de estos puntos se obtiene un vector característico que se crea una vez que se aplica el filtro correspondiente, esto se muestra en la siguiente figura.

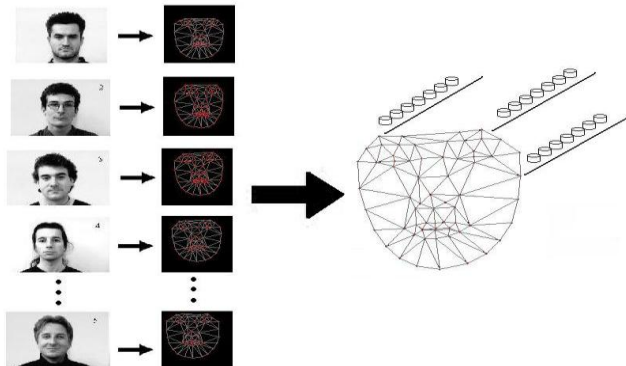


Figura 1.20: Para cada rostro encontrado se crea un grafo con los puntos de ASM y para cada uno de estos puntos se obtiene una vecindad de 7x7 píxeles de la cual se extraen las características correspondientes (28).

- **Filtro a utilizar**
 - **Transformada discreta del coseno (DCT)**

La aplicación de las transformadas al procesamiento digital de señales e imágenes ha aumentado considerablemente en los últimos años. Entre las transformadas más comúnmente utilizadas se destaca el uso de la transformada discreta coseno (DCT: Discrete Cosine Transform, por sus siglas en inglés) (29), siendo la transformada que más se emplea en aplicaciones de compresión de imágenes y de video. Las propiedades únicas de la DCT permiten obtener altos índices de compresión a muy bajo coste. Por tanto,



es la transformada utilizada en los actuales estándares de compresión de imágenes. La transformada discreta del coseno (DCT) es una transformada basada en la transformada discreta de Fourier, pero utilizando únicamente números reales. Expresa una secuencia finita de varios puntos como resultado de la suma de distintas señales sinusoidales, con distintas frecuencias y amplitudes. Como la Transformada Discreta de Fourier (DFT) (29), la DCT trabaja con una serie de números finitos, pero mientras la DCT sólo trabaja con cosenos la DFT lo hace con exponenciales complejas.

La DCT tiene una buena capacidad de compactación de la energía al dominio transformado, es decir, que la transformada de coseno discreta consigue concentrar la mayor parte de la información en pocos coeficientes transformados. El algoritmo aplicado no varía con los datos que recibe, como sucede en otros algoritmos de compresión. Produce pocos errores en los límites de los bloques imagen. La minimización de los errores en los bloques imagen permite reducir el efecto de bloque en la imágenes reconstruidas.

Las entradas y salidas de esta fase se resumen en la siguiente figura:

$$B_{pq} = \alpha_p \alpha_q \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{m,n} \cos \frac{\pi (2m+1)p}{2M} \cos \frac{\pi (2n+1)q}{2N}, \quad \begin{matrix} 0 \leq p \leq M-1 \\ 0 \leq q \leq N-1 \end{matrix}$$

$$\alpha_p = \begin{cases} \frac{1}{\sqrt{M}}, & p = 0 \\ \sqrt{\frac{2}{M}}, & 1 \leq p \leq M-1 \end{cases} \quad \alpha_q = \begin{cases} \frac{1}{\sqrt{N}}, & q = 0 \\ \sqrt{\frac{2}{N}}, & 1 \leq q \leq N-1 \end{cases}$$

Figura 1.21: Fórmula para calcular la DCT en una matriz 2D (29).

Una vez que se hayan seleccionado los 68 puntos extraídos del rostro, se obtiene para cada uno de ellos una vecindad de 7x7 píxeles de la cual se extraen las características correspondientes. Estas características son extraídas a partir del filtro de la Transformada Discreta del Coseno (DCT), es decir para cada punto se va a obtener una matriz de transformación a partir de la vecindad antes definida (7x7 píxeles = 49 coeficientes) (ver figura 1.22) donde la esquina superior izquierda recoge los valores más significativos (color rojo). Luego de ser distribuidos estos valores se realiza un recorrido en zig-zag a la matriz dando como respuesta el vector característico.

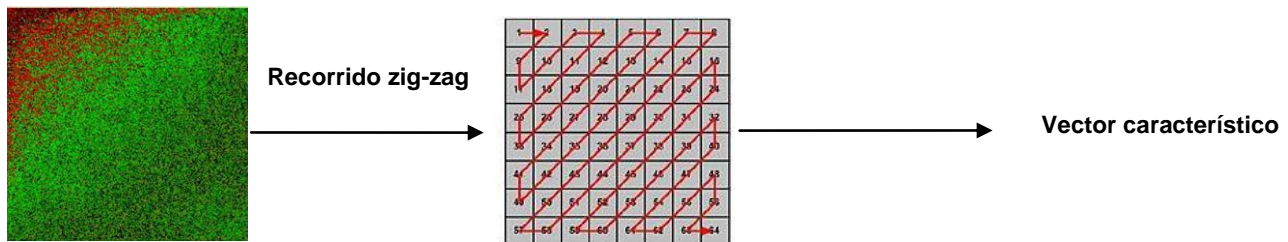


Figura 1.22: Vecindad extraída para cada uno de los puntos antes localizados con un tamaño de 7x7 píxeles a la cual se le aplica



el recorrido en zig-zag dando como salida el vector característico correspondiente.

1.7.1.5 Confrontación con la base de datos

Finalmente se tiene la etapa de confrontación con la base de datos que compara descriptores de rostros para el reconocimiento. Las tareas que se desarrollan en esta etapa dependen del modo de funcionamiento, ya sea identificación o verificación. En modo de identificación, el sistema postula una identidad para el usuario entrante comparando sus características contra las de todos los individuos almacenados en la base de datos. En modo de verificación el usuario es quien postula una identidad, a través de un PIN u otro identificador, que es validada por el sistema (28).

En la **figura 1.23** se muestra cómo se lleva a cabo el proceso de comparación con la base de datos utilizando la técnica ASM para la localización de los puntos principales en el rostro (28).

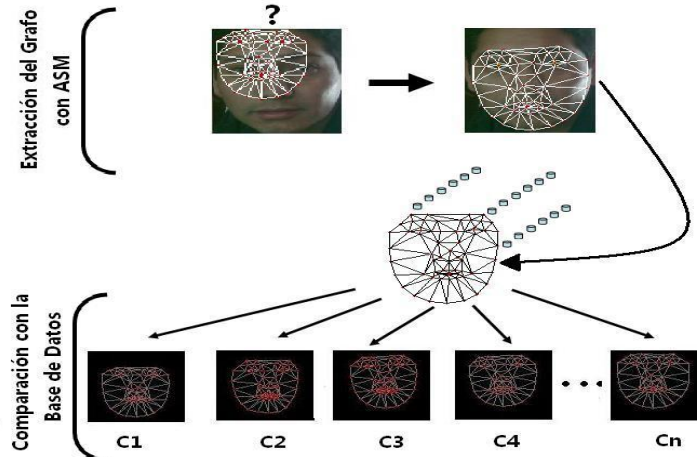


Figura 1.23: Proceso de comparación con la base de datos (28).

Este proceso ayuda a evaluar si dos elementos son iguales entre ellos. Esto se puede saber una vez que sean comparados a partir de la fórmula de distancia entre dos vectores (29). Su procedimiento es el siguiente:

- Dada una imagen de entrada la aplicación intenta encontrar otra imagen con un vector similar en la base de datos.

$$D = \sqrt{\sum_i W_{ri}(DR_i - DR'_i)^2} + \sqrt{\sum_i W_{gi}(DG_i - DG'_i)^2} + \sqrt{\sum_i W_{bi}(DB_i - DB'_i)^2}$$

Figura 1.24: Ecuación para el cálculo de la distancia entre dos vectores (29).



El subíndice i representa el orden de los coeficientes de la matriz, conjuntamente se utiliza el peso de los coeficientes (w) con el fin de ajustar el rendimiento del proceso del emparejamiento, en este caso (w) toma un valor 1. Analizando la fórmula, se puede inferir que:

- 2 imágenes son idénticas si la distancia es 0.
- 2 imágenes son similares si la distancia es cercana a 0.

Por tanto, este proceso de emparejamiento permite identificar imágenes similares. La complejidad del proceso para encontrar coincidencias es baja, y con este método se pueden alcanzar búsquedas de alta velocidad.

El objetivo es identificar personas basado en sus rasgos faciales, por lo tanto, una vez obtenido el vector característico para cada usuario y luego comparar estos datos descriptivos con una base de datos ya procesada basado en la técnica del emparejamiento se identificada la persona mostrando los datos identificativos de la misma como son el nombre, apellidos, carné de identidad y sexo.

En la última fase se trabaja con la base de datos BiID (30). Esta base de datos se puede obtener de forma gratuita en internet, ha sido creada y publicada para dar la posibilidad a todos los investigadores que trabajan en el área del reconocimiento facial, de comparar la calidad de sus algoritmos de detección de rostro y puntos característicos faciales, con otros. Incluye 1521 imágenes en escala de grises, todas ellas de un tamaño de 384x286 píxeles. La base de datos contiene imágenes de caras de 23 sujetos distintos, con una gran variedad de condiciones de iluminación, tamaños de cara, expresiones y fondos complejos. La **figura 1.25** muestra algunos ejemplos de dicha base de datos.



Figura 1.25: Tres usuarios, a modo de ejemplo, de la base de datos BiID (30).

1.8 Lenguajes, Metodologías y Herramientas de desarrollo

1.8.1 Metodología para el desarrollo del software

Todo desarrollo de software es riesgoso y difícil de controlar, siéndolo aún más, cuando no se siga una guía que pauté los pasos a seguir y muestre cómo actuar ante las dificultades. Este papel lo juegan las metodologías de desarrollo de software (31), ellas definen Quién debe hacer Qué, Cuándo y Cómo debe



hacerlo. Sin ellas, el resultado de un proyecto sería desastroso, con clientes insatisfechos con el resultado, malas planificaciones con fechas límite imposibles de cumplir y desarrolladores aún más insatisfechos.

Si se toma como criterio las notaciones utilizadas para especificar artefactos producidos en actividades de análisis y diseño, se pueden clasificar las metodologías en dos grupos: Metodologías Estructuradas y Metodologías Orientadas a Objetos. Por otra parte, considerando su filosofía de desarrollo, se pueden clasificar en dos grupos: Metodologías Tradicionales (o Pesadas) y Metodologías Ágiles (31).

Es esta última clasificación la que se tiene en cuenta en el desarrollo del presente trabajo. Las metodologías tradicionales son aquellas con mayor énfasis en la planificación y control del proyecto, en especificación precisa de requisitos, tienen una rigurosa definición de roles, actividades y artefactos, incluyendo modelado y documentación detallada. Sin embargo, las metodologías ágiles están más orientadas a la generación de código con ciclos cortos de desarrollo, se dirigen a equipos de desarrollo pequeños, hacen especial hincapié en aspectos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso.

Basándose en este planteamiento, se hizo un estudio entre una metodología tradicional y dos ágiles, en el primer caso se tendrá en cuenta el Proceso Unificado de Rational (RUP), en el segundo caso se analizó Desarrollo Guiado por la Funcionalidad (FDD) y Programación Extrema (XP) (32).

1.8.1.1 RUP

RUP (Rational Unified Process) es uno de los procesos más riguroso y completo de los existentes hoy día, pensado para cualquier proyecto, siempre y cuando se configure (30). Consta de 4 fases: Inicio, Elaboración, Construcción y Transición. Está distribuido por 9 flujos de trabajo, 6 de ellos considerados ingenieriles: Modelado de negocio, Requisitos, Análisis y Diseño, Implementación, Prueba y Despliegue y los otros tres son considerados flujos de apoyo: Gestión de proyecto, Gestión de Configuración y Cambio, Despliegue. Cada fase es realizada por iteraciones, y en cada iteración se desarrollan actividades de los flujos de trabajo, algunos más que otros, dependiendo del avance que tenga el software.

Es un proceso dirigido por casos de uso, ya que desde el modelado del negocio se captan en forma de casos de uso lo que ocurre en la organización, de cada uno se realizan diagramas de actividades para seleccionar las actividades a ser automatizadas, estas son la entrada para determinar las funcionalidades del sistema, las cuales se expresan mediante casos de uso del sistema, luego se realiza cada caso de uso mediante el análisis y diseño, para su posterior implementación y prueba. Es centrado en la arquitectura,



la cual captura los elementos más importantes del sistema, y los refleja mediante vistas arquitectónicas. Es iterativo e incremental, ya que en cada iteración se desarrollan actividades de todos los flujos de trabajo, y cada resultado es un incremento para el software.

1.8.1.2 FDD

FDD (Feature Driven Development) es un proceso iterativo, con iteraciones cortas (2 semanas), las cuales producen un software funcional que los clientes pueden ver y monitorizar (32). Ha sido pensado para proyectos con un tiempo de desarrollo menor de dos años. Las iteraciones se deciden teniendo en cuenta las funcionalidades, que representan fragmentos del software con significado para el cliente. FDD consta de 5 fases: desarrollo de un modelo global, construcción de la lista de funcionalidades, planificación por funcionalidad, diseño por funcionalidad e implementación por funcionalidad. El trabajo tanto de modelado como de desarrollo se realiza en grupo.

Las funcionalidades a implementar en una entrega son repartidas entre los distintos subgrupos del equipo, las clases de software tienen propietario, es decir, sólo el creador puede modificarlas. Es por ello que el equipo que implementa cierta funcionalidad tiene que tener todos los propietarios de las clases implicadas. Implementar una funcionalidad lleva implícito la preparación y ejecución de pruebas, así como revisión del código e integración de las partes que componen el software.

1.8.1.3 XP

XP (Extreme Programming) es una metodología que intenta reducir la complejidad del software por medio de un trabajo orientado al objetivo, basado en las relaciones interpersonales y la velocidad con la que trabajan (33). Intenta minimizar los fallos en el proceso, a través de la colaboración permanente de algún representante del cliente, el cual debe tener conocimientos profundos, para responder de una forma correcta y en un tiempo breve cualquier pregunta realizada por algún miembro del equipo de desarrollo.

Define historias de usuario para describir las funciones del sistema, las cuales son escritas por el cliente. Se crea un plan de entrega entre clientes y equipo de desarrollo, tomando como base las historias de usuario y la arquitectura. En cada entrega se discuten entre las partes los objetivos y se definen las iteraciones necesarias para cumplir los objetivos. Cada iteración tiene como resultado un programa que se entrega al cliente para que lo analice, en base a su respuesta se planifican las próximas iteraciones, y si no está de acuerdo se ajusta el plan de entrega hasta que se satisfagan sus necesidades.

Los proyectos realizados bajo esta tecnología cumplen con lo estrictamente necesario en su funcionalidad en el momento necesario: hacer lo que se necesita cuando se necesita. En XP no es



conveniente precipitarse o adelantarse a las tareas que se han establecido previamente sin el consentimiento del cliente, estos hechos conllevan a inyectar complejidad al sistema, alejándolo del concepto de simplicidad. En la figura 1.26 se muestran las fases que rigen la propia metodología de desarrollo (32).

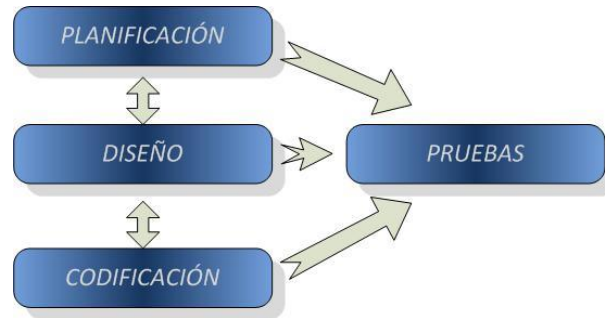


Figura 1.26: Fases de la Metodología XP (33).

La codificación en XP es desarrollada por parejas, por lo que se espera un software con rigurosa calidad. El código de cada pareja es socializado entre los miembros del equipo, de forma tal que puede ser modificado por cualquier integrante, y las parejas son rotativas, de forma tal que todos tengan conocimiento del software.

1.8.1.4 Selección de la metodología de desarrollo de software

Para la selección de la metodología de desarrollo a utilizar, se hace necesario analizar el entorno en el que se construirá el producto.

