Universidad de las Ciencias Informáticas

Facultad 1

Trabajo de diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Título:

Componente de validación y post- procesamiento de imágenes faciales para su uso en documentos oficiales.

Autor:

Daily Armada Viera

Tutor

Ing. Yerandy Arias González

Co-Tutor

Ing. Alexander Leyva Morales

La Habana, 2011.

Año 53 de la Revolución.

Declaración de Autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio. Para que así conste firmamos la presente:

_____

Autor: Daily Armada Viera

_____ _____

Tutores: Ing. Yerandy Arias González. ***Cotutor: Ing. Alexander Leyva Morales.***

Datos del Contacto

Tutor: Ing. Yerandy Arias González(yariasg@uci.cu)

Profesor graduado de Ingeniería en Ciencias Informáticas. Se desempeña como jefe del Departamento de Biometría del Centro CISED (Centro de Identificación y Seguridad Digital). Ha participado en eventos científicos como UCIENCIA, RECPAT, Fórum de Ciencia y Técnica, TNT y fue seleccionado como mención al premio CITMA en el curso 2008-2009. Posee dos años de experiencia como profesor recién graduado, el curso 2009-2010 ejerció como tutor de tesis, obteniendo esta última la máxima puntuación.

Co-Tutor: Ing. Alexander Leyva Morales (almorales@uci.cu)

El profesor es Ingeniero en Ciencias Informáticas. Actualmente pertenece al Departamento de Biometría del Centro CISED (Centro de Identificación y Seguridad Digital). Ha participado en eventos científicos como UCIENCIA, RECPAT, Fórum de Ciencia y Técnica. Tiene experiencia impartiendo asignaturas del departamento de programación. Durante el curso 2009-2010 se desempeñó como oponente de tesis, obteniendo esta la calificación máxima de 5 puntos.

Agradecimientos

Agradezco a:

Mi mamá Milagros por apoyarme de forma incondicional,

Mi papá Sigilfridio por su cariño

Mi hermano Dayron por ser tan insoportable y ruidoso, mi amigo, mi apoyo y mi mejor compañía durante el resto de mi vida,

Mi abuela Rosario por ser la mejor abuelita del mundo.

Mis amigos de la universidad los que me han acompañado durante estos cincos anos y me han soportado, tarea la cual considero bastante difícil en algunos momentos. En especial a Yissel mi mejor amiga y confidente, a Pedro(mi negrito lindo)por ser tan pesado conmigo y mostrarme un gran cariño , a Yoel Camilo(ito) por su compañía y por alégrame con sus ocurrencias y hacerme enfadar más de una vez, por criticarlo todo y decirme las cosas que nunca nadie me dirá, a mi prima Alianis que siempre ha sido como una hermana para mi, a Dianiselis, Wilfredo, Yusniera , Guillermo, Onnier, Rubén, Ramón, Darien, Anet y Susana por ser tan buenos amigos y soportar mis malcriadeces más de una vez. A Rafael y Gregorio por el tiempo que dedicaron para que me tesis se terminara.

Mi tutor Yerandy por su apoyo y guía.

Y en general a todas aquellas personas que contribuyeron a mi formación y me apoyaron siempre e hicieron posible el desarrollo de este trabajo.

Dedicatoria

Dedicó este trabajo a las personas más importantes de mi vida que me han hecho quien soy hoy con su cariño y apoyo incondicional:

A mi mamá Milagros,

Mi papá Sigilfridio,

Mi hermano Dayron,

Mi abuela Rosario,

Mi abuelo Ramón,

Y a toda mi familia.

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 reconociendo de personas utilizando los rasgos conductuales o físicos intrínsecos de los individuos, a la cual se le conoce con el nombre de Biometría. La biometría se basa en el reconocimiento de patrones ,como el reconociendo facial, el cual está siendo utilizado hoy en día para evitar el fraude en los pasaportes, soporte al orden público, sistemas de seguridad, identificación de niños extraviados

Nuestro país está llevando consigo el desarrollo de estas nuevas tecnologías en instituciones como el Centro de Aplicaciones de Tecnológicas de Avanzada (CENATAV), en el que no se han obtenido todavía los mejores resultados. Por lo que nuestra universidad pretende contribuir con el desarrollo de nuevas soluciones, en centros como el Centro de identificación y Seguridad Digital CISED, el cual está desarrollando un componente para la validación y post- procesamiento de imágenes faciales para uso en los documentos oficiales, objetivo fundamental en la presente investigación.

Para darle un mayor alcance al componente antes mencionado se ha realizado un estudio de las normas internacionales de la Organización de la Aviación Civil Internacional (OACI), la cual rige como deberá ser la utilización de las imágenes faciales en los documentos oficiales. Además se han elegido las herramientas y metodologías más adecuadas para la facilitación del trabajo, siendo de suma importancia el uso de la librería OpenCV la que proporciona un alto nivel de funciones para el procesamiento de imágenes.

Palabras claves: Biometría, reconocimiento facial.

Índice de Contenido

Índice de Figuras

Índice de Tablas

Introducción

Los humanos a menudo utilizan los rostros para reconocer individuos, los avances en las capacidades de computación en las últimas décadas, ahora permiten reconocimientos similares de forma automática. El estudio de métodos automáticos para el reconocimiento único de personas basadas en uno o más rasgos conductuales o físicos intrínsecos, se conoce como biometría. La biometría informática es la aplicación de técnicas matemáticas y estadísticas sobre los rasgos físicos o de conducta de un individuo, para verificar identidades o para identificar individuos. Una de las características de la biometría son los patrones faciales que utilizan algoritmos de reconocimiento facial y estos anteriormente usaban modelos geométricos simples, pero el proceso de reconocimiento actualmente ha madurado en una Ciencia de Sofisticadas representaciones matemáticas y procesos de coincidencia. Importantes avances e iniciativas han ocurrido en el paso de diez a quince años las cuales han propulsado a la tecnología de reconocimiento facial al centro de la atención.

El reconocimiento facial automatizado es relativamente un concepto nuevo que se enfoca en dos problemas principalmente: el geométrico(basado en rasgos) y el fotométrico(basado en lo visual).El desarrollado de reconocimiento facial comienza en los años 60, el primer sistema semiautomático para reconocimiento facial requería del administrador para localizar rasgos (como ojos, orejas, nariz y boca) en las fotografías antes de que este calculara distancias a puntos de referencia en común, los cuales eran comparados luego con datos de referencia.

En los años 70 Goldstein, Harmon, Lesk, (1)usaron 21 marcadores subjetivos específicos tales como el color del cabello y grosor de labios para automatizar el reconocimiento facial. El problema con estas soluciones previas era que se computaban manualmente. En 1988 Kirby y Sirobich aplicaron análisis de componentes principales, una técnica estándar del álgebra lineal, al problema del reconocimiento facial. 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. (2)

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 (3)en las imágenes, un descubrimiento que permitió sistemas automatizados de reconocimiento facial en tiempo real seguros. Si bien la

aproximación era un tanto forzada por factores ambientales, creó sin embargo un interés significativo en posteriores desarrollos de éstos sistemas.

De1993 a 1997 corrió el programa FERET (FacE REcognition Technology, tecnología de reconocimiento de facial) patrocinado por el departamento de defensa hasta la Agencia de Investigación de Productos de Avance de Defensa (DARPA) de Estados Unidos, 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.

La tecnología inicialmente capturó la atención del público a partir de la reacción de los medios a una prueba de implementación en el Super Bowl de la NFL en enero de 2001, la cual capturó imágenes de vigilancia y las comparó con una base de datos de foto archivos digitales. Esta demostración inició un muy requerido análisis sobre cómo usar la tecnología para satisfacer necesidades nacionales, mientras se tomaban en consideración las preocupaciones sociales y de privacidad del público.

En Mayo de 2004 empezó El gran reto del reconocimiento facial (The Face Recognition Grand Challenge FRGC) (4)que consiste en una serie de problemas reto 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) (5).

Hoy la tecnología de reconocimiento facial está siendo utilizada para combatir el fraude de pasaportes, soporte al orden público, identificación de niños extraviados y minimizar el fraude en las identificaciones.

Actualmente 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. 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.

Hoy en nuestra Universidad existe el Centro de Identificación y Seguridad Digital (CISED), y en él se encuentra el Departamento de Biometría, que estudia los patrones

para el reconocimiento de personas según sus características físicas, como huellas dactilares, firma digital y reconocimiento facial. El centro no cuenta actualmente con ninguna solución que utilice el reconociendo facial para validar la calidad de las imágenes faciales utilizadas en los documentos oficiales dígase documentos oficiales a los pasaportes e identificaciones. Tomando en cuenta que realizándose este proceso de verificación en las imágenes de forma manual el mismo resultaría algo engorroso y tedioso, y la adquisición del algún sistema para la realización automática del proceso resultaría un gran gasto de dinero para el país puesto que la biometría es una tecnología altamente costosa , CISED se dispone a crear soluciones para la identificación de personas y está desarrollando componentes con fines de obtener un buen funcionamiento de estos patrones biométricos, y uno de ellos es el Componente de validación y post- procesamiento de imágenes faciales para documentos oficiales.

Con lo antes mencionado se da a conocer la necesidad que tiene la universidad en el tema de reconocimiento de facial y así se formula como **problema científico** ¿cómo automatizar el proceso de validación y post- procesamiento de imágenes faciales para utilizar en los documentos oficiales?

Con el objetivo de lograr una alternativa de solución, el presente trabajo propone eliminar el problema existente, razón por la cual el **objeto de estudio** queda enmarcado en los procesos de validación y post- procesamiento de imágenes faciales y el **campo de acción** se ha delimitado en los procesos de validación y post- procesamiento de imágenes faciales para la utilización en documentos oficiales. El **objetivo del trabajo** es desarrollar un componente para validar y procesar imágenes faciales en documentos oficiales. Teniendo como **idea a defender** la realización de un componente que automatice la validación y post- procesamiento de imágenes faciales, mejorara la calidad de las imágenes para uso en documentos oficiales.

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

- Diseñar el marco teórico de la investigación.
- Definir la propuesta de herramientas a utilizar para la validación de imágenes faciales en documentos oficiales.
- Diseñar e implementar el componente de validación y post- procesamiento de imágenes faciales en documentos oficiales.

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

- Estudiar los sistemas existentes a nivel mundial y nacional que validen y post procesen las imágenes faciales para un documento oficial.
- Estudiar y clasificar las normas y estándares internacionales de la Organización de la Aviación Civil Internacional (OACI).
- Identificar las herramientas y tecnologías a utilizar en el desarrollo del componente para la validación y post-procesamiento de una imagen.
- Especificar los requisitos del software.
- Definir una arquitectura para la validación y post-procesamiento de una imagen para un documento oficial.
- Realizar diseño de la interfaz según sea necesario.
- Realizar diseño de clases.
- Implementar las funcionalidades del componente para la validación y post-procesamiento de una imagen, en cuanto a:
  - ✓ Buena postura.
  - ✓ Nitidez de la imagen facial.
  - ✓ Mirada al frente.
  - ✓ Balance de colores.
  - ✓ Presencia de espejuelos.
- Aplicar las pruebas de calidad al componente para la validación y el post-procesamiento de imágenes para documentos oficiales.

Métodos teóricos

Analítico-Sintético: Se consulta la bibliografía referente al tema de detención de rostro y la validación de imágenes para documentos oficiales, así como las normas y estándares internacionales para el procesamiento y validación de imágenes en documentos oficiales y se hace uso de la información más adecuada al presente trabajo.