En primer lugar, es necesario destacar que el usuario final será el propio CISED, pues es el que tienen la necesidad de un sistema de identificación de personas basado en rasgos faciales, lo cual significa que en todo momento se estará interactuando con el mismo. Otro de los factores es que los requisitos asociados a la aplicación pueden sufrir cambios debido a que el centro puede arrojar nuevas necesidades y funcionalidades a incluir durante el proceso de desarrollo del sistema. Y mencionar además el equipo de desarrollo, que cuenta con poco personal, lo cual dificulta la adopción de roles en cada etapa así como la generación de múltiples artefactos.

Se adoptaron los valores de XP, comunicación, sencillez, retroalimentación, coraje y respeto. La comunicación entre el equipo de desarrollo y los clientes será maximizada y efectuada de forma directa e interpersonal con el fin de evitar los problemas y errores causados por la mala adopción. Se realizará un trabajo tan simple como sea posible, favoreciendo la comunicación, reduciendo el código sin utilizar y garantizando la calidad. Se estará constantemente midiendo el sistema para conocer cuánto se acerca a



las funcionalidades necesarias mediante pruebas por parte de los usuarios. Los valores antes mencionados contribuirán a enfrentar con coraje cada cambio en los requisitos y refactorizaciones al sistema.

Lo antes descrito evidencia cómo las características del sistema a desarrollar se inclinan a adoptar las prácticas y valores de la **metodología XP**, por lo cual se selecciona la misma como rectora del proceso de desarrollo del software.

1.8.2 Lenguaje de modelado

Para el desarrollo del sistema de identificación de personas basado en rasgos faciales se tiene como propuesta inicial utilizar: como lenguaje de modelado el Lenguaje Unificado de Modelado UML (34), por sus siglas en inglés, es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Se escoge UML porque permite modelar sistemas utilizando técnicas orientadas a objetos. Mediante él se pueden especificar todas las decisiones de análisis y diseño, construyéndose así modelos precisos, no ambiguos y completos. Además puede conectarse con lenguajes de programación (ingeniería directa e inversa) y permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones).

1.8.3 Herramienta de modelado

Dentro de las herramientas claves en el desarrollo de aplicaciones informáticas se encuentran las herramientas de Ingeniería de Software Asistida por Ordenador (CASE) (por sus siglas en inglés), las cuales son las encargadas de ayudar en el ciclo de desarrollo, con el fin de aumentar la productividad y reducir el costo en términos de tiempo y dinero (34). Las herramientas CASE se han venido ampliando y desarrollando, algunos de sus ejemplos son: Microsoft Project, Rational Rose, JDeveloper, MagicDraw, Visual Paradigm, Microsoft Visio, Enterprise Architect, entre otros. En la presente investigación se analizaron solo dos de ellas dentro de las que se encuentran:

1.8.3.1 Visual Paradigm

Es una de las herramientas CASE más usadas, es la suite creada por Visual Paradigm International (VPI) (35). VPI es un proveedor de soluciones informáticas que incluye organizaciones para desarrollar aplicaciones de calidad, rápidas y baratas. Está compuesta por productos que facilitan a las organizaciones la visualización y diseño de diagramas. Sus soluciones se enfocan en eliminar la complejidad, aumentando así la productividad y disminuyendo el tiempo de desarrollo de las aplicaciones



informáticas. Visual Paradigm para UML (VP) en su versión 8.0 Enterprise Edition es una herramienta muy completa y permite la representación de varios artefactos.

1.8.3.2 Rational Rose

Rational Rose es una herramienta de producción y comercialización establecida por Rational Software Corporation. Utiliza el Lenguaje Unificado (UML) para facilitar la captura de dominio de la semántica, la arquitectura y el diseño (36). Este software tiene la capacidad de crear, ver, modificar y manipular los componentes de un modelo. No es gratuito, admite la integración con otras herramientas de desarrollo. Habilita asistentes para crear clases y provee plantillas de código que pueden aumentar significativamente la cantidad de código fuente generada.

Después de haber realizado un estudio entre estas dos herramientas de modelado se llegó a la conclusión que se utilizará Visual Paradigm en su versión 8.0 para el sistema de identificación de personas basado en rasgos faciales porque se cuenta con mayor experiencia en el trabajo con esta herramienta. Además Visual Paradigm tiene la ventaja que es software libre. Permite realizar ingeniería tanto directa como inversa y además es una herramienta colaborativa, es decir, soporta múltiples usuarios trabajando sobre el mismo proyecto; genera la documentación del proyecto automáticamente en varios formatos como web o PDF y es además una herramienta multiplataforma.

1.8.4 Lenguajes de desarrollo

Los lenguajes de programación son herramientas que permiten crear programas y software. Es un idioma artificial diseñado para expresar procesos que pueden ser llevadas a cabo por máquinas como las computadoras. Durante el desarrollo del presente trabajo se analizaron los lenguajes de programación C++, Java y C# de los cuales se trataron las características principales.

1.8.4.1 Visual C++

C++ es un lenguaje de programación de propósito general que ofrece economía sintáctica, control de flujo y estructuras sencillas y un buen conjunto de operadores (33). No es un lenguaje de muy alto nivel y más bien un lenguaje pequeño, sencillo y no está especializado en ningún tipo de aplicación. Esto lo hace un lenguaje potente, con un campo de aplicación ilimitado y sobre todo, se aprende rápidamente. En poco tiempo, un programador puede utilizar la totalidad del lenguaje.

1.8.4.2 Lenguaje Java

Java está inspirado en C++ y se proyectó con la finalidad de obtener un producto de pequeñas dimensiones, simple y portátil sobre diferentes plataformas y sistemas operativos ya sea a nivel de código



fuelle como a nivel de código binario. Es un lenguaje orientado a objetos, eso implica que su concepción es muy próxima a la forma de pensar humana, genera ficheros de clases compiladas, pero estas son en realidad interpretadas por la máquina virtual de Java, quien mantiene el control sobre las clases que se estén ejecutando. El mismo es multiplataforma y seguro (33). Su sintaxis ha sido trabajada mejorando la de C++ logrando mayor sencillez y legibilidad. Presenta mayor robustez al simplificar la gestión de memoria y eliminar las complejidades del manejo explícito de punteros. Se puede compilar y ejecutar en cualquier plataforma de sistema operativo por ejemplo en Windows, Solaris o Linux gracias a su máquina virtual. La ejecución de programas escritos en Java suele comportarse más lenta que la de aplicaciones de otro lenguaje haciendo un uso voraz de recursos como memoria y procesador. Esto se hace más notorio si la ejecución se basa en cálculos matemáticos complejos o si la aplicación presenta un diseño cargado de componentes visuales.

1.8.4.3 Lenguaje C#

Es un lenguaje de programación orientado al desarrollo y estandarizado por Microsoft como parte de su plataforma .NET, que más tarde fue aprobado como un estándar por la ISO (37). Aunque C# forma parte de la plataforma .NET, ésta es una interfaz de programación de aplicaciones (API), mientras que C# es un lenguaje de programación independiente diseñado para generar programas sobre dicha plataforma. Algunas de las características que presenta (37):

- Sencillez.
- Modernidad.
- Orientación a objetos.
- Orientación a componentes.
- Gestión automática de memoria.
- Instrucciones seguras.
- Extensibilidad de tipos básicos.
- Extensibilidad de operadores.
- Extensibilidad de modificadores.
- Compatibilidad.

Para la implementación del sistema se utilizan los lenguaje de programación C# y C++.

El lenguaje de programación C# fue seleccionado para la implementación del sistema, ya que es mucho más sencillo e intuitivo que cualquiera de los otros lenguajes, es orientado a objetos diferenciando este



enfoque respecto al de otros lenguajes. Tiene como propósito general ofrecer una economía sintáctica, control, flujo y estructuras sencillas y un buen conjunto de operadores. Se ahorra tiempo en la programación ya que presenta bibliotecas de clases muy completa y bien diseñada y sobre todo se alinean a las necesidades del cliente de una aplicación que presente gran velocidad en la manipulación de los datos así como una interfaz que sea lo más rápida posible y que permita una fácil navegación. Además el equipo de desarrollo cuenta con suficiente experiencia en la plataforma .NET y en este lenguaje. El lenguaje C++ se utiliza para el uso de las bibliotecas de clase es decir para poder interactuar con el algoritmo ASM ya que esta última fue creada en el propio lenguaje.

1.8.5 Entorno de Desarrollo Integrado (IDE)

Los entornos de desarrollo integrado (IDEs) son herramientas pensadas para escribir, compilar, depurar y ejecutar programas. Esta herramienta puede estar pensada para su utilización con un único lenguaje de programación o bien puede dar cabida a varios de estos. Hoy en día los entornos de desarrollo proporcionan un marco de trabajo para la mayoría de los lenguajes de programación existentes en el mercado como es el caso de C++, C#, Java, Python y Visual Basic (37).

Entre los entornos integrados de desarrollo que proporciona un marco de trabajo para el lenguaje seleccionado se encuentran los siguientes:

- Visual Studio.net.
- SharpDevelop.
- Mono.
- MonoDevelop.

Para el desarrollo de la aplicación se estudiaron dos de ellos, Visual Studio.net y el MonoDevelop.

1.8.5.1 Visual Studio .NET

Es un entorno de desarrollo integrado para sistemas operativos Windows. Soporta varios lenguajes de programación tales como C++, C#, J#, ASP.NET y Visual Basic .NET, aunque actualmente se han desarrollado las extensiones necesarias para muchos otros (38). Visual Studio permite a los desarrolladores crear aplicaciones, sitios y aplicaciones web, así como servicios web en cualquier entorno que soporte la plataforma .NET. Otra de las funcionalidades del entorno de desarrollo es que permite la ejecución de los programas paso a paso e incluir puntos de interrupción diversos, además de poder analizar el contenido de las variables a medida que se está ejecutando la aplicación.



1.8.5.2 MonoDevelop

MonoDevelop es un proyecto separado del proyecto Mono, pero muy unido a este. Es un entorno de desarrollo bastante completo, muy parecido al SharpDevelop y que solo funciona en entornos Unix/Linux. Permite la creación de proyectos diversos, ya sean simples o complejos, incluye plantillas y permite compilar diversos lenguajes, C#, C++, Visual Basic (38). Al igual que Visual Studio o SharpDevelop, dispone de una función de autocompletado que es útil a la hora de escribir el código, aunque no dispone de funciones de depuración como la de establecer puntos de ruptura o ejecutar paso a paso, por lo que en algunos casos encontrar un fallo específico puede ser tedioso.

Luego de haber analizado los entornos integrados de desarrollo propuestos se llegó a la conclusión que se utilizará el IDE Visual Studio 2010 ya que permite a los desarrolladores crear aplicaciones de alta calidad con gran rapidez, más seguras, confiables y administrables. Además de modelar la arquitectura y el diseño del sistema. Es un entorno de desarrollo integrado para sistemas operativos Windows. Proponiendo muchas opciones que permiten desde editar y compilar los programas hasta editar los recursos que se quieran añadir al proyecto, tales como tablas de lenguaje, iconos y formularios. También incluye avances clave para la colaboración eficiente entre los miembros del equipo.

1.8.6 Gestores de base de datos

Para la selección de la base de datos se analizaron los tres gestores de base de datos más populares en la actualidad. Primeramente se analizó Oracle que presenta 4 ediciones de base de datos pero una sola gratuita (Express Edition). Esta edición tiene grandes limitaciones respecto a los otros gestores de base de datos que se analizarán posteriormente, ya que presenta limitaciones de tamaño y de uso de la memoria RAM. Otra de las opciones que se analizó fue MySQL, el cual es un gestor de base de datos libre con características que compiten con los gestores más usados en la actualidad pero presenta un esquema de licenciamiento dual: por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para incorporarlo en productos privativos se debe comprar a la empresa una licencia específica que permita su uso. Y finalmente se analizó y seleccionó PostgreSQL con versión 9.1 (39), el cual tiene licencia BSD (Berkeley Software Distribution, por sus siglas en inglés) la cual permite el uso de este en software privativo. Además PostgreSQL incluye características de la orientación a objetos, como son los tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. Es hoy, el sistema gestor de bases de datos libre más avanzado que existe, contando con características tales como:



- Extensible: El código fuente está disponible para todos sin costo. Si su equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales. Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días.
- Multiplataforma: PostgreSQL está disponible en casi cualquier Unix (34 plataformas en la última versión estable), y una versión nativa de Windows está actualmente en estado beta de pruebas.
- Diseñado para ambientes de alto volumen: PostgreSQL está considerado como la base de datos de código abierto más avanzada del mundo. Cuenta con una intuitiva herramienta de administración con interfaz gráfica llamada pgAdmin, la cual proporciona una forma fácil para ejecutar tareas de configuración, creación de bases de datos, tablas, usuarios, etc.

1.8.7 Plataforma de desarrollo

Durante el desarrollo de la presente investigación se utilizará el Framework.net en su versión 4.0 de Microsoft, con independencia de plataforma de hardware. Permite un rápido desarrollo de aplicaciones. Ofrece una manera rápida y económica, que a la vez es segura y robusta, a la hora de desarrollar aplicaciones o soluciones como las denomina la misma plataforma. Provee un extenso conjunto de soluciones predefinidas para necesidades generales de la programación de aplicaciones, y administra la ejecución de programas escritos específicamente con la plataforma (40). En términos simples un framework es un conjunto de clases base que pueden ser reutilizadas para la construcción de un nuevo software (40). A continuación se muestran algunas características que lo identifican:

- Completamente orientado a objetos.
- Multilenguaje.
- Modelo de programación único para todo tipo de aplicaciones y dispositivos de hardware.
- Se integra fácilmente con aplicaciones desarrolladas en plataformas Microsoft o en otras plataformas.

1.8.8 Bibliotecas de clases utilizada en el sistema

Las bibliotecas contienen código y datos, que proporcionan servicios a programas independientes, es decir, pasan a formar parte de éstos. Esto permite que el código y los datos se compartan y puedan modificarse de forma modular. Las bibliotecas hacen referencias entre sí a través de un proceso conocido como enlace, que por lo general es realizado por un software denominado enlazador (41).



Dentro de las distintas bibliotecas que existen, se ha decidido trabajar con aquella que facilitará la detección del rostro en una imagen digital, esta es conocida como la **OpenCV** (41). Una de las características de OpenCV es que sólo es compatible con los lenguajes de programación C + +, C, y Python. Con el fin de usarlo en el lenguaje C #, debe utilizarse un contenedor, uno fácilmente disponible y fácil de usar, por lo que se decidió utilizar la biblioteca **Emgu cv**.

Emgu CV esencialmente proporciona una interfaz entre OpenCV y C #. Las llamadas a OpenCV se realizan a través Emgu CV, lo que hace la magia para hacer que funcione (42). Emgu CV es también bastante fácil de usar. La misma fue seleccionada ya que es una biblioteca libre, multiplataforma y brinda el algoritmo de AdBoost creado por Viola y Jones con un 99 % de efectividad en la detección de rostro como una vía para poder desarrollar esta fase en el proceso de reconocimiento facial en el sistema.

1.9 Conclusiones parciales

En el presente capítulo se plantearon conceptos fundamentales para lograr una mejor comprensión de la problemática, así como las temáticas que permiten el desarrollo de la investigación. Además se analizaron las diferentes tecnologías, herramientas y metodologías utilizadas en la actualidad definiendo las que más se adecúan al sistema a desarrollar. Se puede concluir que para dar solución a la problemática planteada es necesario desarrollar un sistema de identificación de personas basado en rasgos faciales. Para ello, la selección de un lenguaje de programación como C#, con el uso de las bibliotecas de clase que lo hace más potente como es Emgu cv. La herramienta CASE Visual Paradigm y el IDE Visual Studio, permitirán la agilización del desarrollo de la investigación. XP será de suma importancia para guiar el proceso de desarrollo del sistema respectivamente.



Capítulo II: “Análisis y Diseño”

2.1 Introducción

El objetivo que se persigue con la elaboración de este capítulo es mostrar la evolución del sistema a desarrollar, se realizará un correcto análisis y una amplia comprensión de todos los elementos que se relacionan con el tema de identificación de personas basado en rasgos faciales. Al mismo tiempo se desarrollarán las fases iniciales de la metodología de desarrollo XP, planificación y diseño, además de presentar los diferentes artefactos, los cuales serán premisas cruciales para la entrega final del sistema.

2.2 Propuesta del sistema

El sistema estará basado en una aplicación de escritorio el cual permitirá identificar a una persona mediante sus rasgos faciales, el cual será posible una vez que se realice la captura de la imagen facial del usuario a identificar o cargar la imagen desde un directorio fijo (ver **figura 2.1**). El sistema contará con una base de datos que contendrá las imágenes faciales de diferentes personas junto con el vector característico correspondiente a cada imagen almacenada.

El primer paso en el proceso de identificación de personas es la detección del rostro en la imagen capturada, utilizando el algoritmo AdaBoost de la biblioteca de clase Emgu cv, una vez que es detectado el rostro se normaliza la imagen estandarizándola y regulando ciertos factores que influyen en la misma como la rotación y la iluminación. Seguido de este primer paso se localizan los puntos principales del rostro, es decir las características faciales del individuo, recurriendo al método estadístico basado en características locales, Active Shape Model (ASM). Culminado este proceso la imagen está lista para aplicar la extracción de características con el filtro correspondiente, la transformada discreta del coseno (DCT), dando como respuesta el vector característico para dicha imagen.

Una vez que se obtenga el vector característico de la persona que se desea identificar, es comparado con la base de datos para cada uno de los vectores ya anteriormente calculado, dando como resultado la aceptación o rechazo del usuario. En caso de ser identificada la persona se mostrarán los datos identificativos de la misma como son nombre, apellidos, carné de identidad y sexo.

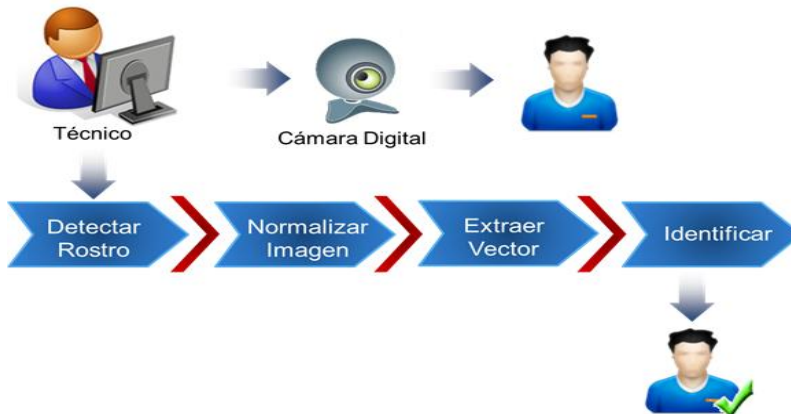


Figura 2.1: Etapas del sistema de identificación de personas basado en rasgos faciales.

2.3 Modelo de dominio

El modelo de dominio es una representación visual estática del entorno real del sistema, es decir un diagrama con los objetos que existen en el mundo real relacionados con el sistema a desarrollar y las relaciones que existen entre ellos. Su objetivo es ayudar a comprender los conceptos que utilizan los usuarios y los conceptos con los se trabajará durante el desarrollo del sistema. El diagrama de dominio se representa a través de diagramas de clases, lo que más simplificados. No son objetos de software, sino un diccionario visual de conceptos del dominio.

En la figura 2.2, se muestra el modelo de dominio que conceptualiza los elementos principales y sus relaciones.

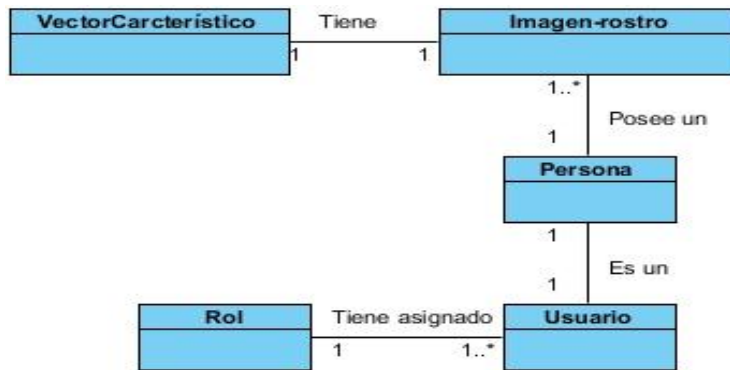


Figura 2.2: Diagrama del modelo de dominio.



2.4 Fase de planificación

La planificación es la etapa inicial de todo proyecto en XP. En este punto se comienza a interactuar con el cliente y el resto del grupo de desarrollo para descubrir los requerimientos del sistema. Se identifican el número y tamaño de las iteraciones al igual que se plantean ajustes necesarios a la metodología según las características del proyecto.

2.4.1 Actores del sistema

Las personas relacionadas con el sistema son todas aquellas que obtienen un resultado del valor de uno o varios procesos que se ejecutan. De igual forma son las que se encuentran involucradas en la aplicación pero no obtienen ningún valor a pesar de participar en ellos. En la tabla 2.1, se muestra los actores identificados en el sistema.

Actor del sistema	Descripción
Administrador	Es el encargado de gestionar los usuarios y las personas del sistema y conjuntamente puede realizar el proceso de la captura del rostro de la persona a identificar.
Técnico	Es el encargado de realizar la captura del rostro de la persona a identificar.

Tabla 2.1: Actores del sistema.

2.4.2 Captura de requisitos

La obtención de los requisitos es un paso muy importante para las siguientes etapas del desarrollo del software, pues un error en estas fases iniciales puede ocasionar que el sistema no cumpla las expectativas de los usuarios y difícilmente aporte valor agregado al negocio para el que debe ser concebido.



2.4.2.1 Requerimientos funcionales (RF)

RF1. Autenticar Usuario.

RF2. Capturar imagen con dispositivo de video.

RF3. Cargar imagen.

RF4. Guardar imagen.

RF5. Procesar imagen.

RF6. Gestionar usuarios.

RF6.1 Registrar usuario.

RF6.2 Modificar usuario.

RF6.3 Eliminar usuario.

RF6.4 Listar usuarios.

RF7. Gestionar personas.

RF7.1 Registrar persona.

RF7.2 Modificar persona.

RF7.3 Eliminar persona.

RF7.4 Listar personas enroladas.

2.4.2.2 Requerimientos no funcionales (RNF)

1. Software

- El sistema será usado bajo los sistemas operativos Windows.
- Framework de desarrollo: .NET Framework 4.0

2. Hardware

- PC Intel Pentium 4 o superior.
- CPU 2.13GHZ o superior.
- 512MB de Memoria RAM o superior.

3. Restricciones en el diseño y la implementación

- Lenguaje de programación: Visual C++ y C#.
- IDE: Visual Studio 2010.
- Para el Modelado de UML se utiliza: Visual Paradigm 8.0.
- Gestor de base de datos: PostgreSQL 9.1.



- Biblioteca de clases: Active Shape Model (ASM), Emgu CV.

4. Rendimiento

- La aplicación está concebida para un ambiente donde se le pueden integrar otros componentes, por lo que los tiempos de respuestas de identificación deben ser ágiles, así como la velocidad de procesamiento de información.

5. Requisitos de seguridad

- Disponibilidad: Sólo los usuarios que tengan permiso a las informaciones y a los dispositivos utilizados se les garantizará el acceso al sistema.

6. Apariencia e Interfaz interna

- Utilizar colores frescos y ofrecer suficiente contraste entre texto y fondo para no dificultar la lectura.
- La interfaz interna estará determinada por los desarrolladores, construyendo así una vista escalable de las clases o agrupaciones de clases que permitirán un mejor encapsulamiento de las funcionalidades y una mayor abstracción modular del sistema.

2.4.3 Historias de Usuarios (HU)

Las historias de usuario (HU) representan las funcionalidades que el cliente desea que estén presentes en el sistema; por lo tanto, todo el trabajo futuro debe girar en torno a satisfacer estas expectativas. La descripción de las historias de usuarios capturar imagen con dispositivo de video y procesar imagen se muestra a continuación. El resto de las historias de usuario se encuentran descritas en los anexos.

Historia de Usuario	
Número: 5	Usuario: Técnico, Administrador.
Nombre historia: Procesar imagen	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Alta



Puntos estimados: 3	Iteración asignada: 1
Programador responsable: Rafael Quiles Velázquez	
Descripción: Se realiza el pre-procesado y procesado de la imagen mediante el algoritmo propuesto para la identificación de la persona.	
Observaciones: En caso de no detectarse ningún rostro en la imagen no se podrá realizar el proceso de identificación	

Tabla 2.2: Descripción de la HU. Procesar imagen.

2.4.4 Plan de iteraciones

Después de ser identificadas y descritas las HU, se procede a la planificación de la fase de implementación estableciendo una división de dos iteraciones.

- **Iteración 1:** En esta iteración se realizarán las historias de usuarios más complejas y que inciden críticamente en la lógica de la aplicación. Una vez concluida esta iteración el sistema se encontrará en un estado plenamente funcional.
- **Iteración 2:** En esta iteración se realizarán las historias de usuarios que son menos complejas y que no inciden críticamente en la lógica de la aplicación. Una vez concluida esta iteración la aplicación se encontrará completamente concluido y listo para su explotación.

2.4.4.1 Plan de duración de las iteraciones

El plan de duración de las iteraciones se encarga de mostrar las HU en el orden en que se implementarán en cada iteración así como la duración estimada de las mismas. La siguiente tabla muestra la estimación de esfuerzo por cada historia de usuario.

No	Historias de usuario	Estimación (Semanas)
1	Autenticar usuario	1
2	Capturar imagen	1
3	Cargar imagen	1



4	Guardar imagen	1
5	Procesar imagen	3
6	Gestionar usuario	2
7	Gestionar persona	2

Tabla 2.3: Estimación de esfuerzo por Historias de Usuario.

La tabla 2.4 muestra el plan de duración de las iteraciones que permite el desarrollo del sistema:

Iteración	Código	Historias de usuarios	Duración estimada (semanas)
Iteración 1	HU2	Capturar imagen	6
	HU3	Cargar imagen	
	HU4	Guardar imagen	
	HU5	Procesar imagen	
Iteración 2	HU1	Autenticar usuario	5
	HU6	Gestionar usuario	
	HU7	Gestionar persona	

Tabla 2.4: Plan de duración de las iteraciones.

- La Iteración 1 durará en total 1 mes y dos semanas.
- La Iteración 2 durará en total 1 mes y una semana.

2.4.5 Plan de entregas.

Una vez que se concluye la tarea por parte del cliente de elaborar las distintas HU, se comienza con la creación del plan de entrega, para estimar el tiempo de desarrollo de las mismas. Este artefacto se elabora con la intención de fijar qué período de tiempo puede tardar la implementación de cada una de las historias en cada iteración, definiéndose las fechas en que serán liberadas las versiones funcionales del producto.



- Plan de entrega del sistema de identificación de personas es:

1ra Entrega(Iteración 1)	2da Entrega(Iteración 2)
20 de marzo de 2013	27 de abril de 2013

Tabla 2.5: Plan de entrega.

2.5 Fase de diseño

Siguiendo las guías de la metodología XP, debe aparecer un diseño que describa qué clases hay y cómo interactúan. La metodología XP no requiere la representación del sistema mediante diagramas de clases utilizando el Lenguaje Unificado de Modelado (UML); en su lugar usa una técnica para representar clases, esta es la que usa tarjeta Clase-Responsabilidad-Colaboración (CRC). No obstante, el uso de diagramas UML puede aplicarse siempre y cuando influyan en el mejoramiento de la comunicación, es por esta razón se decide diseñar el diagrama de clases del diseño para un mejor entendimiento del sistema a desarrollar.

2.5.1 Tarjetas CRC

La tabla 2.6 muestra la tarjeta CRC de la clase ReconocimientoFacialASM perteneciente al paquete de Algoritmo de Reconocimiento Facial, citar que las clases que no se describen en el epígrafe se encuentran en los anexos.

Nombre de la Clase: ReconocimientoFacialASM	
Responsabilidades	Colaboradores
<ol style="list-style-type: none"> 1. ObtenerVectorCaracteristico 2. CompararVectorCaracteristicoDTC 	<ol style="list-style-type: none"> 1. ModeloFormaActiva 2. ProcesamientoImagen 3. DLLConsum 4. EDVector

Tabla 2.6: Tarjeta CRC correspondiente a la clase ReconocimientoFacialASM.

2.5.2 Diagramas de clases del diseño

Para un mejor entendimiento del funcionamiento y relación entre los archivos del sistema desarrollado se diseñan los diagramas de clases del diseño.



Para tener una mejor organización de las clases que conforman el sistema de identificación de personas basado en rasgos faciales, se decidió diseñar el siguiente diagrama de paquetes (ver **figura 2.3**), el cual está compuesto por dos sub-paquetes, el primero es gestión de usuarios, donde en él se encuentran las clases que permiten el manejo con la gestión de las personas, usuarios del sistema (ver **figura 2.4**) y el segundo es nombrado algoritmo de reconocimiento facial que sería el corazón del sistema, ya que es aquí donde se encuentran con las clases que rigen el proceso de identificación de personas basado en rasgos faciales (ver **figura 2.5**).

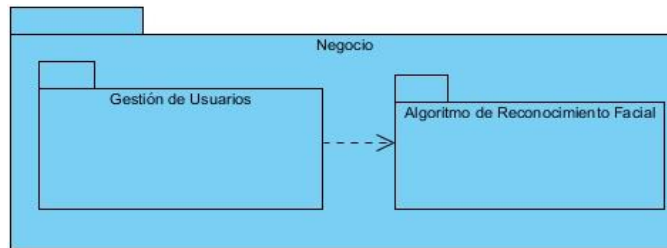


Figura 2.3: Diagrama de paquetes del sistema.

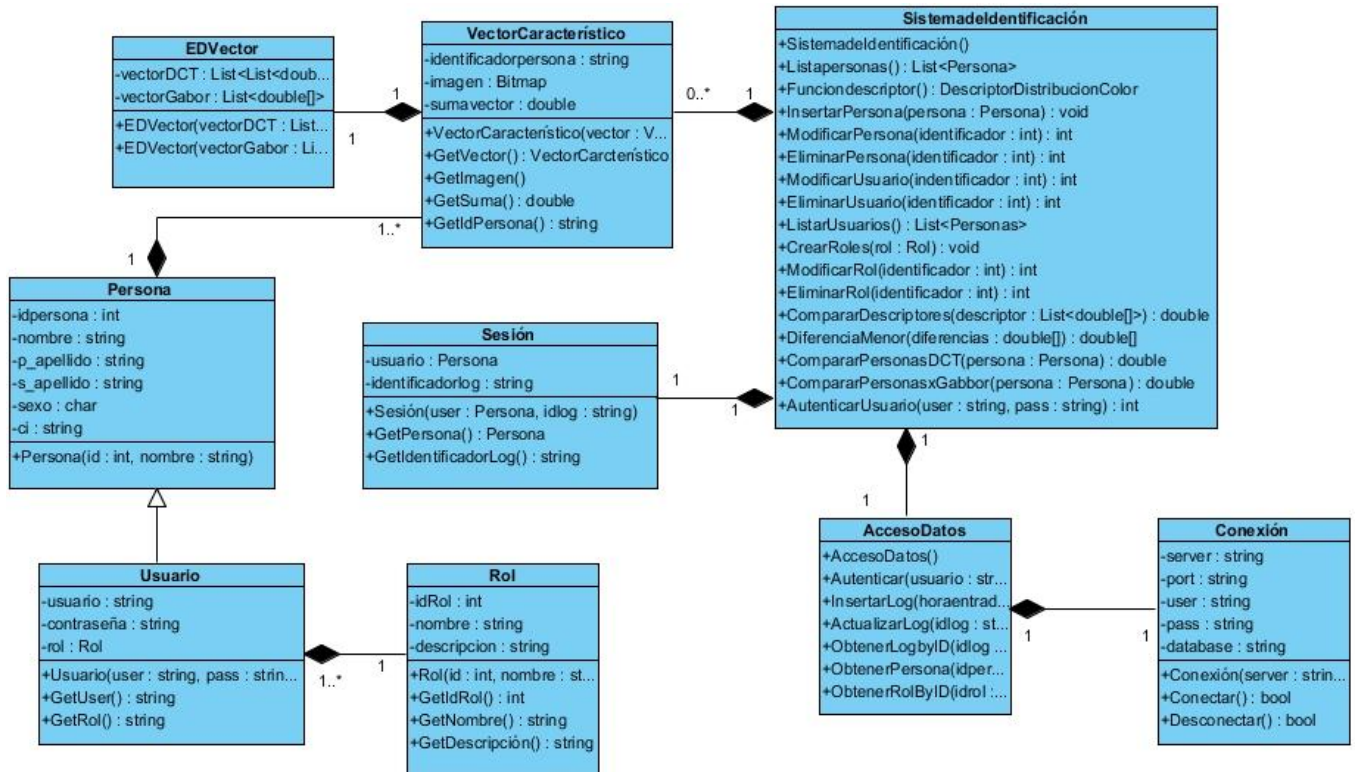


Figura 2.4: Diagrama de clases perteneciente al sub-paquete de Gestión de usuarios.