Análisis Histórico-Lógico: Se realiza un estudio de la evolución del reconocimiento de rostro y del nivel que han alcanzado los sistemas desarrollados hasta el momento.

El siguiente trabajo tiene como propósito llegar hasta de fase de prueba del componente propuesto a construir, el mismo está estructurado en 3 capítulos.

En el **Capítulo 1** se hace un estudio sobre el estado del arte de las principales empresas que hacen uso del sistema de reconocimiento facial. Se hace un análisis de las técnicas, tecnologías, herramientas y metodologías a emplear durante la investigación.

En el **Capítulo 2** se abordan las características del sistema y del flujo de trabajo de análisis y diseño, se hace la propuesta del sistema y además se ofrece el modelo de dominio, la definición de los casos de usos, brindan las clases del análisis, las clases del diseño y los diagramas correspondientes a la realización de cada caso de uso que fueron imprescindibles para una mayor precisión en el momento de la constitución de la solución propuesta.

El **Capítulo 3** se muestra el modelo de implementación. Este está compuesto por los diagramas de despliegue y componente. Se aplica pruebas de calidad al componente de validación y post- procesamiento.

Capítulo I Fundamentación Teórica

1.1Introducción

En este capítulo se muestra los resultados de la investigación realizada sobre el estado del arte de las empresas que hacen uso de sistemas de reconocimientos de rostros que existen tanto a nivel internacional como nacional. También se hace referencia a los principales conceptos y características vinculados al tema, se mencionan y explican las normas ICAO por la cuales estará defina la validación de imágenes para el componente de validación y post- procesamiento de imágenes faciales para usar en los documentos oficiales.

También se dan a conocer las principales características de las tecnologías, herramientas, metodologías seleccionadas para el desarrollo del componente, y se argumenta porque el uso de las mismas.

1.2 ¿Qué es un sistema de reconocimiento facial?

El 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 del sujeto extraídas de la imagen o de un fotograma clave de una fuente de video, y comparándolas con una base de datos.

Es utilizado principalmente en Sistemas de Seguridad para el reconocimiento de los usuarios. Consiste en un lector que define las características del rostro, y al solicitar acceso se verifica que coincidan las características del usuario con la BD.

1.3 Funcionamiento del sistema de reconocimiento facial

Primer Paso:

 Detectar una imagen Facial.

Segundo Paso:

 Validar imagen según las Normas ICAO.

Tercer Paso:

 Procesar imagen facial.

1.3.1 Detección de una imagen

La detección de rostros puede ser considerada como un caso específico de la detección de objetos. Lo que se pretende con la detección de rostros es localizar el

rostro o los rostros, si existen, y devolver su ubicación en la imagen y el tamaño de estas. La localización facial se realiza mediante la detección de características faciales, como los ojos, la nariz, etc.

Existen varios factores que impiden la localización del rostro. Uno de estos factores es la posición de este respecto a la de la cámara; otro factor seria la presencia o ausencia de componentes estructurales como la barba, el bigote o unas gafas, 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.

Métodos basados en el aspecto

En contraste con los otros métodos, estos son generados desde una colección de imágenes de entrenamiento de las cuales han de capturar las variaciones representativas del aspecto final. Los sistemas más utilizados son:

- ✓ Eigenface[1]
- ✓ Distribution-based
- ✓ Neural Network
- ✓ Support Vector Machine
- ✓ NaiveBayesClassifier
- ✓ HiddenMarkovModel

Estos métodos son los más usados en la actualidad ya que son los que dan mejores resultados. Esto se debe a que en función de la variabilidad de la colección de imágenes o muestras con las que se realiza en entrenamiento se obtendrían detectores con tasas altas de detección y bajas tasas de falsa alarma. Además de una gran robustez, presentan una eficiencia en el sistema de detección y reducción de coste computacional.

La detección de rostros se utiliza normalmente para:

El reconocimiento o identificación de rostros: donde se compara una imagen de entrada frente una base de datos de imágenes y se confirmará si hay coincidencia o no.

Autentificación de rostros: para verificar la afirmación de la identidad de un individuo en una imagen previa.

[1]Eigenface: Sistema de vectores propios utilizado en la ciencia y las tecnologías de las computadoras, que obtiene información de una imagen mediante una aplicación informática para automáticamente identificar o verificar a una persona.

Seguimiento de rostros: este proceso estima continuamente la localización y posible orientación de un rostro en una secuencia de imágenes en tiempo real.

Reconocimiento de expresiones faciales: identifica los estados de ánimo.

1.3.2 Validar imagen según las Normas ICAO

La Organización de la Aviación Civil Internacional (OACI) establece normas y regulaciones internacionales necesarias para garantizar la seguridad y eficiencia y regularidad del transporte aéreo y sirve de catalizador para la cooperación en todas las esferas de la aviación civil entre sus 185 Estados contratantes.

En la tercera edición del Doc. 9303, Parte 3 incorpora la nueva forma de interfuncionamiento mundial opcional para la identificación biometría del titular y para el almacenamiento de los datos conexos en un circuito integrado sin contacto. (6)

Existen especificaciones técnicas comunes para los documentos de viaje oficiales de lectura mecánica de tamaño 1 y tamaño 2(dv1 y dv2).

Configuración general del dvLM[2] (dv1y dv2)

El dvLM tiene una configuración normalizada para facilitar la lectura visual y mecánica de los datos a escala mundial (interfuncionamiento mundial).

Para satisfacer los distintos requisitos de las leyes y prácticas de los Estados, y para lograr la normalización máxima dentro de las divergencias de estos requisitos, el dvLM está divido en siete zonas como se indica a continuación. Las zonas I a VI constituyen la zona de inspección visual (ZVL).La zona VII es la lectura mecánica (ZVM).

Zona I Encabezamiento (obligatorio)

Zona II Datos personales (obligatorios y opcionales)

Zona III Datos del documento (obligatorios y opcionales)

Zona IV Firma o marca habitual (obligatoria)

Zona V Elemento de identidad (obligatorio)

Zona VI Datos (opcionales)

Zona VII Zona de lectura mecánica (ZML) (obligatoria)

Contenido y empleo de las zonas

1. Datos .Los datos que se incluirán en las zonas y la preparación de estas, así como las orientaciones sobre sus dimensiones, serán como se describe a continuación.

Zonas obligatorias.

1.1 La Zona I en el anverso del dvLM identifica el Estado u organismo expedidor y el documento y el documento.

[2] dvLM :Documento de viaje oficial de lectura mecánica en forma de tarjeta

1.2 Los datos aparecerán en orden normalizado en las Zonas II y III.

1.3 Las zonas II y III contienen un campo en el que pueden incluirse datos opcionales. El campo opcional en la Zona II se utilizara para datos personales y el campo opcional de la Zona III para detalles relacionados con el documento. Cuando un Estado u organismo expedidor no utiliza los campos opcional es del as Zonas II y III, no hay necesidad de reservar espacio para ello en el dvLM.

1.4 La Zona IV contiene la firma y la marca habitual del titular. El Estado expedidor decidirá la aceptabilidad de una arca habitual del titular.

1.5 La Zona V contendrá los elementos de identificación personal que incluirán un retrato del titular solamente .A discreción del Estado u organismo expedidor, los campos de nombre en la Zona II y la firma o marca habitual del titular de la Zona IV pueden suponerse a la Zona V siempre que este no afecte el reconocimiento de datos en ninguna de las tres zonas.

1.6 Las Zonas VII contendrán los datos de lectura mecánica. Debido al menor tamaño del dv1, para hacer lugar a los datos requeridos, se incluyen en la ZML tres líneas de datos de lectura mecánica, mientras que el dv2 de mayor tamaño tiene dos líneas de datos de lectura mecánica.

2. Zona de datos opcionales. La Zona VI, que aparece en el reverso del dvLM, es una zona para datos opcionales que se utilizaran a discreción del Estado u organismo expedidor. La Zona VI aparecerá siempre, independientemente de si se usa o no.

3. Flexibilidad dimensional de las Zonas I a V.

3.1Las Zonas I a V pueden ajustarse en tamaño y forma de las especificaciones generales sobre dimensiones del dvLM para amoldarse a los diversos requisitos de los Estados u organismos expedidores. No obstante, todas las zona sestarán demarcadas por líneas rectas y todos los ángulos donde se unen las líneas rectas serán ángulos rectos(es decir 90 grados).

3.2 Si un Estado u organismo expedidor decide producir un dvLM que contenga un borde transparente o imprimible alrededor de la tarjeta, el resultado será la reducción del espacio disponibles dentro de las zonas. Las dimensiones completas y las demarcaciones de las zonas en el dvLM se medirán a partir del canto exterior del borde inimprimible, que es el borde exterior del dvLM.

3.3 La Zona I estará situada a lo largo del canto superior del dvLM y se extenderá por toda su anchura. El Estado u organismo expedidor podría variar la dimensión vertical de la Zona I, según lo requieran, pero la dimensión será suficiente para permitir la interpretación legible de los datos en la zona y no excederá de 11,0 mm (0,43 in) para cada cara del dvLM.

3.4 La Zona V estará situada de modo que su borde izquierdo coincida con el borde izquierdo del dvLM. La Zona V puede variar de tamaño pero no superará la tolerancia máxima especificada.

3.5 La Zona V podrá moverse verticalmente a lo largo del borde izquierdo del dvLM y cubrir parte de la Zona I, siempre y cuando no se oculten los detalles que figuran en una u otra zona.

3.6 El límite superior de la Zona II coincidirá con el límite inferior de la Zona I.

3.7 Cuando exista necesidad especifica de que el campo de nombre se extienda a todo lo ancho del dvLM, la Zona II podrá ampliarse igualmente a todo lo ancho del dvLM.

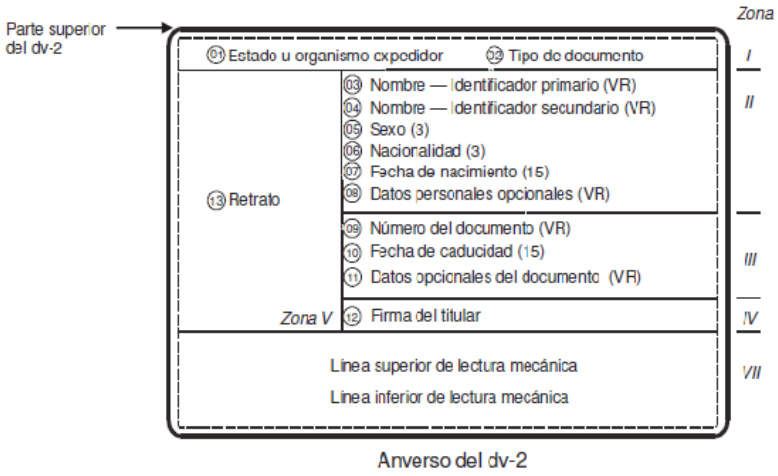

Figura 1.Orden datos DVML. Anverso

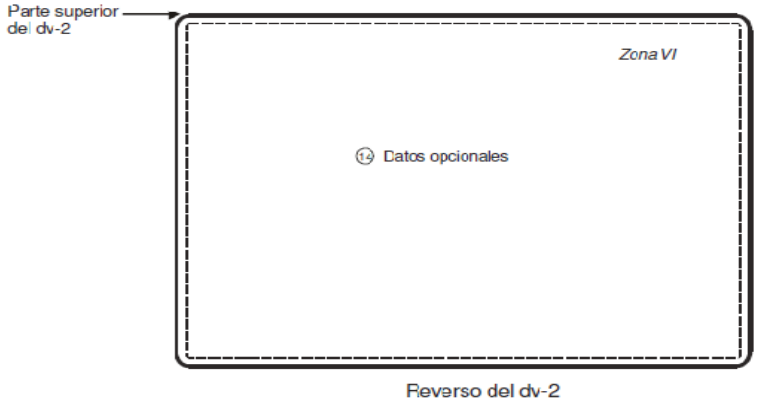

Figura 2.Orden datos DVML. Reverso.