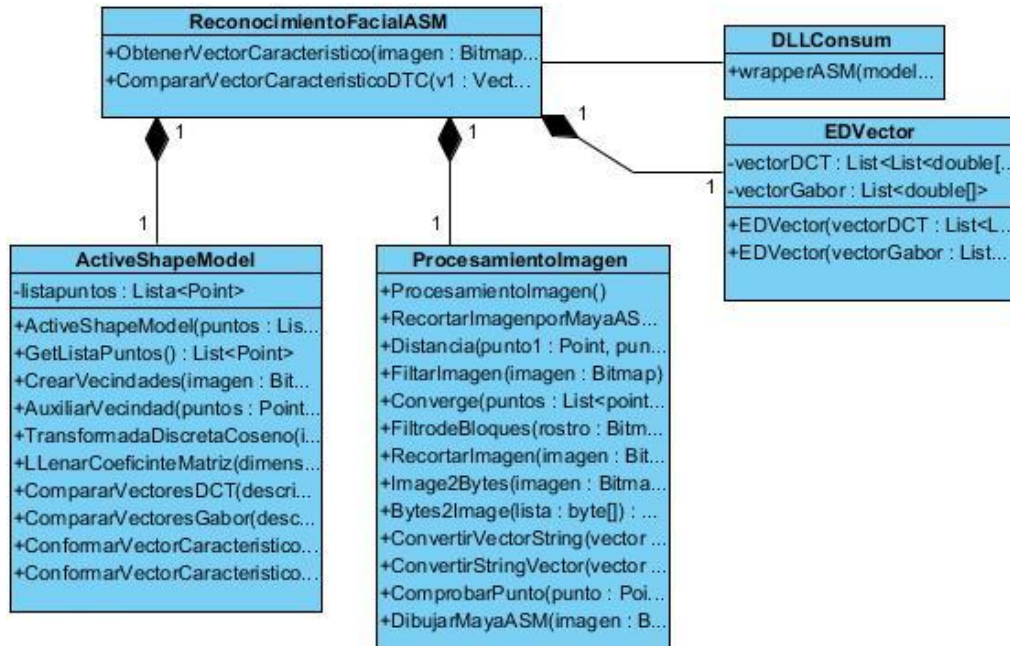


Figura 2.5: Diagrama de clases perteneciente al sub-paquete de Algoritmo de reconocimiento facial.

2.5.3 Modelo de datos del sistema

El modelo de datos del sistema de identificación de personas basado en rasgos faciales (figura 2.6) está diseñado por diferentes tablas como son:

1. Persona: Guarda los datos personales de los usuarios del sistema.
2. Log: Se registran los usuarios autenticados en el sistema.
3. Usuario: Guarda los datos de los usuarios del sistema incluyendo el rol que se corresponde.
4. Vector característico: De esta tabla se puede obtener en vector característico para cada persona del sistema con la imagen que lo identifica.
5. Rol: Se registran los tipos de roles y la descripción de cada uno de ellos.

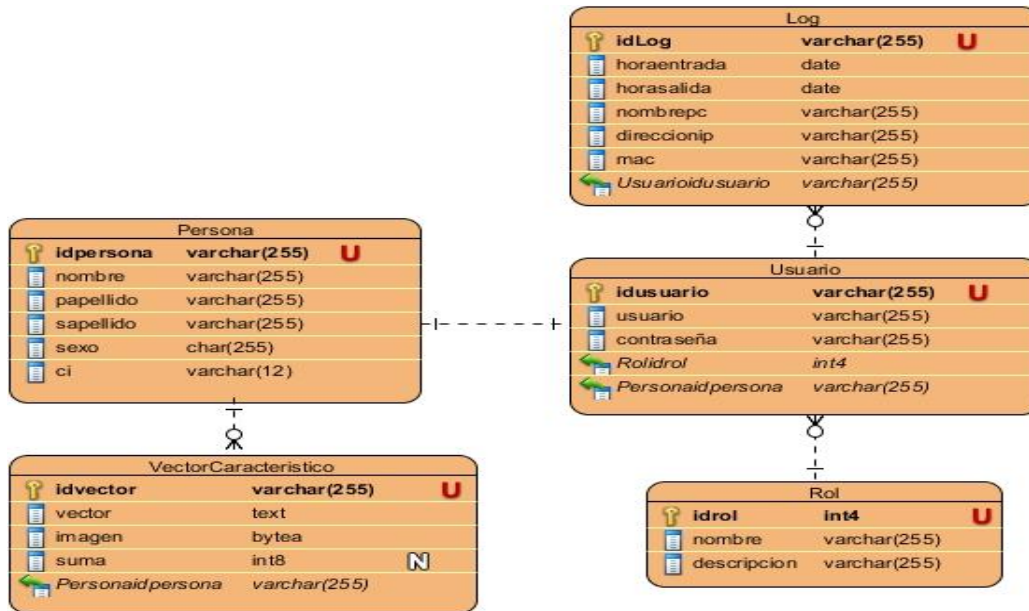


Figura 2.6: Diagrama entidad-relación.

2.5.4 Arquitectura del sistema

La arquitectura de software es, a grandes rasgos, una vista del sistema que incluye los principales componentes del mismo, la conducta de esos componentes según se percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. Establece los fundamentos para que analistas, diseñadores, programadores y otros roles, trabajen en una línea común que permita alcanzar los objetivos del sistema, cubriendo todas las necesidades (38).

Las arquitecturas y estilos arquitectónicos más universales son (31):

- Cliente-servidor: Esta arquitectura se divide en dos partes claramente diferenciadas, la primera es la parte del servidor y la segunda la de un conjunto de clientes. Normalmente el servidor es una máquina bastante potente que actúa de depósito de datos y funciona como un sistema gestor de base de datos (SGBD).
- Arquitectura n-capas: Se basa en una distribución jerárquica de roles y responsabilidades para proporcionar una división efectiva de los problemas a resolver. Los roles indican el tipo y la forma de interacción con otras capas, y las responsabilidades la funcionalidad que implementan.
- Tuberías y Filtros: Interacción sucesiva, transformación de flujos de datos. Los datos llegan a un filtro, se transforman y son pasados a través de tubos al siguiente filtro. Es apropiada para sistemas que implementan transformaciones de datos en pasos sucesivos. Un patrón tubería y



filtro tiene un grupo de componentes llamados filtros, conectados por tuberías que transmiten datos de un componente al siguiente. El filtro está diseñado para recibir entrada de datos de una forma y producir la salida de datos de una forma específica.

- Arquitectura Orientada a Servicios (SOA): Establece un marco de diseño para la integración de aplicaciones independientes de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios. La forma más habitual de implementarla es mediante Servicios Web, una tecnología basada en estándares e independiente de la plataforma, con la que SOA puede descomponer aplicaciones monolíticas en un conjunto de servicios e implementar esta funcionalidad en forma modular.
- Modelo Vista Controlador (MVC): Es un patrón de diseño de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones Web, donde la vista es la página HTML y el código que provee de datos dinámicos a la página.

Se utilizó la arquitectura n-capas (ver **figura 2.7**) como línea base para la organización estructural de la aplicación del sistema de reconocimiento facial. El sistema se divide en las siguientes capas:

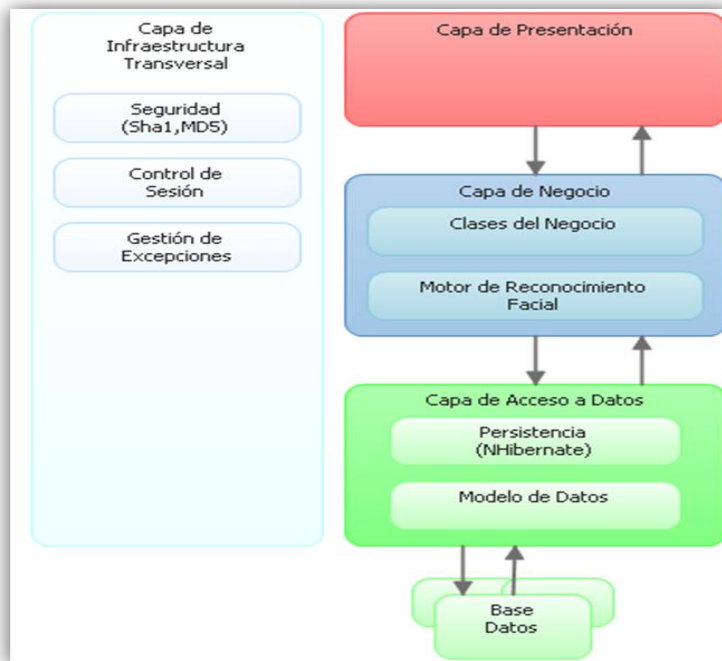


Figura 2.7: Arquitectura del sistema de identificación de personas.



- Capa Presentación: Es la que ve el usuario, presenta el sistema al mismo, le comunica la información y captura los datos que el usuario introduce en la aplicación. Esta capa se comunica únicamente con la del negocio enviando peticiones a la misma. También es conocida como interfaz gráfica y tiene la característica de ser entendible y fácil de usar para el usuario.
- Capa del Negocio: Es donde residen los programas que se ejecutan durante la ejecución de la aplicación. Esta capa se comunica con la de presentación, para recibir las peticiones y presentar los resultados, y con la de acceso a datos, para interactuar con la base de datos, consultando y modificando sus informaciones. En la misma se encuentran las clases del negocio que no son más que aquellas que permiten la gestión de usuarios y de personas, sin olvidar las clases que conforman el motor de reconocimiento facial, que son las encargadas del realizar todo el proceso de identificación de la persona, como el procesamiento de la imagen y la extracción de características del rostro.
- Capa de Acceso a Datos: Es donde residen los datos y la encargada de acceder y modificar a los mismos. Está formada por uno o más sistemas de gestión de bases de datos que realizan todo el almacenamiento de informaciones y que reciben solicitudes de almacenamiento o recuperación de datos desde la capa de negocio.
- Capa de infraestructura transversal: Esta capa encierra las operaciones de autenticación, gestión de excepciones, registros (logging), trazas y validaciones de datos entrada.

El estilo arquitectónico propuesto es Tuberías y Filtros para la infraestructura de comunicación, ya que provee una estructura para procesar flujos de datos. El sistema se percibe como una sucesión de transformaciones que sufre una serie de datos de entrada. Los datos ingresan al sistema y fluyen a través de los componentes uno a uno hasta que se asignan a un destino final: salida o almacenamiento.

El sistema se divide en varios pasos secuenciales de procesamiento; los pasos se conectan con flujos de datos; cada filtro consume y procesa sus datos en forma incremental; el destino y los filtros se conectan con tubos que implementan el flujo de datos entre los pasos de procesamiento, estos pasos están dados en (ver **figura.2.8**):

1. Entrada: Imagen del rostro capturada por medio de una cámara digital.
2. Filtro #1: Detección del rostro en la imagen de entrada.
3. Filtro #2: Segmentación y normalización del rostro.
4. Filtro #3: Localización de los puntos característicos.



5. Salida: Rostro procesado.

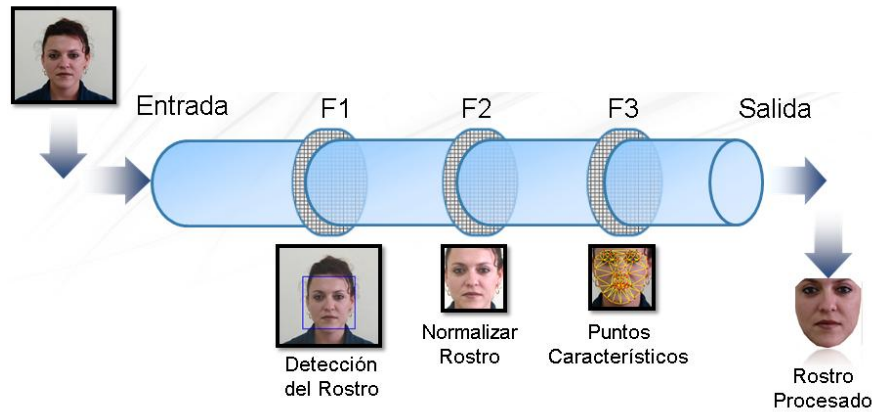


Figura 2.8: Estilo arquitectónico tuberías y filtros del sistema.

2.5.5 Patrones de diseño utilizados

2.5.5.1 Patrones GRASP

En el diseño de la aplicación se utilizaron los patrones GRASP (General Responsibility Assignment Software Patterns) que significa patrones generales de software para asignar responsabilidades. Describen los principios fundamentales de la asignación de responsabilidades a objetos, expresados en forma de patrones (38). Dentro de los patrones de diseño GRASP se utilizó:

- **Experto en información:** Se utiliza porque conserva el encapsulamiento, pues los objetos se valen de su propia información para realizar la tarea encomendada. Esto soporta un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento. Además, el comportamiento se distribuye entre las clases que cuentan con la información requerida, haciéndolas más fáciles de comprender y de mantener, garantizando una alta cohesión. Este patrón se utiliza en todas las clases del sistema de identificación de personas.
- **Creador:** El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos. El patrón se pone de manifiesto en la clase de reconocimiento facial a la hora de crear varias instancias de clases que relacionan con ella.
- **Alta Cohesión:** Asignar una responsabilidad de modo que la cohesión siga siendo alta. Una clase con baja cohesión hace muchas cosas no afines o un trabajo excesivo. Una clase con mucha



cohesión es útil porque es fácil darle mantenimiento, entenderla y reutilizarla. En el sistema de identificación propuesto se encuentra la clase Persona, Usuario y Rol.

- **Bajo Acoplamiento:** Asignar una responsabilidad para mantener bajo acoplamiento. Las clases deben comunicarse con un número pequeño de clases tanto como sea posible. Ejemplo de las clases con bajo acoplamiento se encuentran, clase ProcesarImagen y ActiveShapeModel.
- **Controlador:** El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa. Ejemplo de este patrón se encuentra la clase controladora del sistema SistemadIdentificacion.

2.5.5.2 Patrones GOF

Los patrones GOF que se utilizaron en el desarrollo del sistema de identificación de personas basado en rasgos faciales son:

- **Patrones de Creación:** El objetivo de estos patrones es de abstraer el proceso de instanciación y ocultar los detalles de cómo los objetos son creados o inicializados.
 1. **Prototype (Prototipo):** permite crear nuevos objetos clonándolos de una instancia ya existente, copiando el prototipo de esta clase.
Ejemplo:
`SistemadIdentificacionPersonas sistema;`
`this.sistema = sistema`
 2. **Singleton (Instancia única):** Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia.
Ejemplo:
`sistema = SistemadIdentificacionPersonas ObtenerInstancia();`
`public static SistemadIdentificacionPersonas ObtenerInstancia()`
`{ if (instancia == null)`
`instancia = new SistemadIdentificacionPersonas();`
`return instancia;`
`}`
- **Patrones de Estructurales:** Los patrones estructurales describen como las clases y objetos pueden ser combinados para formar grandes estructuras y proporcionar nuevas funcionalidades.



1. Facade (Fachada): Provee de una interfaz unificada simple para acceder a una interfaz o grupo de interfaces de un subsistema.

Ejemplo:

```
public interface IReconocimientoFacial
{ActiveShapeModel ASM1();
  ProcesaminetoImagen Procesamiento();
  EDVector  ObtenerVectorCaracteristico(Bitmap imagen, int proc, string dir ,bool
  ecularhistograma);
  double CompararVectorCaracteristicoDTC(EDVector v1, EDVector v2, int procedimiento);
}
```

2.6 Conclusiones parciales

Luego de definir las características que debe cumplir el Sistema de identificación de personas basado en rasgos faciales y de realizar el análisis y diseño correspondiente, se concluye que durante el desarrollo del capítulo se precisaron las diferentes historias de usuario definiendo así un plan de duración y un plan de entrega para la realización del sistema, permitiendo que el mismo cumpla con las funcionalidades requeridas y con el tiempo estimado. Además se seleccionó la arquitectura n-capas que garantizará el desarrollo de la aplicación con el uso apropiado de los patrones de diseño para lograr un software con calidad.

Capítulo III: “Implementación y Pruebas”

3.1 Introducción

Luego de realizar el diseño del sistema propuesto y de definir su arquitectura, se procede a implementar un software que cumpla con los requisitos que se necesitan. El presente capítulo está dedicado a detallar las dos iteraciones llevadas a cabo durante la etapa de construcción de la aplicación, mostrando los principales artefactos de la implementación del mismo. Además se realizarán pruebas de funcionalidad y efectividad al sistema implementado.