1. La ZIV se basa en una densidad máxima de impresión de 8 líneas por 2.54 mm (0.4 in) y una densidad de impresión horizontal de 15 caracteres por 25.4 mm (1.0in).
2. (VR)= número máximo de caracteres por 25.4 mm (1.0in).
3. ( )= número máximo o fijo de caracteres.
4. O= número de campos.
5. Las líneas que limitan las zonas pueden omitirse en el documento real.

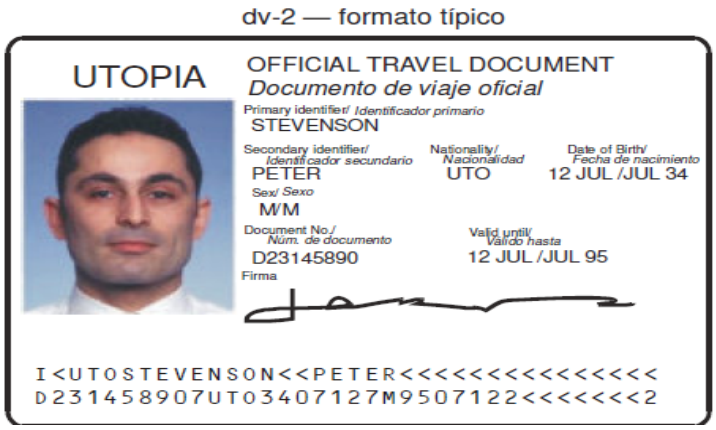

Figura 3.Documento de viaje oficial.

Iluminación: Para tomar la pose de frente del rostro completo se empleará una luz uniforme apropiada, o sea, se empleará técnicas adecuadas de iluminación para lograr tonos de piel naturales (evitando la predominancia de color) y un alto grado de detalle y reducir al mínimo las sombras, las manchas luminosas, los ojos rojos y los reflejos (como los que causan a veces las gafas).

Fondo: Se empleará un fondo uniforme de color claro que contraste con el rostro y el cabello. En el caso de retratos de a color, se recomienda un fondo azul claro, beige, marrón claro, gris pálido o blanco.

Calidad del retrato captado: La calidad del retrato original debe ser por lo menos comparable a la calidad mínima aceptable para las fotografías (una resolución equivalente a 6-8 pares de líneas por milímetro.)

Color: El retrato será en blanco y negro o en colores verdaderos que representen bien al titular.

Adornos Faciales: El estado expedidor empleará su discreción con respecto a la medida en que pueden aparecer en el retrato adornos faciales (p. Ej. Anillos de nariz, clavos ornamentales). Sólo podrán aparecer los adornos faciales que tengan uso permanente.

Reproducción digital:

Calidad de la reproducción digital: Con la reproducción digital debe lograrse una representación exacta y reconocible del titular legítimo del documento. La calidad de un retrato reproducido debe ser visualmente comparable a la de una fotografía mínimamente aceptable.

Bordes: No se emplearán bordes o encuadres para resaltar una reproducción impresa digitalmente.

Compatibilidad con el fondo de seguridad aplicado: Las reproducciones impresas digitalmente serán compatibles con el fondo de seguridad aplicado en la zona V, o

sea, la impresión del fondo de seguridad no interferirá con la lectura apropiada del retrato exhibido y viceversa.

Retratos de bebé: El retrato de un bebé debería producirse en lo posible ajustándose a las especificaciones mencionadas. Idealmente el bebé deberá fotografiarse en posición erguida pero es aceptable captar un retrato con el bebé acostado sobre una manta blanca o de color uniforme. Alternativamente, puede colocarse al bebé en un asiento para bebés pero deberá haber un fondo blanco o de color uniforme detrás de su cabeza. Los ojos deberán estar abiertos y no deberán verse manos que lo sostengan.

Directrices ilustrativas para los retratos en los DVLM

Pose

1.1 La fotografía tendrá menos de seis meses de captada a la fecha de la solicitud.

1.2 Deberá mostrar una toma de primer plano de la cabeza y los hombros.

1.3 La fotografía debe captarse de modo que una línea horizontal imaginaria entre los centros de los ojos resulte paralela al borde superior de la imagen.

1.4 El rostro debe ser enfocado con total nitidez y no deberán aparecer imperfecciones como manchas de tinta o arrugas.

1.5 La fotografía deberá mostrar al sujeto en pose frontal y mirando directamente a la cámara con expresión neutra y boca cerrada.

1.6 La distancia desde el mentón hasta la coronilla (coronilla es la parte superior de la cabeza si no hubiera cabello) ocupará del 70% al 80% de la dimensión vertical de la fotografía.

1.7 Los ojos deberán estar abiertos y no deben haber cabello que los oculte.

1.8 Si en sujeto usa gafas, la fotografía debe mostrar los ojos claramente sin reflejos luminosos en ellas. Las gafas no tendrán lentes coloreadas. De ser posible se evitarán las monturas(armazones) gruesas y se asegurará de que estas no cubren parte de los ojos

1.9 No se permiten tocados, cabellos, peinados o adornos faciales que oculten el rostro.

1.10 La fotografía tendrá un trasfondo monocromático de color claro

1.11 No habrán otras personas u objetos en la fotografía.

Iluminación, exposición y equilibrio cromático

2.1 La iluminación será uniforme sin sombras o reflejos sobre el rostro o en el trasfondo.

2.2 Los ojos del sujeto no deben aparecer rojos.

2.3 La fotografía debe tener brillo y contrastes apropiados.

2.4 Cuando la fotografía sea de color, el proceso de iluminación y fotográfico debe equilibrar los colores para producir tonos cutáneos naturales.

Presentación del retrato a la autoridad expedidora

3.1 Cuando el retrato se presente a la autoridad expedidora e forma impresa, la fotografía producida ya sea utilizando técnicas fotográficas convencionales o técnicas digitales, deberá estar sobre el papel fotográfico o de buena calidad y tenerlas dimensiones máximas específicas.

3.2 Cuando el retrato se presente a la autoridad expedidora en forma digital, deben cumplirse los requisitos especificados por la autoridad expedidora.

Cumplimiento de las normas internacionales

4.1 La fotografía cumplirá con las definiciones apropiadas establecidas en ISO/IEC 19794-5

Mostrar tonos cutáneos naturales si se presenta impreso, lo estará en papel de alta calidad con alta resolución. Los retratos captados con cámara digital serán de alta calidad y resolución se imprimirán en papel fotográfico.

Calidad del retrato

El retrato tendrá menos de 6 meses de captado. Sus dimensiones serán de 35x45 mm (1.38 x 1.77 in) en ancho y alto. El estado expedidor ajustará el tamaño adecuado del retrato para un dv1 o un dv2. El retrato mostrará una toma de primer plano de la cabeza y los hombros del solicitante. El rostro ocupará hasta el 70%-80% de la dimensión vertical de la fotografía. El retrato estará nítidamente enfocado será de elevada calidad y no presentará arrugas o manchas de tinta. El retrato mostrará al solicitante mirando directamente a la cámara. Tendrá el brillo y contraste apropiados. Si es de color deberá

Estilo de iluminación

El retrato será de color neutro y mostrará al solicitante con los ojos abiertos y claramente visibles; no habrá cabello sobre los ojos. El solicitante estará de frente a la cámara, no mirando sobre el hombre (estilo de retrato artístico).

La cabeza estará derecha de modo que una línea horizontal imaginaria trazada ente los centros de los ojos sea paralela al borde superior de la imagen. Ambos bordes de la ara serán claramente visibles. Él fondo será monocromático de color ligero. La iluminación será uniforme sin sombras ni reflejos sobre el rostro. No habrá ojos cerrados

Figura 4.Ejemplo de iluminación.

Gafas y tocados

Gafas:

El retrato mostrara los ojos claramente sin reflejos luminosos en las gafas y sin lentes coloreadas. De ser posible evítense las monturas gruesas. Las monturas no cubrirán parte alguna de los ojos.

Figura 5.Ejemplo de uso de gafas.

Tocados

No se aceptaran tocados excepto en circunstancias en que la autoridad estatal competente los apruebe específicamente. Tales circunstancias pueden ser religiosas, médicas o culturales.

Figura 6.Ejemplo de uso de tocados.

Expresión del rostro

El retrato mostrara al solicitante solo son otras personas, respaldo de sillas o juguetes visibles. El solo solicitante mirara directamente a la cámara con expresión neutra y boca cerrada.

Figura 7.Ejemplo de expresión del rostro.

1.3.3 Procesamiento de imágenes

El procesamiento digital de imágenes es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información.

En presente trabajo pretende que al terminar el proceso de normalización cumplan con lo siguiente:

Profundidad del Campo: La pose central del rostro completo estará en foco desde la coronilla hasta la barbilla y desde la nariz hasta las orejas.

Retrato: Las dimensiones del retrato no serán superiores a 45.0 mm x 35.0 mm (1.77 in x 1.38 in) e inferiores a 32mm x 26.0 mm (1.26 in x 1.02 in). Se recomienda la utilización de imágenes digitales, dado que las fotografías adheridas pueden ser objeto de sustituciones fraudulentas.

Pose: En el retrato exhibido se mostrará el rostro del titular legítimo en una pose frontal del rostro completo en la que ambos ojos sean visibles.

Proceso de filtrado

Es el conjunto de técnicas englobadas dentro del pre procesamiento de imágenes cuyo objetivo fundamental es obtener, a partir de una imagen origen, otra final cuyo resultado sea más adecuado para una aplicación específica mejorando ciertas características de la misma que posibilite efectuar operaciones del procesado sobre ella.

Los principales objetivos que se persiguen con la aplicación de filtros son:

- ✓ Suavizar la imagen: reducir la cantidad de variaciones de intensidad entre píxeles vecinos.

- ✓ Eliminar ruido: eliminar aquellos píxeles cuyo nivel de intensidad es muy diferente del de sus vecinos y cuyo origen puede estar tanto en el proceso de adquisición de la imagen como en el de transmisión.
- ✓ Realzar bordes: destacar los bordes que se localizan en una imagen.
- ✓ Detectar bordes: detectar los píxeles donde se produce un cambio brusco en la función intensidad.
- ✓ Por tanto, se consideran los filtros como operaciones que se aplican a los píxeles de una imagen digital para optimizarla, enfatizar cierta información o conseguir un efecto especial en ella. (7)

Tipos

Filtro paso bajo (suaviza miento): utilizados para eliminar ruido o detalles pequeños de poco interés puesto que sólo afecta a zonas con muchos cambios. La frecuencia de corte se determina por el tamaño del kernel y sus coeficientes. Se emplean diversos kernels:

Promedio: Promedio de pixeles vecinos

(Kernel de unos).