3.2 Estándar de codificación

Es necesario establecer un criterio fijo que proporcione reglas para la creación de nombre para variables y métodos, permitiendo una mejor lectura del software, y un mejor entendimiento del código más rápidamente. A continuación se describe el estándar utilizado en la implementación de la solución.

Reglas de codificación

1. El código fuente debe ser escrito en español.
2. Se debe evitar las líneas de más de 80 caracteres, ya que no son bien manejadas por muchas herramientas.
3. Cada línea debe contener cuando más una sentencia.
4. Cada funcionalidad debe tener comentario de su funcionamiento.
5. Se debe seguir las convenciones de nombre mostradas en la **tabla 3.1**.

Tipos de identificadores	Reglas de nombre	Ejemplos
Clases o Interfaces.	Los nombres de las clases o interfaces debe tener la primera letra de cada palabra en mayúscula.	Persona, Sistemaldentificacion
Métodos.	Los nombres de los métodos deben reflejar la acción a realizar y siempre comenzando con letra mayúscula, en caso de ser compuesto ambas palabras en Mayúscula.	DiferenciaMenor, InsertarPersona, ModificarPersona
Variables.	Todas las variables empezarán con minúscula y la primera letra de las siguientes palabras en minúscula.	imagenpersonal, identificador.
Constantes.	Cada carácter que pertenezca al nombre de la constante se escribirá en mayúscula y en caso de ser un nombre compuesto, cada palabra se separará por un guión bajo “_”.	PI, UMBRAL_ASM

Tabla 3.1: Convenciones de nombre.

3.3 Tareas de ingeniería

XP plantea que la implementación de un software se hace iterativamente, obteniendo al culminar cada iteración un producto funcional, que debe ser probado y mostrado al cliente. Durante el transcurso de las iteraciones, se realiza la implementación de las HU definidas por el cliente y descritas por el equipo de



desarrollo en la etapa de planificación. Como parte de este plan, se descomponen estas HU en tareas de la ingeniería las cuales son asignadas a los programadores para ser implementadas durante la iteración correspondiente. En la tabla 3.2 se describen las tareas correspondientes a las historias de usuario en la primera iteración del proceso de desarrollo del sistema de identificación. Las tareas de la segunda iteración se encuentran en los anexos tabla 4.13.

Iteración 1	
Historias de Usuario	Tareas
Capturar imagen	<ol style="list-style-type: none"> 1. Verificar que existe dispositivo de video conectado 2. Obtener la imagen para el procesado
Cargar imagen	<ol style="list-style-type: none"> 1. Buscar directorio donde se encuentra la imagen
Guardar imagen	<ol style="list-style-type: none"> 1. Especificar directorio donde desea guardar la imagen capturada
Procesar imagen	<ol style="list-style-type: none"> 1. Detectar el rostro en la imagen obtenida. 2. Realizar el pre-procesado de la imagen captura. 3. Ubicación de los puntos característicos en el rostro de la persona a identificar. 4. Extracción del vector característico. 5. Comparación con la base de datos.

Tabla 3.2: Tareas de ingeniería en la primera iteración.

3.4 Diagrama de componentes

Para un mejor entendimiento del funcionamiento y relación entre los archivos del sistema desarrollado se realiza el diagrama de componentes. El mismo ilustra los componentes de software que se usarán para



construir el sistema. Además se muestra la relación entre ellos, sus dependencias, su comunicación, su ubicación y otras condiciones. En la **figura 3.1** se muestra el diagrama de componentes de la aplicación y en la **tabla 3.2** se define el propósito de cada componente.

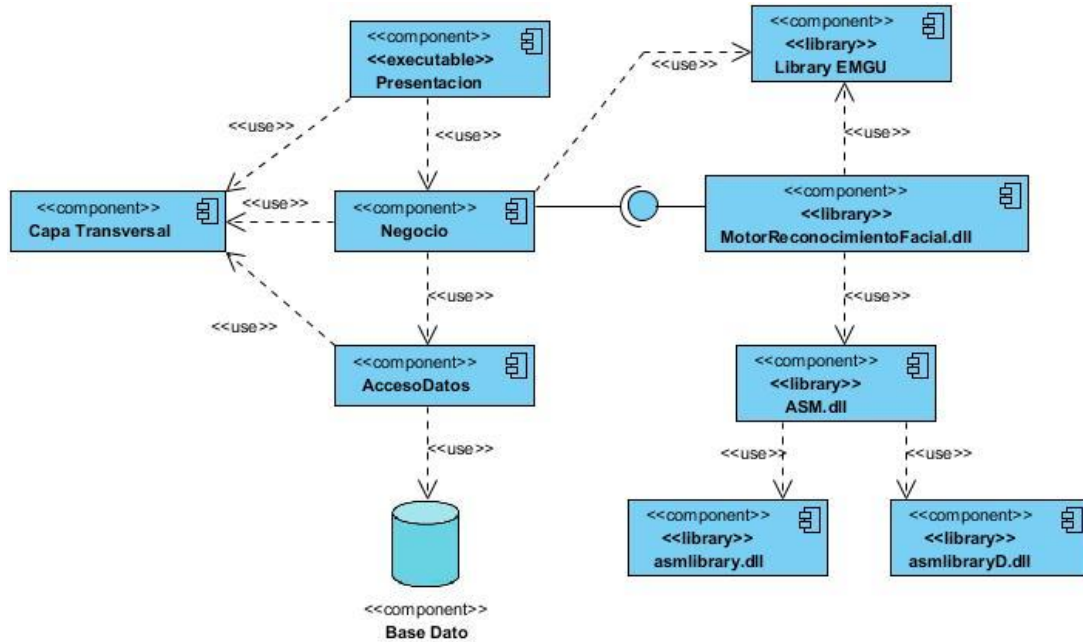


Figura 3.1: Diagrama de componentes del Sistema de identificación.

✓ **Descripción de componentes:**

Componente	Propósito
Capa Transversal	Componente que contiene operaciones de autenticación, gestión de excepciones, registros (logging), trazas y validaciones de datos entrada.
Presentación	Componente que contiene la información y realiza la captura de los datos que el usuario introduce en la aplicación y además se comunica con el componente de negocio enviando peticiones a la misma.
Negocio	Componente que se comunica con la de presentación, para recibir las peticiones y presentar los resultados y con el componente AccesoDatos, consultando y verificando sus informaciones.



AccesoDatos	Componente encargo de acceder y modificar los datos del sistema.
Library EMGU	Componente encargado de la detección del rostro y procesamiento de imágenes.
MotorReconocimientoFacial	Componente en el que se encuentran las funcionalidades principales del software encargadas de realizar todo el proceso de identificación de la persona, como el procesamiento de la imagen y la extracción de características del rostro.
ASM	Componente responsable de la localización de los puntos característicos en el rostro.
asmlibrary	Componente responsable del procesamiento de la imagen para la localización de los puntos característicos en el rostro.
asmlibraryD	Componente responsable del procesamiento de la imagen para la localización de los puntos característicos en el rostro.

Tabla 3.3: Descripción de los componentes del sistema de identificación.

3.5 Diagrama de despliegue

La siguiente figura muestra el diagrama de despliegue correspondiente al sistema de identificación a desarrollar. Este está compuesto por una cámara digital mediante la cual será posible realizar la captura del rostro de la persona que será identificada en el sistema. La cámara se conectará a través del protocolo USB con la computadora que tendrá instalado el sistema y mediante el protocolo TCP se realizará el intercambio de información con el servidor de base de datos.

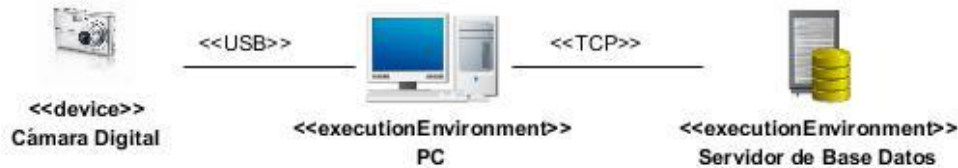


Figura 3.2: Diagrama de despliegue del Sistema de identificación.

3.6 Pruebas

Las pruebas de software consisten en la dinámica de la verificación del comportamiento de un programa en un conjunto finito de casos de prueba, debidamente seleccionados de por lo general infinitas



ejecuciones de dominio, contra la del comportamiento esperado. Son una serie de actividades que se realizan con el propósito de encontrar los posibles fallos de implementación, calidad o usabilidad de un programa u ordenador; probando el comportamiento del mismo.

Con el fin de validar el correcto funcionamiento del sistema de identificación de personas basado en rasgos faciales se realizarán pruebas de rendimiento, caja blanca y caja negra. A continuación se describe el proceso necesario para llevar a cabo dichas pruebas.

3.6.1 Pruebas de efectividad

3.6.1.1 Evaluación del rendimiento del sistema

Para evaluar las prestaciones o rendimiento del sistema, se deben analizar y valorar los siguientes parámetros estándares:

- **FAR** (False Acceptances Rate o falsos positivos): Se produce cuando el sistema acepta a un usuario no autorizado y le facilita el acceso. Se cuantifica mediante la probabilidad (o tanto por ciento) de falsas aceptaciones. Es un error crítico para la seguridad.
- **FRR** (False Reject Rate o falsos negativos): Se produce cuando el sistema rechaza a un usuario autorizado. Se cuantifica mediante la probabilidad (o tanto por ciento) de falsos rechazos. Es un error molesto para los usuarios, pero no crítico para la seguridad.
- **EER** (Equal Error Rate): los indicadores FAR y FRR son parámetros inversamente proporcionales, y variarán en función de las condiciones prefijadas por el programa de identificación biométrica. Si el algoritmo se va a utilizar en un sistema de seguridad, se ha de intentar que el FAR sea lo más pequeño posible, aunque esto signifique el incremento del FRR. Se debe fijar un umbral que permita igualar estos dos factores, momento en el que se considera óptimo el funcionamiento del sistema. Este umbral se denomina Equal Error Rate (**Figura 3.3**) y es el que determina el poder de identificación del sistema.



Figura 3.3: Determinación del umbral Equal Error Rate (30).



La FRR es una función estrictamente creciente y la FAR una estrictamente decreciente. La FAR y la FRR al ser modeladas como función del umbral de aceptación tienen por dominio al intervalo real $[0,1]$, que es además su recorrido, puesto que representan frecuencias relativas. En la figura que se mostró anteriormente puede apreciarse el umbral de aceptación, donde la FRR y la FAR toman el mismo valor y puede ser utilizado como medida única para caracterizar el grado de seguridad del sistema. Usualmente se elige un umbral de aceptación por debajo del punto donde se igualan estos dos factores con el objetivo de reducir la FAR y aumentar los FRR.

En la figura 3.4 se puede observar la prueba realizada al sistema de reconocimiento facial para determinar el umbral correspondiente.

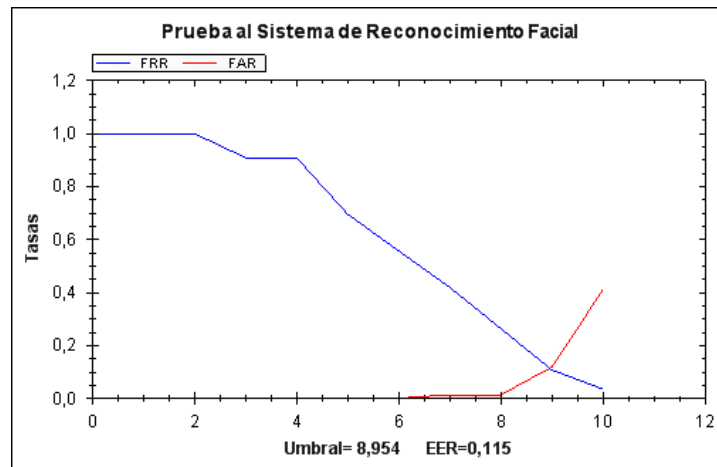


Figura 3.4: Determinación del umbral de aceptación para el sistema de identificación de personas.

Luego de analizar los resultados del experimento se concluye que para un umbral de 0,88 la efectividad es máxima.

La siguiente tabla muestra las pruebas realizadas al sistema de identificación que indican el tiempo que demora el proceso de comparación sin involucrar la extracción de características, mostrando en la **figura 3.5** la gráfica que relacionan dichas pruebas.

Prueba 1		
Total Imágenes en la base de datos	Tiempo en Segundo	Tiempo en Milisegundo
1	0	0
10	0,001	1
100	0,012	12



1000	0,12	120
Prueba 2		
Total Imágenes en la base de datos	Tiempo en Segundo	Tiempo en Milisegundo
1	0	0
10	0,001	1
100	0,011	11
1000	0,11	110
Prueba 3		
Total Imágenes en la base de datos	Tiempo en Segundo	Tiempo en Milisegundo
1	0	0
10	0,001	1
100	0,011	11
1000	0,12	120
Prueba 4		
Total Imágenes en la base de datos	Tiempo en Segundo	Tiempo en Milisegundo
1	0	0
10	0,001	1
100	0,011	11
1000	0,11	110

Tabla 3.4: Tiempo de comparaciones de imágenes faciales sin el tiempo de extracción de características.

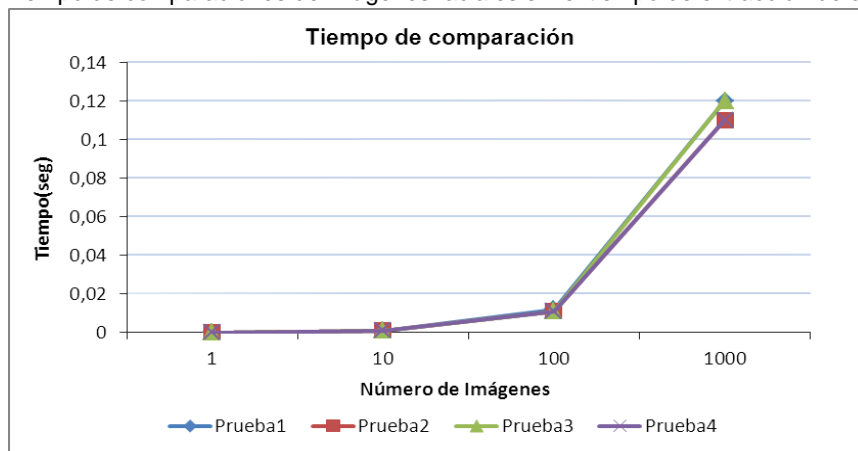


Figura 3.5: Representación de tiempo de comparación.



De igual forma se decidió calcular el tiempo promedio en identificar una persona, este experimento se realizó en una computadora Intel Core i3 a 3GHz (4 CPUz), 2GB de Memoria RAM DDR3, con Sistema Operativo Windows 7 Ultimate. Los resultados se muestran en la **figura 3.6**, del cual se concluye que el tiempo promedio de aceptación o rechazo de una persona es aproximadamente 1,28 segundos.

Prueba 1		
Total Imágenes en la base de datos	Tiempo en Segundo	Tiempo en Milisegundo
1	1,335	1335
10	1,246	1246
100	1,361	1361
1000	1,208	1208
Prueba 2		
Total Imágenes en la base de datos	Tiempo en Segundo	Tiempo en Milisegundo
1	1,289	1289
10	1,249	1249
100	1,204	1204
1000	1,232	1232
Prueba 3		
Total Imágenes en la base de datos	Tiempo en Segundo	Tiempo en Milisegundo
1	1,241	1241
10	1,05	1050
100	1,453	1453
1000	1,243	1243
Prueba 4		
Total Imágenes en la base de datos	Tiempo en Segundo	Tiempo en Milisegundo
1	1,117	1117
10	1,434	1434
100	1,418	1418
1000	1,421	1421
Prueba 5		
Total Imágenes	Tiempo en Segundo	Tiempo en Milisegundo
1	1,454	1454
10	1,06	1060
100	1,072	1072



1000	1,262	1262
------	-------	------

Tabla 3.5: Tiempos de extracción de características conjuntamente con la comparación con la base de datos.

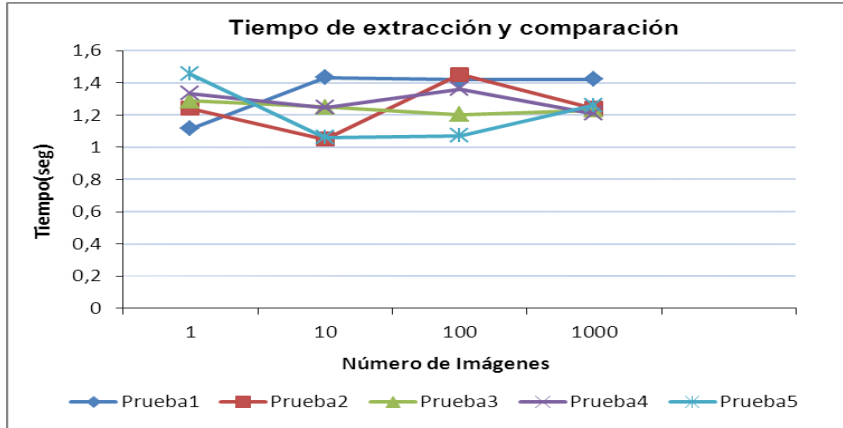


Figura 3.6: Representación de los tiempos de identificación de una persona.

La tabla 3.6 indica el tiempo de extracción de vectores, revelando que este tiempo aumenta a medida que se adicionan más número de imágenes (ver **figura 3.7**)

Prueba 1		
Total Imágenes	Tiempo en Segundo	Tiempo en Milisegundo
1	0,887	887
10	9,883	9883
50	50,583	50583
100	90,848	90848
Prueba 2		
Total Imágenes	Tiempo en Segundo	Tiempo en Milisegundo
1	1,06	1060
10	10,652	10652
50	39,801	39801
100	87,344	87344
Prueba3		
Total Imágenes	Tiempo en Segundo	Tiempo en Milisegundo
1	1,189	1189
10	9,425	9425
50	47,729	47729
100	91,551	91551



Prueba4		
Total Imágenes	Tiempo en Segundo	Tiempo en Milisegundo
1	1,078	1078
10	9,797	9797
50	53,961	53961
100	89,378	89378

Tabla 3.6: Tiempo de Extracción de Vectores.

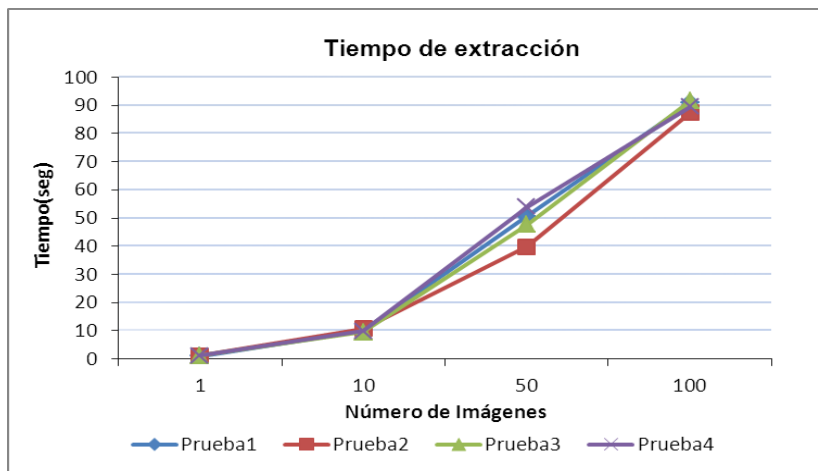


Figura 3.7: Representación de los tiempos de extracción de los vectores característicos.

3.6.2 Pruebas de funcionalidad

Las pruebas de funcionalidad se enfocan en las acciones por parte del usuario y las respuestas por parte del sistema, se llevan a cabo para comprobar los requerimientos y se considera que una funcionalidad tiene éxito cuando se comporta de la manera esperada por el cliente. Para comprobar el cumplimiento de las funcionalidades del sistema se diseñaron casos de prueba para cada HU y a continuación se relacionan las más importantes.

3.6.2.1 Pruebas de caja negra

La prueba de caja negra se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. O sea, a través de los casos de prueba se demuestra que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene. Se realizaron pruebas a diferentes funcionalidades del sistema una de ellas se muestra a continuación y las restantes se pueden observar en los anexos 4.4.



Casos de Prueba de Funcionalidad	
Código Caso de Prueba: 1	Nombre Historia de Usuario: Procesar imagen
Nombre de la persona que realiza la prueba: Amelia Aguilera Reyes	
Descripción de la prueba: Dado la captura de la imagen de un rostro se realizarán los procesos, detección del rostro en la imagen, normalización de la imagen, extracción de características y la comparación con la base de datos que permitirá la identificación de la persona.	
Condiciones de ejecución: La imagen que se utilice debe poseer buenas condiciones de iluminación y la resolución debe ser mayor de 250x250, para obtener resultados satisfactorios. Además para poder identificar una persona, esta debe estar en la base de datos.	
Entrada/Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Imagen capturada por un dispositivo de video. 	
Resultado Esperado: Devuelve los datos identificativos de la persona, nombre, apellidos, sexo, carné de identidad y la diferencia entre los vectores, conjuntamente con la imagen por la cual fue identificada en la base de datos.	
Evaluación: Satisfactoria	

Tabla 3.7: Caso de prueba de la funcionalidad: Procesar imagen.

3.6.2.2 Pruebas de caja blanca

La prueba de caja blanca se basa en el diseño de casos de prueba que usa la estructura de control del diseño procedimental para derivarlos. Mediante la prueba de la caja blanca el ingeniero del software puede obtener casos de prueba que (31):

1. Garanticen que se ejerciten por lo menos una vez todos los caminos independientes de cada módulo, programa o método.
2. Ejerciten todas las decisiones lógicas en las vertientes verdadera y falsa.
3. Ejecuten todos los bucles en sus límites operacionales.
4. Ejerciten las estructuras internas de datos para asegurar su validez.



Es por ello que se considera a la prueba de Caja Blanca como uno de los tipos de pruebas más importantes que se le aplican a los software, logrando como resultado que disminuya en un gran porcentaje el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad.

- **Prueba del camino básico.**

La prueba del camino básico es una técnica de prueba de la Caja Blanca propuesta por Tom McCabe. Esta técnica permite obtener una medida de la complejidad lógica de un diseño y usa esta medida como guía para la definición de un conjunto básico. La idea es derivar casos de prueba a partir de un conjunto dado de caminos independientes por los cuales puede circular el flujo de control. Para obtener dicho conjunto de caminos independientes se construye el Grafo de Flujo asociado y se calcula su complejidad ciclomática. Los pasos que se siguen para aplicar esta técnica son:

1. A partir del diseño o del código fuente, se dibuja el grafo de flujo asociado.
2. Se calcula la complejidad ciclomática del grafo.
3. Se determina un conjunto básico de caminos independientes.
4. Se preparan los casos de prueba que obliguen a la ejecución de cada camino del conjunto básico.

Tres formas fundamentales de calcular la complejidad:

1. El número de regiones del grafo de flujo coincide con la complejidad ciclomática. La complejidad ciclomática, $V(G)$, se define como:

- $V(G) = A - N + 2$

Dónde: A es el número de aristas del grafo y N es el número de nodos.

2. La complejidad ciclomática, $V(G)$, también se define como:

- $V(G) = P + 1$

Dónde: P es el número de nodos predicado contenido en el grafo G.

3. Números de regiones del grafo.

- $V(G) = R.$

Dónde: R es el número de regiones.

Realizar prueba de caja blanca a la funcionalidad obtener vector característico y comparar vector característicos.

- ✓ **Funcionalidad: ObtenerVectorCaracteristico.**

```
public EDVector ObtenerVectorCaracteristico(Bitmap original,int proc, string path,bool ecualizarhist)
{
```



//Paso 1: Detectar Rostro y ubicar puntos característicos.

//Crear Active Shape Model.

```
procesamiento.GuardarImagen(path + "\\imagen.jpg", original, 255); //1
```

```
puntosoriginal=DLLConsum.wrapperASMsinImagen("my68-1d.amf", "haarcascade_frontalface_alt2.xml",  
24, "imagen.jpg"); //1
```

```
ASM = new ActiveShapeModel(puntosoriginal);//1
```

//Paso 2: Normalizar imagen.

```
Bitmap imagennormalizada=procesamiento.NormalizarImagen(puntosoriginal, original, ecualizarhist, path);
```

```
//1
```

//Paso 3: Crear Vecindades de los puntos característicos.

```
List<Bitmap> vecindades = ASM.CrearVecindades(imagennormalizada); //1
```

//Paso 4: Crear vectores por la Transformada Discreta del Coseno.

```
List<List<double[]>> DTC = new List<List<double[]>>();//1
```

```
for (int i = 0; i < vecindades.Count; i++)//2 //3
```

```
{
```

```
List<double[,]> matricesDTC = ASM.TransformadaDiscretaCoseno(vecindades[i]);//4
```

```
double[] vectorR = ASM.ConformarVectorCaracteristicoDCT(matricesDTC[0]);//4
```

```
double[] vectorG = ASM.ConformarVectorCaracteristicoDCT(matricesDTC[1]);//4
```

```
double[] vectorB = ASM.ConformarVectorCaracteristicoDCT(matricesDTC[2]);//4
```

```
List<double[]> vectoresDTC = new List<double[]>();//4
```

```
vectoresDTC.Add(vectorR); //4
```

```
vectoresDTC.Add(vectorG);//4
```

```
vectoresDTC.Add(vectorB);//4
```

```
DTC.Add(vectoresDTC);//4
```

```
}
```

```
EDVector vector = new EDVector(DTC);//5
```

```
return vector; //5
```

```
}
```

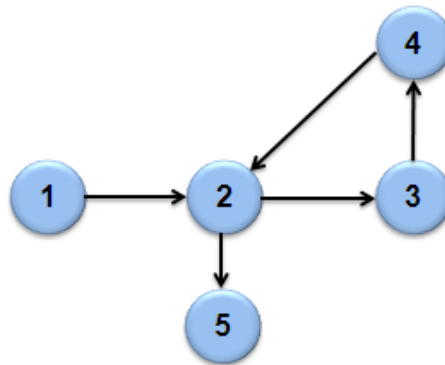


Figura 3.8: Grafo de Flujo: Funcionalidad Obtener vector característico.

Complejidad ciclomática para la funcionalidad ObtenerVectorCaracteristico:

- **Fórmula 1**

$$V(G) = (A \text{ (Aristas)} - N \text{ (Nodos)}) + 2$$

$$V(G) = (5 - 5) + 2 = 2$$

- **Fórmula 2**

$$V(G) = P \text{ (Nodos Predicados)} + 1$$

$$V(G) = 1 + 1 = 2$$

- **Fórmula 3**

$$V(G) = R = 2$$

Caminos independientes obtenidos:

1. 1-2-5
2. 1-2-3-4-5

- ✓ **Funcionalidad: CompararVectorCaracteristicoDTC.**

```

public double CompararVectorCaracteristicoDTC(EDVector v1, EDVector v2,int procedimiento)
{
    int cantidadptos = 68;//1
    double[] diferenciaslocal = new double[cantidadptos];//1
    double diferenciaglobal = 0;//1
    for (int j = 0; j < cantidadptos; j++)//2
    {
        if (procedimiento == 0)//3
            diferenciaslocal[j]=ASM.CompararVectoresDCT(v1.Vectores[j],v2.Vectores[j]);//4
        else//6
    }
}
  
```



```

diferenciaslocal[j]=ASM.CompararVectoresGabor(v1.VectoresGabor, v2.VectoresGabor);//6
}
double valor = 0;//7
for (int j = 0; j < cantidadptos; j++)//7
{
    valor += diferenciaslocal[j];//8
}
valor = valor / cantidadptos;//9
diferenciagobal = valor; //9
return diferenciagobal; //9
}

```

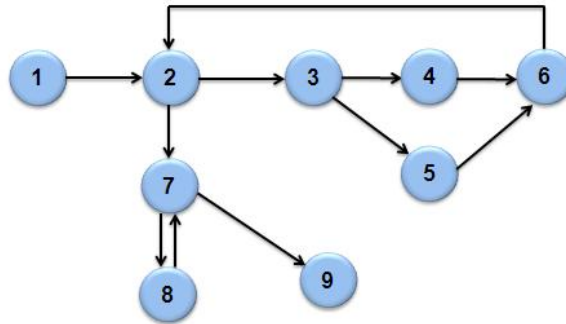


Figura 3.9: Grafo de Flujo: Funcionalidad Comparar vector característico.

✓ Complejidad ciclomática para la funcionalidad CompararVectorCaracteristicoDTC:

- **Fórmula 1**

$$V(G) = (A \text{ (Aristas)} - N \text{ (Nodos)}) + 2$$

$$V(G) = (11 - 9) + 2 = 4$$

- **Fórmula 2**

$$V(G) = P \text{ (Nodos Predicados)} + 1$$

$$V(G) = 3 + 1 = 4$$

- **Fórmula 3**

$$V(G) = R = 4$$

Caminos independientes obtenidos:

- 1-2-7-9
- 1-2-7-8-7-9



- 1-2-3-4-6-2-7-9
- 1-2-3-5-6-2-7-9

3.7 Conclusiones parciales

En este capítulo se definió el estilo de codificación utilizado en la implementación del sistema, lo que permitió mantener una uniformidad en la codificación y mayor claridad a la hora de leer o agregar líneas de código. Se realizaron los diagramas de componentes y despliegue, los cuales posibilitaron un mejor entendimiento del software desarrollado, ayudando así a su implementación. Las pruebas de efectividad permitieron comprobar el buen funcionamiento del algoritmo de reconocimiento facial obteniendo una tasa de error igual de un 1% con un umbral óptimo de aceptación de 0.8. Además se calculó el tiempo promedio de identificación de una persona bajo diferentes condiciones de hardware e imágenes siendo este aproximadamente de 1.2 segundos. El diseño de pruebas de caja blanca y negra contribuyó a establecer los parámetros para maximizar la efectividad del sistema.



Conclusiones generales

Durante el desarrollo de la presente investigación se consultó bibliografía referente al tema de reconocimiento facial para tener un mayor conocimiento sobre la identificación de personas a través de patrones biométricos, para lo cual:

- El análisis de los sistemas existentes tanto en Cuba como en el mundo dio como resultado la necesidad real de desarrollar un sistema de identificación de personas basado en rasgos faciales.
- El análisis de las herramientas y tecnologías necesarias para el desarrollo del sistema de identificación permitió crear una solución conforme a las necesidades del cliente.
- La realización de las pruebas de efectividad, caja blanca y caja negra permitió comprobar el buen funcionamiento del sistema, demostrando la validez del mismo con resultados satisfactorios.

Recomendaciones

Para mejorar el sistema y seguir profundizando en el campo del reconocimiento facial, se recomienda:

- Integrar el sistema a futuras soluciones de identificación que se desarrollen en el departamento de biometría del CISED.
- Utilizar diferentes técnicas para la identificación de rostros, donde el usuario seleccione la técnica deseada y probar la efectividad de la misma.
- Realizar el proceso de identificación a partir de imágenes de video en tiempo real.