Paso bajo en frecuencia.

Media: reemplaza cada pixel por el valor medio de sus continuos.

Mediana: Sustituye por el valor de la mediana de los píxeles vecinos (normalmente se comporta mejor que el de promedio).

Gaussiano: Aproximación a la distribución gaussiana.

Filtro paso alto (atenuamiento): intensifica los detalles, bordes y cambios de alta frecuencia, mientras que atenúa las zonas de tonalidad uniforme. Esto permite una mejor identificación posterior de los objetos que se encuentren en la imagen, puesto que el brillo se hace mayor en las zonas con frecuencias más altas, al mismo tiempo que se oscurecen las zonas de frecuencias bajas. Es común la aparición de ruido tras el proceso.

- **Realce de bordes por desplazamiento y diferencia:** sustrae de la imagen original una copia desplazada de la misma. Así, es posible localizar y hacer resaltar los bordes existentes y que se quieran obtener según el modelo de kernel aplicado:
  - ✓ Horizontal.
  - ✓ Vertical.
  - ✓ Horizontal/Vertical (diagonal).
- **Realce de bordes mediante Laplace**: este tipo de filtros realza los bordes en todas direcciones (los resultados que se obtienen pueden considerarse como una “suma” de los obtenidos tras aplicar todos los modelos del tipo anterior). En esta ocasión se

trabaja con la segunda derivada, que permite obtener unos mejores resultados, a pesar del aumento del ruido que se produce en la imagen.

Figura 8.Imagen original y resultante tras filtros Laplaciano y Sobel.

- **Resalte de bordes con gradiente direccional:** empleado para destacar y resaltar con mayor precisión los bordes que se localizan en una dirección determinada. Trabaja con los cambios de intensidad existentes entre píxeles contiguos.

- **Detección de bordes y filtros de contorno (Prewitt y Sobel):** al igual que los anteriores, se centra en las diferencias de intensidad que se dan píxel a píxel. Son utilizados para obtener los contornos de objetos y de este modo clasificar las formas existentes dentro de una imagen. Este tipo de filtros requieren un menor coste computacional. (8)

1.4 Empresas que utilizan el reconocimiento de rostro

ZKSoftware Inc.

ZKSoftware Inc. es una compañía china de alta tecnología especializada en inversiones de riesgo, finanzas, valores, bienes raíces y la producción de la tecnología biométrica basada en productos a través de OEM. ZKSoftware se dedica a la promoción, divulgación y localización de la tecnología biométrica. Con más de 2.500 grupos de productos de tipo biométrico de huellas digitales vendidas y aplicadas en las instituciones gubernamentales, el Ministerio de Seguridad Pública, Ministerio de Seguridad Nacional, Ministerio de Telecomunicaciones, el Banco de China. (9)

Identix

Identix, es una de las compañías que fabrican tarjetas de identidad con datos biométricos como el iris o de huellas dactilares. Identix participó en un ensayo de seis meses de la tecnología biométrica para el Servicio de Pasaportes del Reino Unido, que comenzó en abril de 2004. Identix provee la tecnología que verifica la captura facial y la firma digital. El 4 de octubre de 2005, Identix recibido pedidos por un valor total de $ 1,3 millones a partir de dos organismos no identificado del gobierno de EE.UU. para su tecnología de huellas digitales y reconocimiento facial. Identix ofrece

soluciones para una amplia gama de aplicaciones para mercados que incluyen seguridad de la empresa corporativa, intranet, extranet, Internet, acceso celular a Internet y la seguridad, comercio electrónico, gobierno y agencias de la ley. Identix tiene más instalaciones biométricas en todo el mundo que cualquier otra compañía, y ha formado alianzas estratégicas con líderes en la industria como Motorola, Compaq, Toshiba, VeriSign, Novell, Dell, Tronic clave, Micro SMC, Cherry, y Unisys. (10)

Identity Solutions

L-1 Identity Solutions es una empresa pionera en el campo de reconocimiento de rostros, que ofrece una gama completa de productos biométricos (iris, rostro, huellas, palma de la mano), para programas de administración de identificación de civiles patrocinada por el gobierno y procedimientos para la identificación penal para agencias policiales y militares. La misma cuenta con un amplio mercado, incluyendo federales, estatales y locales, la policía, los servicios financieros, y los viajes. L-1Identity ofrece soluciones como el FaceIt, Mobile-Eyes, Mobile-Eyes, ABIS System para en trabajo con datos biométricos. (11)

CENATAV

El CENATAV es un centro orientado a las investigaciones teóricas y aplicadas en el área del Reconocimiento de Patrones y la Minería de Datos. 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, 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. (12)

El CENATAV se creó en el 2004 y está en proceso de consolidación. Además de su actividad de investigación teórica y aplicada, brinda servicios de información científico - técnica, consultoría, cursos de posgrado en Reconocimiento de Patrones y Minería de Datos.

El Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV) 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. CENATAV está desarrollando una aplicación para el procesamiento de

imágenes de nombre FaceQuality la cual sirve de guía fundamental en el proceso de desarrollo del componente propuesto en esta investigación. (13)

1.5 Tecnologías, metodologías y herramientas

En este epígrafe se tratan las principales características y aspectos fundamentales relacionados con las tecnologías, metodología y herramientas que pueden ser utilizadas para el desarrollo del componente de validación y post- procesamiento de imágenes faciales para documentos oficiales.

1.5.1 Tecnologías que utilizan el reconocimiento de rostro

BMW utiliza reconocimiento de rostro para personalizar los coches

Varias empresas utilizan un **sistema de reconocimiento de rostro** para determinar si el conductor está a punto de quedarse dormido, pero **BMW** ha añadido otras características a su tecnología.

BMW utiliza el reconocimiento de rostro para **identificar a la persona detrás del volante** luego **ajusta de forma automática la configuración del coche** a las características y preferencias del conductor.

Tan pronto como **el sistema reconoce al conductor** se ajustan automáticamente los espejos, volante, asiento e incluso su estación de radio favorita. Otro beneficio es la **capacidad de almacenar perfiles** de diferentes conductores.

Se ve que la tecnología utilizada para la **seguridad del piloto** está tomando otros rumbos pues también puede ser utilizada para brindarle confort y facilitarle la conducción. (14)

Sistema de Acceso por Reconocimiento facial.

Equipo de video vigilancia.

Descripción de sistema de acceso por reconocimiento facial.

El nuevo sistema de acceso por reconocimiento facial dispone de los últimos avances en sensores de imágenes y reconocimiento. De diseño moderno y compatible para cualquier tipo de estética en edificios y oficinas. El dispositivo utiliza reconocimiento de rostro en 3D utilizando 2 cámaras que permiten establecer un reconocimiento facial de forma algorítmica en fracciones de segundos. Elegantemente diseñado con una interface de software amigable, el sistema de reconocimiento facial le permitirá bloquear aquellos accesos que usted considere

pertinentes. Con capacidad de registrar hasta 500 rostros, este modelo CVJB-G107 es perfecto para su mediana empresa como:

- ✓ Empresas de diseños.
- ✓ Compañías de seguros.
- ✓ Negocios electrónicos.
- ✓ Empresas financieras
- ✓ Otras empresas en donde se desee impresionar al cliente manteniendo los más altos estándares.

El equipo cuenta con 2 cámaras con sensores de reconocimiento en 3D equipadas con visión nocturna, monitor 3.5 " tft y teclado táctil, entrada USB y Ethernet para protocolos de conexiones tcp/ip, contribuyendo a una instalación y un uso diario de forma sencilla. En caso que la conexión de red falle, puedes descargar los registros utilizando un Pendrive por el puerto USB.
El equipo incluye un software de reconocimiento facial y de gestión del mismo que te permite controlar horarios de empleos, definir vacaciones, horas extras, etc. El reporte es muy completo y puedes utilizarlo para más de una aplicación. (15)

1.5.2 Lenguaje para el modelado de objetos

Un lenguaje para el modelado de objetos es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar un diseño de software orientado a objetos. El uso de un lenguaje de modelado es más sencillo que la auténtica programación, pues existen menos medios para verificar efectivamente el funcionamiento adecuado del modelo. Esto puede suponer también que las interacciones entre partes del programa den lugar a sorpresas cuando el modelo ha sido convertido en un software que funciona.
Toda metodología de desarrollo de software utiliza un lenguaje para el modelado de objetos, para la representación de sus diagramas y artefactos.

1.5.2.1Lenguaje Unificado de Modelado UML

Por sus siglas en inglés, (Unified Modeling Language) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Esta consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Mediante UML es posible establecer la serie de requerimientos y estructuras necesarias para

plasmar un sistema de software previo al proceso intensivo de escribir código.UML puede utilizarse para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra gran cantidad de software.

UML ofrece un estándar para describir un plano del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. (16) (17)

De forma general las principales características de UML son:

- Lenguaje unificado para la modelación de sistemas.
- Tecnología orientada a objetos.
- El cliente participa en todas las etapas del proyecto.
- Corrección de errores viables en todas las etapas.
- Aplicable para tratar asuntos de escala inherentes a sistemas complejos de misión crítica, tiempo real y cliente/servidor.

Se escoge UML porque:

- Permite modelar sistemas utilizando técnicas orientadas a objetos.
- Permite especificar todas las decisiones de análisis y diseño, construyéndose así modelos, precisos, no ambiguos y completos.
- Puede conectarse con lenguajes de programación (ingeniería directa e inversa).
- Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones).
- Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.
- Algunos beneficios de UML:
- Mejores tiempos totales de desarrollo (de 50% o más).
- Modelar sistemas (y no solo de software) utilizando conceptos orientados a objetos.
- Establecer conceptos y artefactos ejecutables.
- Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimizar de costos.

1.5.3 Herramientas de modelado

Las herramientas de modelado de objetos, son fundamentales para el análisis del sistema. Hay varias herramientas creadas para el desarrollo de la Ingeniería de

Software. Estas existen con el fin de desarrollar programas, utilizando técnicas de diseño y metodologías bien definidas, soportadas por herramientas automáticas.

1.5.3.1 Herramienta Case

CASE es una sigla, que corresponde a las iníciales de: ComputerAided Software Engineering; y en su traducción al Español significa Ingeniería de Software Asistida por Computación.

El concepto de CASE es muy amplio; y una buena definición genérica, que pueda abarcar esa amplitud de conceptos, sería la de considerar a la Ingeniería de Software Asistida por Computación (CASE), como la aplicación de métodos y técnicas a través de las cuales se hacen útiles a las personas comprender las capacidades de las computadoras, por medio de programas, de procedimientos y su respectiva documentación.

1.5.3.1.1 Visual Paradigm

Visual Paradigm para UML es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Este software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, ingeniería inversa, generar código desde diagramas y generar documentación. La herramienta UML CASE también proporciona abundantes tutoriales de UML, demostraciones interactivas de UML y proyectos UML. Es muy sencillo de usar, fácil de instalar y actualizar. (18)

1.5.3.2 Rational Rose Enterprise

Rational Rose es una herramienta de desarrollo basada en modelos que se integra con las bases de datos y los IDE de las principales plataformas del sector. IBM Rational Rose Enterprise es uno de los productos más completos de la familia Rational Rose. Todos los productos de Rational Rose dan soporte a Unified Modeling Language (UML), pero no son compatibles con las mismas tecnologías de implementación. Rational Rose Enterprise es un entorno de modelado que permite generar código a partir de modelos Ada, ANSI C++, C++, CORBA, Java/J2EE, Visual C++ y Visual Basic. Al igual que todos los productos de Rational Rose, ofrece un lenguaje de modelado común que agiliza la creación del software. (19)

1.5.4 Entornos integrados de desarrollo

Un Entorno Integrado de Desarrollo (IDE, Integrated Development Environment) es un sistema que facilita el trabajo del desarrollador de software, integrando sólidamente la edición orientada al lenguaje, la compilación o interpretación, la depuración, etc.

Los IDE proveen un marco de trabajo amigable para la mayoría de los lenguajes de programación tales como C++, Python, Java, C#, Delphi, Visual Basic, etc. En algunos lenguajes, un IDE puede funcionar como un sistema en tiempo de ejecución, en donde se permite utilizar el lenguaje de programación en forma interactiva, sin necesidad de trabajo orientado a archivos de texto, como es el caso de Smalltalk.

Algunos IDE's soportan múltiples lenguajes, tales como Eclipse o NetBeans, ambas basadas en Java o MonoDevelop, basado en C#. El soporte para lenguajes alternativos es a menudo proporcionado por plugins, que les permite ser instalado en el mismo IDE al mismo tiempo. Por ejemplo, Eclipse y NetBeans tiene plugins para C / C + +, Ada, Perl, Python, Ruby y PHP, entre otros lenguajes. (20)

Entre los entornos integrados de desarrollo se encuentran los siguientes:

- Eclipse
- Zend Studio
- NetBeans
- VisualStudio.NET

1.5.4.1 Eclipse

Eclipse es principalmente una plataforma de programación, usada para crear entornos integrados de desarrollo (del Inglés IDE).

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto.

Está escrito principalmente en Java y se puede utilizar para desarrollar aplicaciones en Java y, por medio de diversos plug-ins de otros lenguajes de programación como Ada , C , C + + , COBOL , Perl , PHP , Python , Ruby (incluyendo Ruby on Rails marco), Scala , Clojure y Régimen . El IDE es a menudo llamado Eclipse ADT para Ada, CDT de Eclipse para C / C + +, Eclipse JDT para Java y Eclipse PDT para PHP.

Eclipse es un entorno de desarrollo integrado de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores. (21)

1.5.4.2 NetBeans

NetBeans es un entorno de desarrollo, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

La plataforma NetBeans permite que las aplicaciones que se desarrolló a partir de un conjunto modular de componentes de software llamados módulos. Las aplicaciones basadas en la plataforma NetBeans (incluyendo el IDE NetBeans) puede ser ampliado por los desarrolladores de terceros. (22)

Entre las características de la plataforma son:

Administración de usuarios de interfaz (por ejemplo, los menús y barras de herramientas)

Ajustes del usuario de gestión

- La gestión del almacenamiento (guardando y cargando cualquier tipo de datos)
- Ventana de gestión de marco Asistente (admite diálogos paso a paso)
- NetBeans Visual Biblioteca
- Herramientas de desarrollo integrado

1.5.4.3Visual Studio 2008

La herramienta Visual Studio 2008, 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 de sistema. También incluye avances clave para la colaboración eficiente entre los miembros del equipo.

1.5.4.4 Framework .net

Es un framework 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, 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.

En términos simples un framework es un conjunto de clases base que pueden ser reutilizadas para la construcción de un nuevo software.

Algunas Características

- 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. (23) (24)

1.5.5 Lenguaje de programación

Los lenguajes de programación son herramientas que nos permiten crear programas y software. Los lenguajes de programación facilitan la tarea de programación, ya que disponen de formas adecuadas que permiten ser leídas y escritas por personas, a su vez resultan independientes del modelo de computador a utilizar.

1.5.5.1 C#

C# (pronunciado si sharp en inglés) es un lenguaje de programación orientado al desarrollado y estandarizado por Microsoft como parte de su plataforma .NET, que más tarde fue aprobado como un estándar por la ECMA e ISO.

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. (25)

Algunas de las características que presenta:

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

- Es versionable.

- Eficiencia.

- Compatibilidad

1.5.5.2 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. 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.

1.5.6Metodología desarrollo de software

Conjunto de procedimientos, técnicas, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software

Una metodología puede seguir uno o varios modelos de ciclo de vida, es decir, el ciclo de vida indica qué es lo que hay que obtener a lo largo del desarrollo del proyecto pero no cómo hacerlo. La metodología indica cómo hay que obtener los distintos productos parciales y finales

Es un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información.(25)

1.5.6.2.1 RUP

El Proceso Unificado Racional (Rational Unified Process en inglés, habitualmente resumido como RUP) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

El RUP no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

Principales características:

✓ Dirigido por casos de uso: La razón de ser de un sistema de software es servir a usuarios ya sean humanos u otros sistemas; un caso de uso es una facilidad que el software debe proveer a sus usuarios. Los casos de uso reemplazan la antigua especificación funcional tradicional y constituyen la guía fundamental, establecida para las actividades a realizar durante todo el proceso de desarrollo incluyendo el diseño y

la implementación.Los casos de uso son los artefactos primarios para establecer el comportamiento deseado del sistema.

✓ Centrado en la arquitectura: La arquitectura involucra los elementos más significativos del sistema y está influenciada entre otros por plataformas de software, sistemas operativos manejadores de bases de datos, protocolos, consideraciones de desarrollo como sistemas heredados y requerimientos no funcionales. Los casos de uso guían el desarrollo de la arquitectura y la arquitectura se realimenta en los casos de uso, los dos juntos permiten conceptualizar, gestionar y desarrollar adecuadamente el software.La arquitectura es utilizada para conceptualizar, construir, administrar y evolucionar el sistema en desarrollo.

✓ Iterativo e Incremental: Para hacer más manejable un proyecto se recomienda dividirlo en ciclos. Para cada ciclo se establecen fases de referencia, cada una de las cuales debe ser considerada como un mini proyecto, cuyo núcleo fundamental está constituido por una o más iteraciones de las actividades principales básicas de cualquier proceso de desarrollo.Maneja una serie de entregas ejecutables.Integra continuamente la arquitectura para producir nuevas versiones mejoradas.

✓ Conceptualmente amplio y diverso

✓ Enfoque orientado a objetos

✓ En evolución continua

✓ Adaptable

✓ Repetible

✓ Permite mediciones

✓ Estimación de costos y tiempo, nivel de avance, etc.

✓Verificación de la calidad del software

✓Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo)

✓Pretende implementar las mejores prácticas en Ingeniería de Software

✓Administración de requisitos

✓Uso de arquitectura basada en componentes

✓Control de cambios

✓Modelado visual del software

✓Verificación de la calidad del software

RUP es un producto de (IBM). Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el

código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso). (26)

1.5.7 Especificación de los requisitos

La obtención de requisitos de software es una de las partes más importantes en desarrollo de un software, de la claridad con que estos logren ser obtenidos dependerá el éxito final del software. Las principales actividades que se desarrollan en esta etapa son:

- ✓ Obtención
- ✓ Elaboración
- ✓ Especificación
- ✓ Negociación
- ✓ Validación

1.5.7.1 Captura de los requisitos

En la captura de requisitos la obtención de información es de suma importancia para la especificación de requisitos. Por ello existen diversos métodos para que la misma se logre de forma satisfactoria. A continuación se muestran algunos de ellos.

- ✓ Entrevista
- ✓ Tormenta de Ideas
- ✓ Encuesta
- ✓ Cuestionario
- ✓ Inspección de registros
- ✓ Sistemas existentes

Entrevista: Las entrevistas se utilizan para recabar información en forma verbal, a través de preguntas. Quienes responden deben ser usuarios implicados directamente con el sistema, o usuarios que tengan altos conocimientos del sistema. La entrevista se puede realizar de forma individual o en grupos.

Tormentas de ideas o Brainstorming: Es una puesta en común de las ideas de los componentes de un grupo sobre un tema en estudio. La información que de su utilización se extrae es una lista de posibilidades que serán el punto de partida para continuar el análisis. La tormenta de ideas no proporciona respuestas a preguntas.

Encuestas: La encuesta, una de las técnicas de investigación social más difundidas, se basa en las declaraciones orales o escritas de una muestra de la población con el objeto de recabar información. Se puede basar en aspectos objetivos (hechos, hábitos de conducta, características personales) o subjetivos (opiniones o actitudes).

Cuestionarios: Consiste en la redacción de un documento con preguntas claras y para que las respuestas sean concretas.

Inspección de registros: El término "registro" se refiere a los manuales escritos sobre políticas, regulaciones y procedimientos de operaciones estándar que la mayoría de las empresas mantienen como guía para gerentes y empleados. Los manuales que documentan o describen las operaciones para los procesos de datos existentes, o sistemas de información que entran dentro del área de investigación, también proporcionan una visión sobre la forma en la que el negocio debería conducirse.

Sistemas existentes: consiste en analizar los sistemas ya desarrollados que estén relacionados con el sistema que va a ser construido. Analizándose las interfaces de usuarios y las entradas y salidas que el mismo produce.

1.5.7.2 Técnicas de validación de requisitos

Los requisitos una vez definidos necesitan ser validados. La validación de requisitos tiene como misión demostrar que la definición de los requisitos define realmente el sistema que el usuario necesita o el cliente desea (Lowe & Hall, 1999). Es necesario asegurar que el análisis realizado y los resultados obtenidos de la etapa de definición de los requisitos son correctos. Pocas son las propuestas existentes que ofrecen técnicas para la realización de la validación y muchas de ellas consisten en revisar los modelos obtenidos en la definición de requisitos con el usuario para detectar errores o inconsistencias.

Aun así, existen algunas técnicas que pueden aplicarse para ello:

- ✓ **Reviews o Walk-throughs:** Está técnica consiste en la lectura y corrección de la completa documentación o modelado de la definición de requisitos. Con ello solamente se puede validar la correcta interpretación de la información transmitida. Más difícil es verificar consistencia de la documentación o información faltante.
- ✓ **Listas de chequeo:** Son frecuentemente usadas en inspecciones o revisiones de artefactos generados en el proceso de producción de software; son listas de aspectos que deben ser completados o verificados.
- ✓ **Auditorías:** La revisión de la documentación con esta técnica consiste en un chequeo de los resultados contra una checklist **(Listas de Chequeo)** predefinida o definida a comienzos del proceso, es decir, sólo una muestra es revisada.
- ✓ **Prototipos:** Algunas propuestas se basan en obtener de la definición de requisitos prototipos que, sin tener la totalidad de la funcionalidad del sistema, permitan al usuario hacerse una idea de la estructura de la interfaz del sistema con el usuario (Olsina, 1999).
- ✓ **Matrices de trazabilidad:** Esta técnica consiste en marcar los objetivos del sistema y chequearlos contra los requisitos del mismo. Es necesario ir viendo

qué objetivos cubre cada requisito, de esta forma, se podrán detectar inconsistencias u objetivos no cubiertos.

La validación de requisitos es una actividad muy importante, pues un levantamiento de requisitos con errores que no se detecten a tiempo, además de no conducir a resultados inesperados provoca costos excesivos y gran pérdida de tiempo.

1.5.8 Librería

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. Algunos programas ejecutables pueden ser a la vez programas independientes y bibliotecas, pero la mayoría de éstas no son ejecutables. Ejecutables y bibliotecas hacen referencias (llamadas enlaces) entre sí a través de un proceso conocido como enlace, que por lo general es realizado por un software denominado enlazador.