Bibliografía referenciada

1. **Bustamante, Isneri Talavera and Hierrezuelo, Jorge Luis Rodriguez.** Reconocimiento de Patrones. La Habana : s.n., 2008.
2. **Goldstein, A.J. and Harmon, L.D. and Lesk, A.B.** Identification of Human faces. Mayo 1971.
3. **Kirby, L. Sirovich.** A Low-Dimensional Procedure for the Chatecterization of Human Faces. 1987.
4. **Delbracio, M. Mateu.** Trabajo final de reconocimiento de patrones: Identificación utilizando PCA, ICA y LDA.
5. **Cenatav** [En línea] <http://www.cenatav.co.cu/es/index.html>
6. **Viera, Daily Armada.** Componente de validación y post- procesamiento de imágenes faciales para su uso en documentos oficiales. La Habana: s.n., 2010.
7. **Echemendia, Harold Riverol and Yoana Pita.** Desarrollo del módulo de reconocimiento de rostros para el Motor de Categorización Inteligente de Contenido. La Habana : s.n., 2012.
8. **Turkand, M. A and Pentland, A.P.** Face Recognition Using Eigenfaces. 1991.
9. **Phillips, P. J.** Overview of the Face Recognition Grand Challenge. San Diego : s.n., 2005.
10. **Blackun, D. M and Bone, J. M and Phillips, P.J.** Facial Recognition Vendor Test. Febrero 2001.
11. **Weissman, Kimberly.** Face Recognition. 2010.
12. **Pomar, Claudia Lora and Mesa, Yisel Delgado.** Algoritmo de detección facial para sistemas de autenticación biométrica. La Habana : UCI : s.n., 2010.
13. **Delbracio, M. and Mateu.** Identificación utilizando algoritmos holísticos .
14. **Albi and Denss.** BMW utiliza reconocimiento de rostro para personalizar los coches. Marzo 18, 2008.
15. **Degtyarev, N. and Seredin, O.** Comparative testing of face detection algorithms. 2010.
16. **Aguerrebere, C..** Proyecto Aguará -Reconocimiento Automático de Caras.
17. **Viola, P. and Jones, M.** Rapid object detection using a boosted cascade of simple features. 2001.
18. **Picasa** [En línea] <http://es.wikipedia.org/wiki/Picasa>
19. **Producto Facebook Facial** [En línea] <http://www.enter.co/vida-digital/facebook-compra-compania-de-reconocimiento-facial/>
20. **Zksoftware**[En línea] <http://www.zk-software.com>
21. **Identix** [En línea] <http://www.identix.com>
22. **Datys.** [En línea] <http://www.datys.cu> 2012



23. Especificaciones Técnicas BIOMESYS AFIS Civil. [En línea] <http://www.datys.cu/wpinfo/producto.aspx?22> 2013
24. **Zhao, W.** Face Recognition. 2003.
25. **Rizo Rodríguez, D, Méndez Vázquez, H and Garcia Reyes.** Estado del arte de los métodos de extracción de rasgos invariantes a la iluminación en imágenes de rostros. Ciudad de la Habana : s.n., 2011.
26. **González, Wintz.** Procesamiento digital de imágenes. 1996.
27. **Acharya, T and Ray, A.K.** Image Processing: Principles and Applications. Canada : s.n., 2005.
28. **Alexander Ocsa M., Alexander Rodriguez V, Herbert Chuctaya H. , Gabriel Humpire M.** Reconocimiento de Rostros mediante Puntos. 2008.
29. **Verdaguer, Sergi Laencina.** Descripción y clasificación de imágenes mediante su color .
30. **Pérez, Luis Blázquez.** Reconocimiento Facial Basado en Puntos Característicos de la Cara en entornos no controlados. Enero, 2013.
31. **Sommerville, Ian.** Ingeniería del Software. Madrid : s.n., 2005.
32. **Correa, Adrian Rivera and González, Guillermo Rodríguez.** Sistema de manejo de llaves y certificados digitales para soluciones de emisión de documentos de identificación electrónicos. La Habana UCI : s.n., 2011.
33. **Escribano, G.F.** Extreme Programming. 2002.
34. **Larmn, Craig.** UML y Patrones. Introducción al análisis y diseño orientado a objetos. . 2004.
35. **Visual Paradigm.** [En línea] <http://www.visual-paradigm.com/>
36. **Jacobson, Booch Rumbaugh.** "Proceso unificado en el desarrollo de software ".
37. **Barahona, Jesus M. Gonzalez.** Intoduccion al software libre. Madrid
38. **Software, Departamento Central de Ingeniería de.** Flujo de trabajo Captura de requisitos. Modelo de Negocio" s.l: Ciudad de la Habana, Universidad de Ciencias Informáticas , 2004.
39. **PostgreSQL.** Documentation. [En línea]. <http://www.postgresql.org/>.
40. **Falgueras, Benet Campderrich.** Ingeniería de Software.
41. **Gary Bradski, Arian Kaebler.** Learning Open CV . 2008.
42. **Agam, Gady.** Introduction to programming with Open CV. Enero 27, 2006.



Bibliografía consultada

1. **Guardado Medina, Ramón Osvaldo and Vallín González, Donato.** Transformada de Fourier en aplicación en el diseño de filtros digitales para el procesamiento de imágenes. 2010.
2. **Alba, José Luis, Cid, Jesús and Mora, Inmaculada.** Extracción de características. 2006.
3. **Hernández, Roger.** Estudio de técnicas de reconocimiento facial. Barcelona: s.n., 2010.
4. **Beck, Kent.** Extreme Programming Explained.2004.
5. **Méndez Vázquez, Heydi.** Algoritmo de reconocimiento de rostros basados en la apariencia local para aplicaciones reales en condiciones variables de iluminación. Ciudad de la Habana .2012.
6. **González, C and Woods, R.** Digital Image Processing. New Jersey. EEUU Prentice Hall, 2007.
7. **Niu, Z. Enhance ASMs Based on AdaBoost-Based Salient Landmarks Localization and Confidence-Constraint Shape Modeling. Berlin: 2005.**
8. **R.S. Pressman.** Software Engineering A Practitioner's Approach. McGraw Hill Publication, 2010.
9. **N. Degtyarev and O. Seredin.** Comparative testing of face detection algorithms. Image and Signal Processing, 2010.
10. **R. Gimeno Hernández.** Estudio de técnicas de reconocimiento facial. 2011.
11. **Patricio Letelier y M^a Carmen Penadés.** Metodologías ágiles para el desarrollo de software: eXtreme Programming (XP), Universidad Politécnica de Valencia.



Glosario de términos

Escala de grises: Modelo de representación de una imagen digital mediante 256 tonalidades de gris.

Ecuilización del histograma: Técnica que consiste en ajustar los niveles de gris de una imagen para obtener una nueva imagen con un histograma uniforme.

Físicos intrínsecos: huella dactilar, geometría de la mano, características del iris, patrones vasculares de la retina, mano, etc.

Framework: estructura de artefactos o módulos concretos con base en la que otro proyecto de software puede ser desarrollado.

Normalización: Acción de transformar una distribución cualquiera a una distribución normal manteniendo la proporción de los datos.

Rasgos conductuales: son aquellos que se soportan sobre características de la conducta del ser humano.

Segmentación: Proceso mediante el cual se divide una imagen en múltiples partes para lograr una mejor representación y obtener así información relevante.

Umbral: Valor mínimo a partir del cual se considera que un elemento pertenece a una clase.



Acrónimos

A	ASM: Active Shape Model
C	<p>CISED: Centro de Identificación y Seguridad Digital.</p> <p>CENATAV: Centro de Aplicaciones de Tecnologías de Avanzadas.</p> <p>CASE: Ingeniería de Software Asistida por Ordenador.</p> <p>CRC: Tarjeta Clase-Responsabilidad-Colaboración.</p>
D	<p>DATYS: Empresa de Desarrollo de Aplicaciones, Tecnologías y Sistemas.</p> <p>DCT: Transformada discreta del coseno.</p> <p>DFT: Transformada discreta de fourier.</p>
E	EER: Taza de error igual.
F	<p>FDD: Feature Driven Development</p> <p>FAR: Falso Positivo. Ocurre cuando un elemento es enmarcado dentro de una clase conocida y este no pertenece a dicha clase.</p> <p>FRR: Falso Negativo. Cuando el sistema devuelve que un elemento es desconocido y sin embargo pertenece una clase conocida.</p>
G	GRASP: Patrones de asignación de



	responsabilidades. GOF: Banda de los Cuatro.
H	HU: Historias de usuario.
I	IDE: Entorno de desarrollo integrado, también conocido como entorno de diseño integrado o entorno de depuración integrada. ISO: La Organización Internacional de Normalización. IPD: Distancia interpupilar.
M	MVC: Modelo Vista Controlador.
O	OpenCV: Visión por Computadora de Código Abierto.
P	PIN: Número de identificación personal, es una contraseña o clave numérica que se utiliza para acceder a cajeros automáticos o servicios de telefonía.
R	RUP: Proceso Unificado de Rational.
S	SGBD: Sistema de Gestor de Base de Datos. SOA: Arquitectura Orientada a Servicios.
U	UCI: Universidad de las Ciencias Informáticas. UML: Lenguaje Unificado de Modelado.
X	XP: Programación Extrema.



Anexos

4.1 Descripción de las historias de usuarios del sistema de identificación de personas basado en rasgos facial.

Historia de Usuario	
Número: 1	Usuario: Técnico, Administrador
Nombre historia: Autenticar Usuario.	
Prioridad en el negocio: Media	Riesgo en desarrollo: Media
Puntos estimados: 1	Iteración asignada: 2
Programador responsable: Rafael Quiles Velázquez	
Descripción: El técnico o el administrador tendrán la posibilidad de identificarse en el sistema.	
Observaciones: Si el sistema no identifica el usuario este mostrará un mensaje comunicando que sus identificadores no son válidos.	

Tabla 4.1: Descripción de la historia de usuario Autenticar Usuario.

Historia de Usuario	
Número: 3	Usuario: Administrador
Nombre historia: Cargar imagen	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 1



Programador responsable: Rafael Quiles Velázquez
Descripción: El administrador cargará la imagen de una dirección física de la computadora.
Observaciones:

Tabla 4.2: Descripción de la historia de usuario Cargar imagen.

Historia de Usuario	
Número: 4	Usuario: Administrador
Nombre historia: Guardar imagen	
Prioridad en el negocio: Baja	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Rafael Quiles Velázquez	
Descripción: El administrador guardara la imagen procesada.	
Observaciones:	

Tabla 4.3: Descripción de la historia de usuario Guardar imagen.

Historia de Usuario	
Número: 6	Usuario: Administrador.
Nombre historia: Gestionar usuario	
Prioridad en negocio:	Riesgo en desarrollo:



Alta	Media
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Rafael Quiles Velázquez	
Descripción: El administrador tendrá la posibilidad de gestionar los usuarios que operarán en el sistema, lo cual incluyen el rol y los datos personales del mismo.	
Observaciones: Gestionar usuario incluye crear, modificar, eliminar y listar usuarios.	

Tabla 4.4: Descripción de la historia de usuario Gestionar usuario.

Historia de Usuario	
Número: 7	Usuario: Administrador.
Nombre historia: Gestionar persona.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Media
Puntos estimados: 2	Iteración asignada: 2
Programador responsable: Rafael Quiles Velázquez	
Descripción: El administrador tendrá la posibilidad de gestionar las personas que serán identificadas, lo cual incluyen los datos personales del mismo.	
Observaciones: Gestionar persona incluye crear, modificar, eliminar y listar personas.	

Tabla 4.5: Descripción de la historia de usuario Gestionar persona.

Historia de Usuario



Número: 2	Usuario: Técnico, Administrador
Nombre historia: Capturar imagen con dispositivo de video.	
Prioridad en el negocio: Alta	Riesgo en desarrollo: Baja
Puntos estimados: 1	Iteración asignada: 1
Programador responsable: Rafael Quiles Velázquez	
Descripción: Los usuarios podrán realizar la captura de la imagen con una cámara digital.	
Observaciones: En caso de que no se pueda realizar la captura por medio de un dispositivo de video se realizará la misma mediante la historia de usuario cargar imagen.	

Tabla 4.6: Descripción de la HU. Capturar imagen con dispositivo de video.

4.2 Descripción de las tarjetas CRC pertenecientes a las clases del negocio del sistema de identificación de persona basado en rasgos faciales.

Nombre de la Clase : ModeloFormaActiva	
Responsabilidades	Colaboradores
<ol style="list-style-type: none"> 1. CrearVecindades 2. AuxiliarVecindad 3. TransformadaDiscretaCoseno 4. LLenarCoeficinteMatriz 5. CompararVectoresDCT 6. CompararVectoresGabor 	



7. ConformarVectorCaracteristicoDCT	
8. ConformarVectorCaracteristicoGabor	

Tabla 4.7: Tarjeta CRC correspondiente a la clase ModeloFormaActiva.

Nombre de la Clase :EDVector	
Responsabilidades	Colaboradores
<ol style="list-style-type: none"> 1. EDVectorDCT 2. EDVectorGabor 	

Tabla 4.8: Tarjeta CRC correspondiente a la clase EDVector.

Nombre de la Clase :DLLConsum	
Responsabilidades	Colaboradores
<ol style="list-style-type: none"> 1. wrapperASM 	

Tabla 4.9: Tarjeta CRC correspondiente a la clase DLLConsum.

Nombre de la Clase :SistemadelIdentificacion	
Responsabilidades	Colaboradores
<ol style="list-style-type: none"> 1. Listapersonas 2. Funciondescriptor 3. InsertarPersona 	<ol style="list-style-type: none"> 1. VectorCaracteristico 2. Sesion 3. AccesoDatos



<ol style="list-style-type: none"> 4. ModificarPersona 5. EliminarPersona 6. ModificarUsuario 7. EliminarUsuario 8. ListarUsuario 9. CompararDescriptores 10. DiferenciaMenor 11. CompararPersonaDCT 12. CompararPersonaxGabor 13. AutenticarUsuario 	
--	--

Tabla 4.10: Tarjeta CRC correspondiente a la clase SistemadelIdentificacion.

Nombre de la Clase: ProcesamientoImagen	
Responsabilidades	Colaboradores
<ol style="list-style-type: none"> 1. RecortarImagenporMayasASM 2. Distancia 3. FiltralImagen 4. Converge 5. FiltrodeBloques 6. RecortarImagen 7. Imagen2Bytes 8. Bytes2Imagen 9. ConvertirVectorString 	



<p>10. ConvertirStringVector</p> <p>11. ComprobarPunto</p> <p>12. DibujarMayasASM</p>	
---	--

Tabla 4.11: Tarjeta CRC correspondiente a la clase ProcesamientoImagen

Nombre de la Clase :VectorCaracteristico	
Responsabilidades	Colaboradores
<p>1. GetVector</p> <p>2. GetImagen</p> <p>3. GetSuma</p> <p>4. GetIdPersona</p>	<p>1. EDVector</p> <p>2. Persona</p>

Tabla 4.12: Tarjeta CRC correspondiente a la clase VectorCaracteristico

4.3 Tareas de Ingeniería del sistema

Iteración 2	
Historias de Usuario	Tareas
Autenticar usuario	<p>1. Validar campos de entrada.</p> <p>2. Encriptar combinación de contraseña.</p> <p>3. Verificar autenticación del usuario.</p> <p>4. Mostrar mensaje de autenticación correcta o incorrecta en el sistema.</p>
Gestionar persona	<p>Insertar persona</p> <p>1. Validar campos de entrada.</p>



	<ol style="list-style-type: none"> 2. Procesado de las imágenes capturadas. 3. Mostrar mensaje de enrolamiento <p style="text-align: center;">Eliminar persona</p> <ol style="list-style-type: none"> 1. Seleccionar la persona que desea eliminar. 2. Eliminar la persona de la base de datos. 3. Mostrar mensaje de persona eliminada correctamente. <p style="text-align: center;">Modificar persona</p> <ol style="list-style-type: none"> 1. Seleccionar la persona a modificar. 2. Validar campos de entrada. 3. Modificar los datos de la persona en la base de datos. 4. Mostrar mensaje modificación efectuada con éxito.
<p>Gestionar usuario</p>	<p style="text-align: center;">Insertar usuario</p> <ol style="list-style-type: none"> 1. Validar campos de entrada. 2. Procesado de las imágenes capturadas. 3. Mostrar mensaje de enrolamiento. <p style="text-align: center;">Eliminar usuario</p> <ol style="list-style-type: none"> 1. Seleccionar el usuario que desea eliminar. 2. Eliminar el usuario de la base de datos. 3. Mostrar mensaje de usuario eliminado



	<p>correctamente.</p> <p>Modificar usuario</p> <ol style="list-style-type: none"> 1. Seleccionar el usuario a modificar. 2. Validar campos de entrada. 3. Modificar los datos del usuario en la base de datos. 4. Mostrar mensaje modificación efectuada con éxito.
--	--

Tabla 4.13: Tareas de ingeniería en la segunda iteración.

4.4 Casos de pruebas de funcionalidad

Casos de Prueba de Funcionalidad	
Código Caso de Prueba: 2	Nombre Historia de Usuario: Registrar persona
Nombre de la persona que realiza la prueba: Amelia Aguilera Reyes	
Descripción de la prueba: Se registran los datos de la persona conjuntamente con la captura de las imágenes faciales y se obtendrá el vector característico para cada una.	
Condiciones de ejecución: Las imágenes que se utilicen deben poseer buenas condiciones de iluminación y la resolución debe ser mayor de 250x250, para obtener resultados satisfactorios en el proceso de identificación	
Entrada/Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Entrada de los datos de la persona: nombre, primer apellido, segundo apellido, carné de identidad y sexo. 2. Obtención de las imágenes faciales a través de un dispositivo de video o se especificará el 	



fichero donde se encuentran las mismas.
3. Proceso de extracción de características para cada una de las imágenes antes capturadas.
4. Proceso de almacenamiento de los datos en la base de datos.
Resultado Esperado: Mensaje que indica el registro satisfactorio de la persona.
Evaluación: Satisfactoria

Tabla 4.14: Caso de prueba de la funcionalidad: Registrar persona.