1.5.8.1 OpenCV

OpenCV (Open source computer visión library) es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Esta librería proporciona un alto nivel funciones para el procesado de imágenes. Estas librerías permiten a los programadores crear aplicaciones poderosas en el dominio de la visión digital.

OpenCV es multiplataforma. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de Visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estéreo y visión robótica. OpenCV ofrece muchos tipos de datos de alto-nivel como juegos, árboles, gráficos, matrices, etc. (27) (28)

Principales Características

- ✓ Manejo de estructuras básicas.
- ✓ Procesamiento de imágenes.
- ✓ Análisis estructural.
- ✓ Análisis de movimiento y seguimiento de objetos.
- ✓ Reconocimiento de objetos.
- ✓ Calibración de cámara y reconstrucción de escenas 3D.
- ✓ Manejo de escenas de vídeo.
- ✓ Construcción de interfaces gráficas de usuario (GUI).
- ✓ OpenCV es multi-plataforma de nivel medio-alto de la API que consiste en unos cuantos cientos (> 300) C funciones. No depende de librerías numéricas externas, aunque se puede hacer uso de algunos de ellas en tiempo de ejecución, si están disponibles.

- ✓ OpenCV es gratuita tanto para uso no comercial y comercial, OpenCV proporciona transparente para interfaz de usuario para Intel ® Integrated Performance Primitives (IPP) (sólo ippcv por ahora). Es decir, se carga automáticamente IPP bibliotecas optimizado para procesadores específicos.

1.5.8.1.1 Funcionalidades

Dentro de las funcionalidades con que cuenta OpenCV para el trabajo con imágenes en el desarrollo del componente de validación y post- procesamiento de imágenes faciales para uso en documentos oficiales serán utilizadas las siguientes.

Funcionalidad	Descripción
IplImage* cvLoadImage(const char* filename, int iscolor=CV_LOAD_IMAGE_COLOR)	La función CvLoadImage cargue una imagen del archivo especificado y retorna el indicador a la imagen cargada.
cvCreateImage(cvSize(original->width,original height),IPL_DEPTH_8U,original->nChannels)	cvCreateImage puede crear sólo las imágenes interfoliadas.
void cvShowImage(const char* name, const CvArr* image)	La función cvShowImage muestra la imagen en la ventana especificada. Si la ventana era creada con CV_WINDOW_AUTOSIZE entonces la imagen es mostrada con su tamaño original, de otra manera la imagen se ajusta a la función.
int cvWaitKey(int delay=0)	Delay demora en milisegundos. La función CvWaitKey espera para el evento clave infinitamente (delay< =0) o para los milisegundos de "demora". Retorne el código de la llave o -1.
cvDestroyAllWindows();	Esta función elimina todas las ventanas gráficas que hayan sido creadas previamente
void cvReleaseImage(& CvArr* img)	Esta función se encarga de liberar el espacio de memoria que ha sido asignado a una estructura

	CvArr*. Posee un único parámetro: img: es el nombre de la imagen que se desea liberar
CvMat* cvCreateMat(int rows, int cols, int type);	Esta función se encarga de crear al encabezado de la imagen y de ubicar sus datos. Exponemos su estructura: rows: número de filas de la matriz cols: número de columnas de la matriz type: tipo de los elementos de las matrices
void cvWarpAffine(const CvArr* src, CvArr* dst, const CvMat*map_matrix, int flags=CV_INTER_LINEAR+CV_WARP_FILL_OUTLIERS, CvScalar fillval=cvScalarAll(0));	La función genera una imagen de salida a la que se la aplica la transformación correspondiente.
cvCloneImage(img);	Hace una copia de la imagen.
cvResize(const CvArr* src, CvArr* dst, int interpolation=CV_INTER_LINEAR);	Función para escalar una imagen Siendo: – src: Imagen fuente. – dst: Imagen destino. – interpolation: Tipo de interpolación. La interpolación puede ser de los siguientes modos: – CV_INTER_NN : Interpolación tipo vecino más próximo – CV_INTER_LINEAR: Interpolación bilineal (es la que se utiliza por defecto) – CV_INTER_AREA: Se utiliza la relación de píxel y área para asignar valores. En el caso de ampliar la imagen actúa análogamente al método CV_INTER_NN. – CV_INTER_CUBIC: Interpolación bicúbica.

void cvmSet (CvMat* mat, int row, int col, double value)	Esta función guarda un elemento en una matriz de un solo canal en punto flotante: mat:La matriz de entrada row: El índice de la fila col: El índice de la columna value: El nuevo valor del elemento de la matriz
cvCreateMemStorage(0);	Crea una memoria de almacenamiento y devuelve el puntero a él. Inicialmente, el de almacenamiento está vacío.
cvHaarDetectObjects(img,pCascade,memory,1.1,3,CV_HAAR_DO_CANNY_PRUNING,cvSize(0,0));	La función cvHaarDetectObjects encuentra en regiones rectangulares de la imagen ya que es probable que contenga objetos de la cascada que ha sido entrenado para esas regiones y devuelve como una secuencia de rectángulos. La función explora la imagen varias veces a diferentes escalas (ver cvSetImagesForHaarClassifierCascade.Cada vez que lo considere regiones superpuestas en la imagen y se aplica a los clasificadores a las regiones utilizando cvRunHaarClassifierCascade.
cvGetSeqElem(cantFace,i)	El cvGetSeqElem función encuentra el elemento con el índice dado en la secuencia y devuelve el puntero a él. Además, la función puede devolver el puntero al bloque de la secuencia que contiene el elemento. Si el elemento no se encuentra, la función devuelve 0.
cvRectangle(img,ini,fin,CV_RGB(0,255,0),3,4,0);	Dibuja un rectángulo con dos esquinas opuestas.

void cvCvtColor(const CvArr* src, CvArr* dst, int code);	Esta función da la posibilidad de transformar una imagen RGB en otra en escala de grises Siendo: – src: La imagen origen codificada en 8 bits o en punto flotante – dst: La imagen destino codificada en 8 bits o en punto flotante – code: Define la operación de conversión a llevar a cabo. Esta operación se define utilizando las constantes
cvSetImageROI(cloneImag,*rect);	La función cvSetImageROI establece la imagen de retorno de la inversión a un rectángulo dado. Si es NULL retorno de la inversión y el valor del parámetro rect no es igual a toda la imagen, retorno de la inversión se destina.
cvResetImageROI(cloneImag);	Devuelve coordenadas de la imagen como altura y ancho
void cvNamedWindow(char name, int type);	Esta función crea una ventana gráfica. parámetros: name: cadena de caracteres que sirve como nombre de la ventana, type: formato de tamaño de la ventana: Utilizaremos CV_WINDOW_AUTOSIZE, o simplemente pondremos un 1 para seleccionar esta opción.
Cv_rgb2lab	Convierte de espacio RGB a espacio de colores lab que almacena en al componente L la iluminación, y en los restantes los colores.

Tabla 1 .Funcionalidades de Open CV.

1.6 Fundamentación de las herramientas, metodologías y tecnologías a utilizar

Para el futuro desarrollo del componente de validación y post- procesamiento de imágenes faciales para uso en documentos oficiales se tiene como propuesta inicial utilizar: como lenguaje de modelado el **Lenguaje Unificado de Modelado UML** que

es el más conocido y utilizado en la actualidad. Esta consolidado como el lenguaje estándar en el análisis y diseño de sistemas de cómputo. Se escoge UML porque: Permite modelar sistemas utilizando técnicas orientadas a objetos. Permite especificar todas las decisiones de análisis y diseño, construyéndose así modelos, precisos, no ambiguos y completos. Puede conectarse con lenguajes de programación (ingeniería directa e inversa). Permite documentar todos los artefactos de un proceso de desarrollo (requisitos, arquitectura, pruebas, versiones). Existe un equilibrio entre expresividad y simplicidad, pues no es difícil de aprender ni de utilizar.

Como herramienta de modelado **Visual Paradigm** quees una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. Este software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste.

Como entorno integrado de desarrollo IDE **Visual Studio** 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 de sistema. También incluye avances clave para la colaboración eficiente entre los miembros del equipo. El **Lenguaje de Programación C++** que es un lenguaje de programación de propósito general que ofrece economía sintáctica, control de flujo y estructuras sencillas y un buen conjunto de operadores. En poco tiempo, un programador puede utilizar la totalidad del lenguaje.

La **librería OpenCV** (Open source computer vision library) es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Esta librería proporciona un alto nivel de funciones para el procesado de imágenes. Estas librerías permiten a los programadores crear aplicaciones poderosas en el dominio de la visión digital. OpenCV es multiplataforma. Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de Visión, como reconocimiento de objetos (reconocimiento facial), calibración de cámaras, visión estéreo y visión robótica. OpenCV ofrece muchos tipos de datos de alto-nivel como juegos, árboles, gráficos, matrices.

Como metodología usaremos **RUP** que no es un sistema con pasos firmemente establecidos, sino un conjunto de metodologías adaptables al contexto y necesidades de cada organización. Las **Razones para usar RUP** son**:** Provee un entorno de proceso de desarrollo configurable, basado en estándares. Permite tener claro y

accesible el proceso de desarrollo que se sigue. Permite ser configurado a las necesidades de la organización y del proyecto. Provee a cada participante con la parte del proceso que le compete directamente, filtrando el resto.

1.7 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. Se describe la metodología que regirá el desarrollo del componente de validación y post- procesamiento de imágenes faciales para uso en documentos oficiales y se abordaron las tecnologías y herramientas seleccionadas para el desarrollo del componente.

Se puede concluir que para dar solución a la problemática planteada es necesario desarrollar el componente que valide y procese una imagen para uso en documentos oficiales. Para ello, la selección de un lenguaje de programación como C++, con el uso de una librería que lo hace más potente. La herramienta CASE Visual Paradigm, el IDE Visual Studio 2008, permitirán la agilización del desarrollo de la investigación. RUP será de suma importancia para guiar el proceso de descubrimiento de conocimiento del desarrollo del componente respectivamente.

Capítulo II Análisis y Diseño

2.1 Introducción

La solución a desarrollar, debe ser fruto de un correcto análisis y una amplia comprensión de todos los elementos que se relacionan en correspondencia con el tema de validación y normalización de una imagen para su uso en documentos oficiales, profundizándose en el estudio de las características que posibilitan desarrollar los mismos.

En este capítulo se interpretan las necesidades del sistema especificándolas mediante los requerimientos funcionales y los no funcionales. Además, se hace un estudio del negocio en que se enmarca el problema concluyendo que se debe realizar una modelación del dominio, identificando para esto las entidades principales que se tendrán y las relaciones entre ellas.

El capítulo al mismo tiempo expone el diagrama de casos de uso del sistema con sus respectivas descripciones, el diseño de la solución; los diagramas de clases de análisis y del diseño y sus respectivos diagramas de secuencias.

El objetivo principal del análisis y diseño del sistema es transformar los requerimientos en especificaciones que describan cómo implementar el sistema. El análisis fundamentalmente consiste en obtener una visión que se preocupa de ver qué hace el sistema de software a desarrollar, por tal motivo este se interesa en los requerimientos funcionales. Por otro lado, el diseño es un refinamiento que toma en cuenta los requerimientos no funcionales, por lo cual se centra en como el sistema cumple sus objetivos.

2.2 Propuesta de solución

Como propuesta de solución para el componente se define la creación de una librería (dll) haciendo uso del lenguaje C++ en la cual un usuario del sistema va interactuar con la misma para realizar los procesos de validación y normalización de imágenes. Para el desarrollo de dicho componente se hará uso del patrón tuberías y filtros, así como las potencialidades de la librería OpenCV. La siguiente figura muestra la propuesta solución definida.

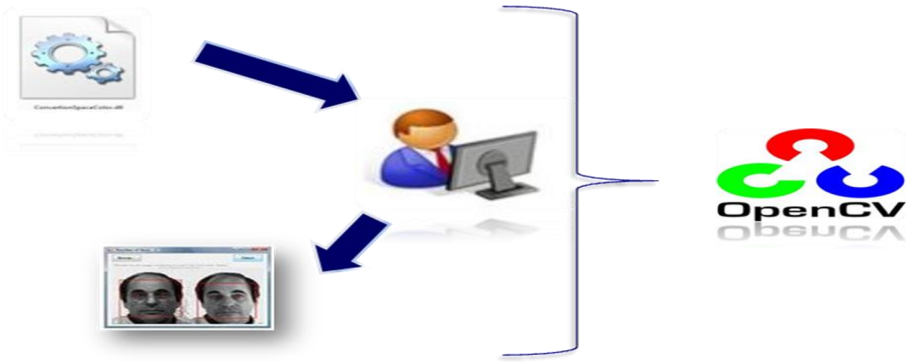

Figura 9.Propuesta de Solución.

2.3 Modelo de dominio

Actualmente ciertas organizaciones no cuentan con una infraestructura dedicada a la monitorización de los procesos de negocio, lo cual no le permite ser parte de una solución para validar imágenes para su uso en documentos oficiales, por tal motivo se determina que no se podían identificar procesos del negocio que enmarcaran el problema.

Sin embargo, los conceptos tecnológicos que definen correctamente cada eslabón de la solución se fueron observando, determinándose entonces la creación de un modelo de dominio el cual deja bien claro cómo funciona el entorno en el cual está enmarcado el problema. El modelo de dominio es una representación visual estática del entorno real del componente. Es decir, un diagrama con los objetos que existen en el mundo real relacionados con el componente 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 trabajara durante el desarrollo del componente. El diagrama de dominio se representa a través de diagramas de clases, lo que más simplificados. No son objetos software, sino un “diccionario visual” de conceptos del dominio. (29)

A continuación, se muestra el modelo de dominio que conceptualiza los elementos principales y sus relaciones.

- ✓ Imagen: Imagen facial que se captura mediante un dispositivo o que esta previamente almacenada.
- ✓ Componente: donde se realizara el proceso de validación y normalización de las imágenes faciales.
- ✓ Documentos oficiales: donde serán utilizadas las imágenes faciales una vez culminado el proceso de validación y normalización del componente.

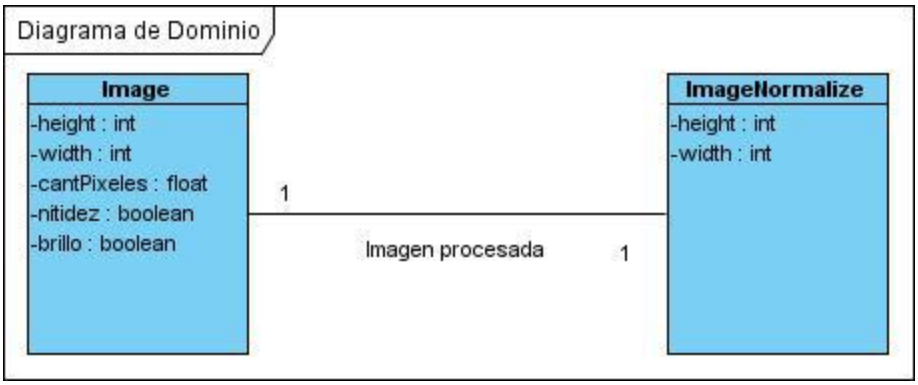

Figura 10.Diagrama del modelo de dominio.

2.4 Especificación de los requisitos de software

La obtención de los requerimientos es un paso muy importante para el posterior desarrollo de las siguientes etapas del desarrollo del software, pues un error en estas fases iníciales puede dar al traste con un sistema que no cumpla las expectativas de los usuarios y difícilmente aporte valor agregado al negocio para el que debe ser concebido. La IEEE Standard Glossary of Software Engineering Terminology define un requerimiento como Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo. El flujo de trabajo de requisitos tiene como objetivos definir el ámbito del sistema, definir una interfaz de usuarios, enfocada a las necesidades y metas del usuario, establecer y mantener un acuerdo entre clientes y otros involucrados sobre qué debería hacer el sistema, proveer a los desarrolladores de un mejor entendimiento de los requisitos del sistema. En este flujo de trabajos se desarrollan principalmente las actividades de identificar y clasificar requisitos, encontrar actores y casos de usos, priorizar y detallar los casos de uso. (30)

Las técnicas seleccionadas para la captura de requisitos son los sistemas existentes, porque observando sus interfaces se puede identificar la información que realmente es importante y la salida de los datos, la otra técnica utilizada es la tormenta de ideas por su sencillez y fácil uso.

2.4.1 Requerimientos funcionales

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir es decir: especifican acciones que el sistema debe ser capaz de realizar, sin tomar en consideración ningún tipo de restricción física. En la realización de los casos de uso del negocio, se obtienen las actividades que serán objeto de automatización. Estas actividades no son exactamente los requisitos funcionales, pero sí son el punto de partida para identificar qué debe hacer el sistema.

Los requisitos funcionales se mantienen invariables sin importar con que propiedades o cualidades se relacionen. (31)

A continuación, se muestran las funcionalidades que el sistema debe cumplir:

RF1. Cargar imagen.

RF2. Validar imagen.

RF2.1. Detectar imagen.

RF2.2. Validar nitidez de la imagen.

RF2.3. Verificar estado de los ojos.

RF2.4. Verificar distancia entre los ojos.

RF2.5. Validar buena pose.

RF2.6. Validar iluminación.

RF2.7. Validar balance de colores.

RF2.8. Verificar mirada al frente.

RF2.9. Validar expresión de la boca.

RF2.10. Verificar presencia de espejuelos, barba o bigote.

RF2.11. Validar rostro centrado.

RF2.12. Verificar fondo uniforme.

RF3. Normalizar imagen.

RF3.1. Obtener calidad del retrato.

RF3.2. Obtener pose.

RF3.3. Obtener profundidad del campo.

RF3.4. Normalizar orientación de la pose.

RF3.5. Normalizar tamaño del rostro.

RF3.6. Normalizar centrado de la imagen.

2.4.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable.

Los requerimientos no funcionales forman una parte significativa de la especificación. Son importantes para que clientes y usuarios puedan valorar las características no funcionales del producto, pues si se conoce que el mismo cumple con toda la funcionalidad requerida, las propiedades no funcionales, como cuan usable, seguro, conveniente y agradable, pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación. (32)

Los requisitos no funcionales se clasifican en:

- ✓ Requerimientos de Software.
- ✓ Requerimientos de Hardware.
- ✓ Restricciones en el diseño y la implementación.
    - Estándares requeridos.
    - Lenguajes de programación a ser usados para la implementación.
    - Uso obligatorio de ciertas herramientas de desarrollo.
    - Restricciones en la arquitectura o el diseño.
    - Bibliotecas de clases.
- ✓ Requerimientos de apariencia o interfaz externa.
- ✓ Requerimientos de seguridad.
    - Confidencialidad.
    - Integridad.
    - Disponibilidad.
- ✓ Requerimientos de usabilidad.
- ✓ Requerimientos de soporte.
- ✓ Requerimientos legales.

Para el desarrollo del componente los requisitos no funcionales a usar son:

2.4.2.1 RNF1.Software

- ✓ El sistema será usado bajo los Sistemas Operativos Windows.
- ✓ Framework .NET.

2.4.2.2 RNF2. Requisitos de Hardware

- ✓ Requerimientos mínimos de hardware.
- ✓ PC Intel Pentium 4 o superior.
- ✓ CPU 3GHZ o superior.
- ✓ HDD SCCI 160 GB o superior.
- ✓ 1GB de Memoria RAM o superior.

2.4.2.3 RNF3. Restricciones en el diseño y la implementación

- ✓ Estándares requeridos por la OACI.
- ✓ Lenguaje de programación: Visual C++.
- ✓ Framework de desarrollo que se utilizará es: .NET Framework.
- ✓ IDE: Visual Studio 2008.
- ✓ Para el Modelado de UML se utilizará: Visual Paradigm 6.4.
- ✓ Biblioteca de clases: Open CV.

2.4.2.4 RNF4. Rendimiento

- ✓ La aplicación está concebida para un ambiente donde la misma se integre a un sistema de identificación, por lo que los tiempos de respuestas deben ser rápidos, así como la velocidad de procesamiento de información.
- ✓ La aplicación requiere de un buen rendimiento en máquinas de pocos recursos de Hardware.

2.4.2.5 RNF5. Requisitos de seguridad

- ✓ Disponibilidad: Solo los usuarios autorizados se les garantizará el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.
- ✓ Confiabilidad: El sistema propuesto debe ser capaz de adaptarse con el fin de alcanzar un equilibrio interno frente a los cambios externos del entorno.

2.4.2.6 RNF6. Apariencia o 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.

2.4.2.6 RNF7.Legales

- ✓ Al componente de validación y post- procesamiento de imágenes faciales para uso en documentos oficiales solo tendrá acceso el proyecto de Biometría del CISED perteneciente a la Universidad de Ciencias Informáticas.

2.4.3 Técnicas de revisión de requisitos

Las técnicas de revisión de requisitos utilizadas es la revisión ver anexo 2 y la matriz de trazabilidad ver anexo 3.

2.5 Modelo de casos de uso del sistema

El modelo de casos de uso ayuda al cliente, a los usuarios y a los desarrolladores a llegar a un acuerdo sobre cómo utilizar el sistema. La mayoría de los sistemas tienen muchos tipos de usuarios. Cada tipo de usuarios se representa mediante un actor. Los actores utilizan el sistema al interactuar con los casos de uso, Todos los actores y casos de uso del sistema forman un modelo de casos de usos .Un diagrama de casos

de usos describe parte del modelo de casos de uso y muestra un conjunto de casos de uso y actores con una asociación entre cada actor/caso de uso que interactúa

2.5.1Actores del sistema

Los actores del sistema no son parte de él. Pueden intercambiar información con él. Pueden ser un recipiente pasivo de información. Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado.

A continuación se muestra el único actor identificado.

Actores	Descripción
Usuario del Sistema	Representa a la persona encargada de cargar la imagen para que se valide y luego sea normalizada.

Tabla 2. Actores del Sistema.

2.5.2Casos de uso del sistema

Los casos de uso están diseñados para cumplir los deseos del usuario cuando utiliza el sistema. El modelo de casos de uso captura todos los requisitos funcionales del sistema. Definiremos un caso de como:

Un caso de uso especifica una secuencia de acciones, incluyendo variantes, que el sistema puede llevar a cabo, y que producen un resultado observable para un actor concreto.

Los caso de usos son identificados examinando cómo lo usuarios necesitan utilizar el sistema para realizar su trabajo.