Casos de Prueba de Funcionalidad	
Código Caso de Prueba: 3	Nombre Historia de Usuario: Eliminar persona
Nombre de la persona que realiza la prueba: Amelia Aguilera Reyes	
Descripción de la prueba: Se selecciona la persona a eliminar de la lista de personas registradas en el sistema y se realizar la acción de eliminar.	
Condiciones de ejecución: El sistema debe tener persona almacenadas en la base de datos.	
Entrada/Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Listar todas las personas registras en el sistema. 2. Seleccionar cual o cuales se quieren eliminar. 3. Activar la acción eliminar persona. 	
Resultado Esperado: Mensaje que indica la eliminación correcta de las personas.	
Evaluación: Satisfactoria	

Tabla 4.15: Caso de prueba de la funcionalidad: Eliminar persona

Casos de Prueba de Funcionalidad



Código Caso de Prueba: 4	Nombre Historia de Usuario: Modificar persona
Nombre de la persona que realiza la prueba: Amelia Aguilera Reyes	
Descripción de la prueba: Se modifican los datos identificativos de una persona como son el nombre, primer apellido, segundo apellido, carné de identidad y sexo.	
Condiciones de ejecución: El sistema debe tener persona almacenadas en la base de datos.	
Entrada/Pasos de ejecución:	
<ol style="list-style-type: none"> 1. Seleccionar en la lista de personas almacenadas en la base de datos del sistema cual se quiere modificar sus datos. 2. Entrada de los datos de la persona: nombre, primer apellido, segundo apellido, carné de identidad y sexo. 3. Proceso de almacenamiento de los datos en la base de datos. 	
Resultado Esperado: Mensaje que indica el registro satisfactorio de la persona.	
Evaluación: Satisfactoria	

Tabla 4.16: Caso de prueba de la funcionalidad: Modificar persona

Prueba de caja blanca a la funcionalidad CompararVectoresDCT.

✓ **Funcionalidad: CompararVectoresDCT.**

```
public double CompararVectoresDCT(List<double[]> descrip1, List<double[]> descrip2)
{
    double valor = 0; //1
    //Pesos para los valores....
    //Para Canal R.....
    int TotalR = 0; //1
    double ValorR = 0; //1
    for (int i = 0; i < 9; i++) //2
```




```
{
    double[] R = descrip1[0];//3
    double[] Rx = descrip2[0]; //3
    double aux = R[i] - Rx[i]; //3
    TotalR += (int)Math.Pow(aux, 2); //3
}
ValorR = Math.Sqrt(TotalR);//4
//Para Canal G.....
int TotalG = 0; //4
double ValorG = 0; //4
for (int i = 0; i < 6; i++)//4
{
    double[] G = descrip1[1];//5
    double[] Gx = descrip2[1]; //5
    double aux = G[i] - Gx[i]; //5
    TotalG += (int)Math.Pow(aux, 2); //5
}
ValorG = Math.Sqrt(TotalG);//6
//Para Canal B.....
int TotalB = 0; //6
double ValorB = 0; //6
for (int i = 0; i < 6; i++)//6
{
    double[] B = descrip1[2];//7
    double[] Bx = descrip2[2]; //7
    double aux = B[i] - Bx[i]; //7
    TotalB += (int)Math.Pow(aux, 2); //7
}
ValorB = Math.Sqrt(TotalB); //8
valor = ValorR + ValorG + ValorB;//8
```



```
return Math.Round(valor);//8
}
```

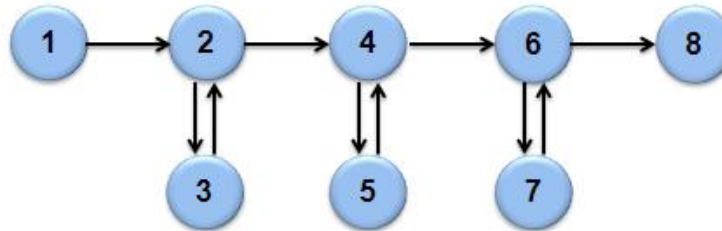


Figura 4.1: Grafo de Flujo: Funcionalidad Comparar Vectores Transformada Discreta del Coseno.

Complejidad ciclomática para la funcionalidad CompararVectoresDCT:

- **Fórmula 1**

$$V(G) = (A \text{ (Aristas)} - N \text{ (Nodos)}) + 2$$

$$V(G) = (10 - 8) + 2 = 4$$

- **Fórmula 2**

$$V(G) = P \text{ (Nodos Predicados)} + 1$$

$$V(G) = 3 + 1 = 4$$

- **Fórmula 3**

$$V(G) = R = 4$$

Caminos independientes obtenidos:

- 1-2-4-6-8
- 1-2-3-4-6-8
- 1-2-3-4-5-6-8
- 1-2-3-4-5-6-7-8

4.5 Pruebas de tiempo al sistema de identificación

Tiempo promedio en identificar una persona, este experimento se realizó en una computadora Intel Celeron a 2.13GHz (1 CPUz), 512MB de Memoria RAM DDR2, con Sistema Operativo Windows XP. Los resultados se muestran en la siguiente figura, del cual se concluye que el tiempo promedio de aceptación o rechazo de una persona es aproximadamente 1,5 segundos.



Pruebas de Tiempo de Comparaciones con extracción imagen entrante		
Prueba1		
Total Imágenes	Tiempo en Segundo	Tiempo en Milisegundo
1	1,456	1456
10	1,46	1460
100	1,475	1475
1000	1,556	1556
Prueba2		
Total Imágenes	Tiempo en Segundo	Tiempo en Milisegundo
1	1,452	1452
10	1,458	1458
100	1,473	1473
1000	1,552	1552
Prueba3		
Total Imágenes	Tiempo en Segundo	Tiempo en Milisegundo
1	1,455	1455
10	1,471	1471
100	1,503	1503
1000	1,532	1532
Prueba4		
Total Imágenes	Tiempo en Segundo	Tiempo en Milisegundo
1	1,428	1428
10	1,452	1452
100	1,536	1536
1000	1,564	1564
Prueba5		
Total Imágenes	Tiempo en Segundo	Tiempo en Milisegundo
1	1,454	1454
10	1,474	1474
100	1,517	1517
1000	1,568	1568
Tiempo Promedio:	Milisegundos	Segundos



	1491,8	1,4918
--	--------	---------------

Tabla 1.16: Representación de los tiempos de identificación de una persona.

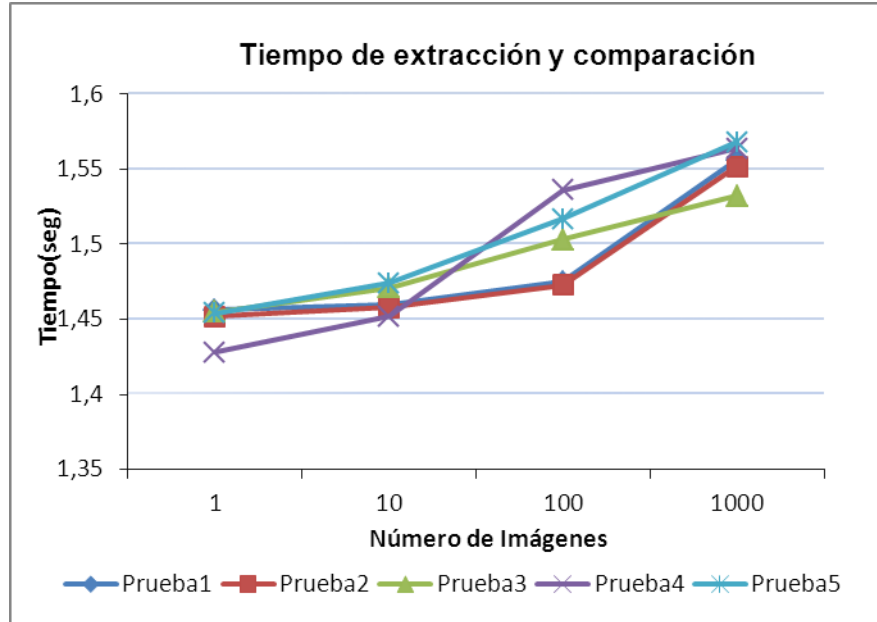


Figura 4.2: Representación de los tiempos de identificación de una persona.

4.6 Interfaces de Usuario.

La interfaz de usuario es el medio con que el usuario puede comunicarse con una máquina, un equipo o una computadora, lo cual su diseño debe ser amigable y consistente por lo que suelen ser fáciles de entender y fáciles de accionar. Una aplicación con una interfaz bien diseñada, además de un buen diseño gráfico, debe tener una buena navegabilidad, usabilidad y distribución de los contenidos.

A continuación se muestran las interfaces para que se tenga un mayor entendimiento de lo que realiza el sistema.



Figura 4.3: Interfaz autenticar usuario.



Figura 4.4: Interfaz de Administración.





Figura 4.5: Interfaz para detectar el rostro.



Figura 4.6: Captura y procesado de la imagen facial.



Figura 4.7: Interfaz para cambiar contraseña.

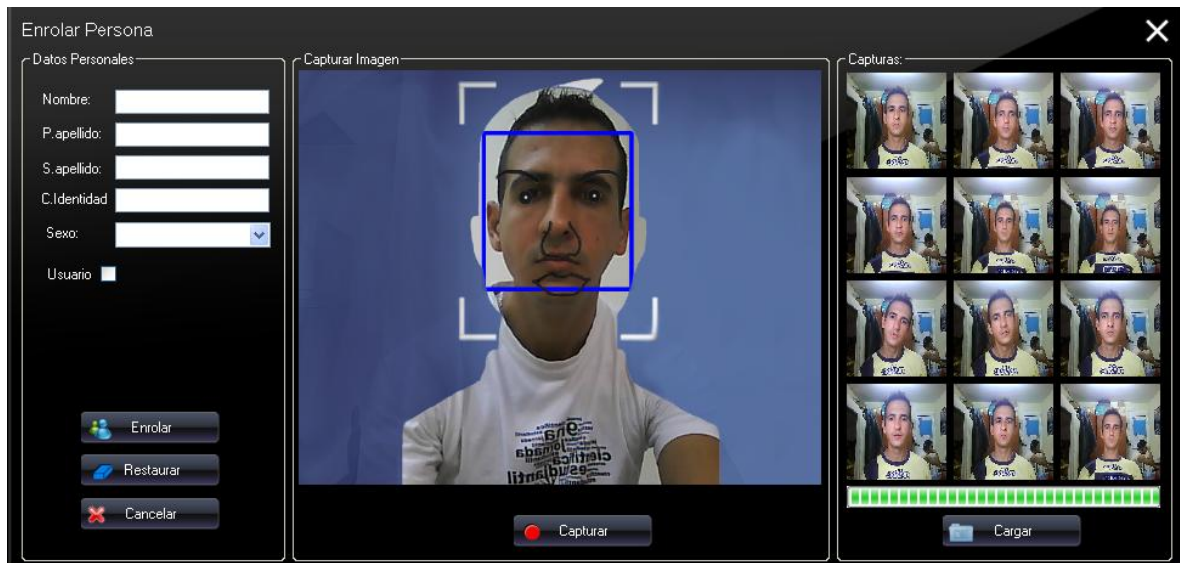


Figura 4.8: Proceso de enrolamiento de una persona.



Modificar Persona

Datos Personales

Nombre: Reisel

P.apellido: de la Rosa

S.apellido: Ge

C.Identidad: 89021183923

Sexo: Hombre

Cancelar Modificar

Figura 4.9: Formulario modificar persona, permite cambiar los datos personales de una persona.

Modificar Usuario

Datos Personales

Nombre: Amelia

P.apellido: Aguilera

S.apellido: Reyes

C.Identidad: 91011429690

Sexo: Mujer

Usuario: areyes

Rol: UsuarioBásico

Cancelar Modificar

Figura 4.10: Formulario modificar usuario, permite cambiar los datos personales del usuario.



Eliminar Persona

Datos Personales:

Nombre	P.Apellido	S.Apellido	Sexo	CI
<input checked="" type="checkbox"/> Roylan	Quiles	Velázquez	M	89071137388
<input type="checkbox"/> Rafael	Quiles	Velázquez	M	89071137380
<input type="checkbox"/> Juan	Perez	Perez	M	78021219230
<input type="checkbox"/> Juan	Perez	Pupo	F	89071137726
<input type="checkbox"/> Reisel	de la Rosa	Ge	F	89022883923
<input type="checkbox"/> Amelíaa	Aguilera	Reyes	F	91011429690
<input type="checkbox"/> Maricel	Palacio	Fagundo	F	80013100000

Cancelar Eliminar

Figura 4.11: Formulario eliminar persona, permite eliminar una persona del sistema.

Eliminar Usuario

Datos Personales:

Nombre	P.Apellido	S.Apellido	Sexo	CI
<input type="checkbox"/> Maricel	Palacio	Fagundo	F	80013100000
<input checked="" type="checkbox"/> Rafael	Quiles	Velázquez	M	89071137380
<input checked="" type="checkbox"/> Amelíaa	Aguilera	Reyes	F	91011429690

Cancelar Eliminar

Figura 4.12: Formulario eliminar usuario, permite eliminar usuarios del sistema.



Ajustar Umbral de Aceptación

Umbral

Umbral de Aceptación: 650

Aceptar

Figura 4.13: Formulario ajustar umbral de aceptación.

4.7 Pruebas al sistema de reconocimiento facial para determinar el umbral de aceptación.

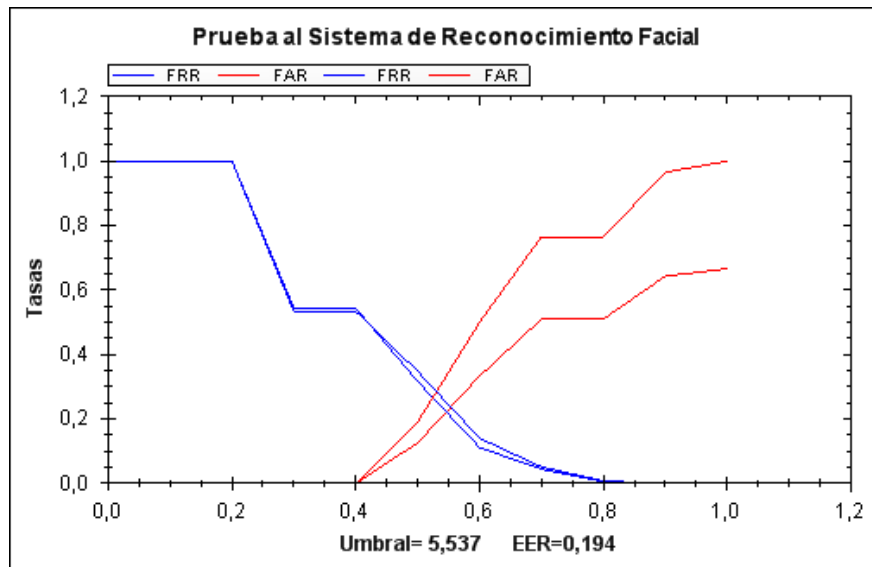


Figura 4.14: Determinación del umbral Equal Error Rate 1.

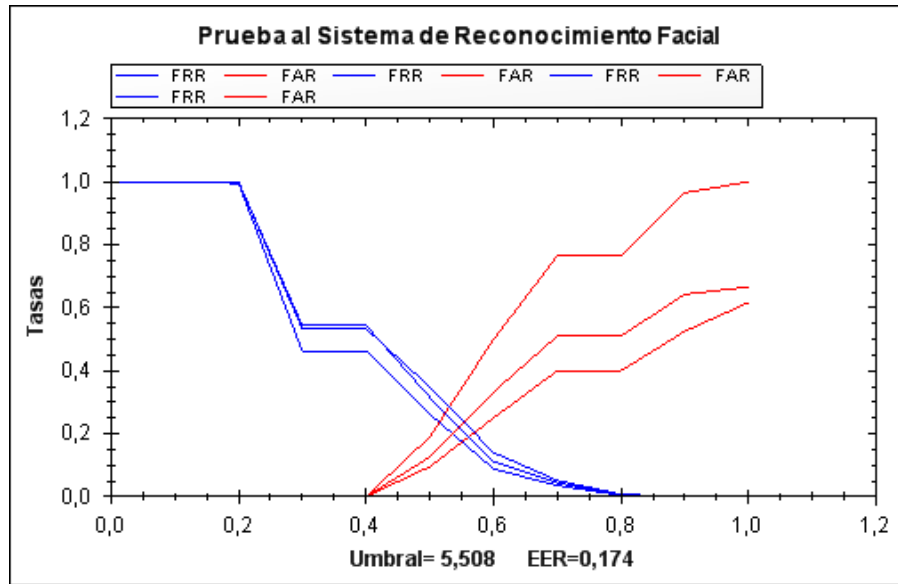


Figura 4.15: Determinación del umbral Equal Error Rate 2.