Casos de Usos	Prioridad
Cargar Imagen	Critico
Validar Imagen	Critico
Normalizar Imagen	Critico

Tabla 3.Casos de Uso.

2.5.3Diagrama de casos de uso del sistema

Un diagrama de casos de uso del sistema representa gráficamente a los procesos y su interacción con los actores.

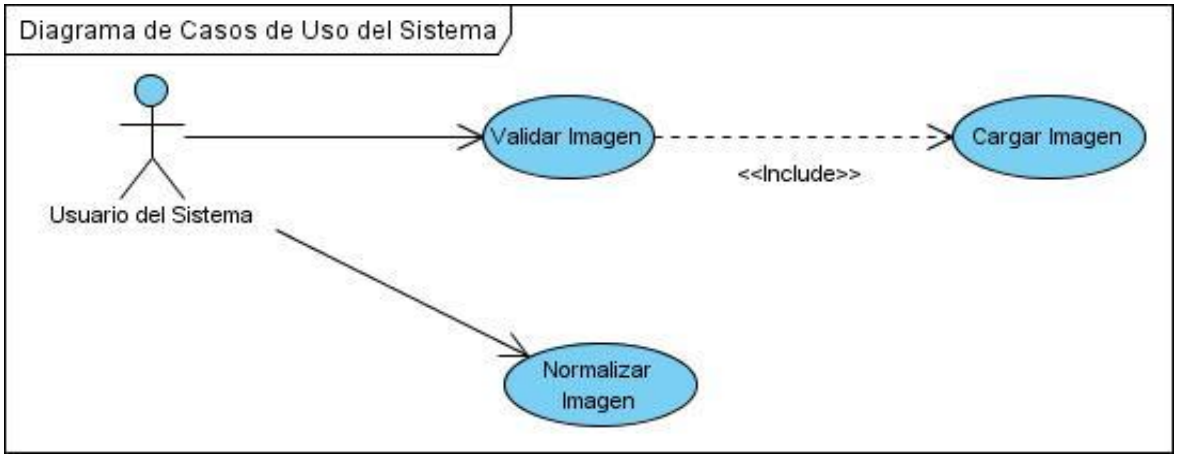

Figura 11.Diagrama de caso de uso del sistema.

2.5.4 Descripción de los casos de uso del sistema

Nombre del caso de uso: Cargar imagen(caso de uso incluido)	
Actor:	Usuario del Sistema.
Propósito:	Cargar una imagen para ser validada.
Resumen:	El caso de uso se inicia cuando el usuario del sistema carga la imagen.
Referencia:	RF1.
Precondiciones:	Debe haberse solicitado a la acción de validar imagen.
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
El usuario del sistema carga la imagen a ser procesada.	El sistema muestra un botón para cargar un conjunto de imágenes para ser validadas, según en la dirección que se encuentre las mismas.
El usuario del sistema espera la respuesta del sistema.	El sistema muestra la imagen validada.

Tabla 4.CU Cargar Imagen.

Nombre del caso de uso: Validar Imagen.	
Actor:	
Propósito:	Validar la imagen.

Resumen:	El caso de uso se inicia cuando el sistema requiere validar la imagen.
Referencia:	RF1
Precondiciones:	Debe haberse cargado una imagen con anterioridad.
"Secciones"	
Sección 1 Detectar rostro.	
	El sistema realiza la detección del rostro mediante una localización facial de la detección de las características faciales, como los ojos, la nariz, la boca, entre otras características físicas.
Sección 2 Validar Nitidez de la Imagen.	
	El sistema verifica que la imagen no tenga arrugas, ni manchas de tinta, que contenga el brillo y contraste adecuado.
Sección 3 Estado de los Ojos.	
	El sistema verifica que los ojos estén con la mirada en una misma dirección, no mirando al lado y que estos no presenten color rojizo producto a los efectos infrarrojos que puede suceder cuando la iluminación de la imagen no es la correcta, y que los ojos estén visibles y abiertos.
Sección 4 Verificar distancia entre los Ojos.	
	El sistema verifica que los ojos estén correctamente alineados con respecto a la nariz, orejas y boca.
Sección 5 Validar Buena Pose.	
	El sistema verifica que trazada una línea horizontal imaginaria trazada al centro de los ojos resulte paralela al borde superior de la imagen, verificando así de esta manera una buena posición del titular con respecto a la cámara, se requiere que él titular no presente silla ni respaldo en la imagen.
Sección 6 Validar Iluminación.	

	El sistema verifica que la iluminación sea uniforme, sin sombras, ni reflejos, ni cabello sobre el rostro del titular.
Sección 7 Validar Balance de Colores.	
	El sistema verifica que el retrato tenga los tonos propios naturales de colores, y una alta resolución para evitar la presencia de pixeles en la imagen.
Sección 8 Mirada al Frente.	
	El sistema verifica que el titular tenga la mirada frente a la cámara, no mirando por encima del hombre. Que no exista una inclinación del titular en la imagen.
Sección 9 Validar expresión de la boca.	
	El sistema verifica que el titular tenga la mirada una mirada neutra y la boca cerrada correctamente, no se acepta inclinación de esta.
Sección 10 Verificar presencia de gafas y tocados.	
	El sistema verifica que el titular no tenga reflejos en las gafas, ni gafas oscuras, y que estas no oculten parcialmente parte del rostro. También verificara que el titular no tenga presencia de gorra, o tocado.
Sección 11 Validar Rostro centrado.	
	El sistema verifica que el titular no esté a un lado de la imagen, y que no esté ni muy lejos ni muy cerca con respecto a la cámara**.**
Sección 12 Verificar fondo uniforme.	
	El sistema verifica el fondo de la imagen este uniforme y no sea colorido, ni presente objetos o que la imagen contenga borde alguno que resalte.
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
	El sistema muestra una interfaz con la imagen

	validada.
Flujo Alterno	
	En caso que la imagen no cumpla con algunos de los parámetros descritos en las secciones anteriores, el usuario deberá seleccionar otra imagen a validar.
Pos condiciones:	Se validó correctamente la imagen.

Tabla 5.CU Validar Imagen.

Nombre del caso de uso: Normalizar Imagen	
Actor:	Usuario de Sistema.
Propósito:	Normalizar la imagen luego de concluida su validación.
Resumen:	El caso de uso se inicia cuando el sistema ha validado correctamente la imagen y es necesario normalizarla.
Referencia:	RF3
Precondiciones:	Debe haberse validado correctamente la imagen.
"Secciones"	
Sección 1 Obtener buena calidad del retrato	
	El sistema convierte la imagen a una dimensión no superior a 35 mn x 45 mn ni inferior a 32 mn x 26 mn (ancho y alto), y el retrato tendrá menos de 6 meses de captado.
Sección 2 Obtener Pose	
	El sistema mostrará el rostro completo del titular y los ojos claramente visibles.
Sección 3 Obtener Profundidad del campo	
	El sistema mostrará el rostro completo que estará desde la coronilla hasta la barbilla y desde la nariz hasta las orejas.
Sección 4 Normalizar orientación de la pose	
	El sistema mostrará la orientación coronilla-

	barbilla que cubrirá la dimensión más larga definida para la Zona V.
Sección 5 Normalizar tamaño del rostro	
	El sistema mostrará la porción coronilla-barbilla de la pose frontal del rostro completo que ocupará entre el 70% y 80% de la dimensión más larga que se define en la Zona V.
Sección 6 Normalizar centrado de la imagen	
	El sistema mostrará la pose frontal de todo el rostro centrada en la Zona V.
Flujo Normal de Eventos	
Acción del Actor	**Respuesta del Sistema**
	El sistema convierte la imagen a sus estándares requeridos ya definidos, para ser procesada y lo almacena.

Tabla 6.CU Normalizar Imagen.

2.6 Análisis

En el flujo de trabajo de análisis se trabaja sobre los aspectos internos del sistema a desarrollar, se pueden estructurar los requisitos de manera que faciliten su comprensión, su preparación, su modificación y en general su mantenimiento.

2.6.1 Modelo de análisis

En el modelo de análisis se obtiene un mayor poder expresivo y una mayor formalización y se puede considerar como una primera aproximación al modelo de diseño. En el modelo de análisis no se tiene en cuenta el lenguaje de programación, ni la plataforma en la que se desarrollara la aplicación, ya que el objetivo del análisis es comprender perfectamente los requisitos del software y no precisar cómo se implementará la solución. El modelo de análisis debe tener un alto nivel de abstracción y minimizar el acoplamiento del sistema. Para la construcción del modelo de análisis se identificaron las clases que describen la realización de los casos de uso, que pueden ser de tres tipos fundamentales: clases interfaz, clases entidad y clases controladoras.

- ✓ Las clases Interfaz se utilizan para modelar la interacción entre el sistema y sus actores, lo que clarifican y reúnen los requisitos en los límites del sistema.

Representan abstracciones de ventanas, formularios, paneles, interfaces de comunicaciones, interfaces de impresoras, sensores y terminales.

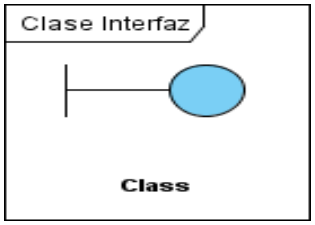

Figura 12.Estereotipo clase interfaz.

- ✓ Las clases entidad se utilizan para modelar la información que posee una vida larga y que es a menudo persistente. Modelan la información y el comportamiento asociado de algún fenómeno o concepto y reflejan la información de modo que beneficia a los desarrolladores al diseñar e implementar el sistema, incluyendo su soporte de persistencia.

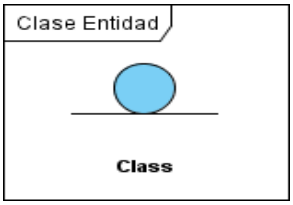

Figura 13.Estereotipo clase entidad.

- ✓ Las clases controladoras representan la coordinación, secuencia, transacciones, y control de otros objetos, y se usan con frecuencia para encapsular el control de un caso de uso en concreto. Se utilizan para representar derivaciones y cálculos complejos.

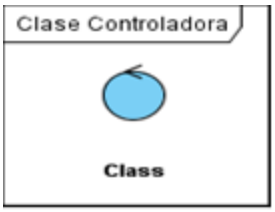

Figura 14.Estereotipo clase controladora.

2.6.2 Diagrama de clases del análisis

Un Diagrama de clases del análisis es un artefacto en el que se representan los conceptos en un dominio del problema. Representa el funcionamiento del mundo real, no de la implementación automatizada del mismo.

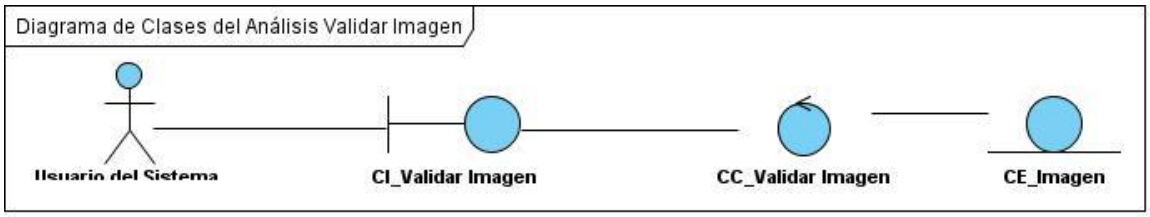

Figura 15.Diagrama de Clases del Análisis Validar Imagen.

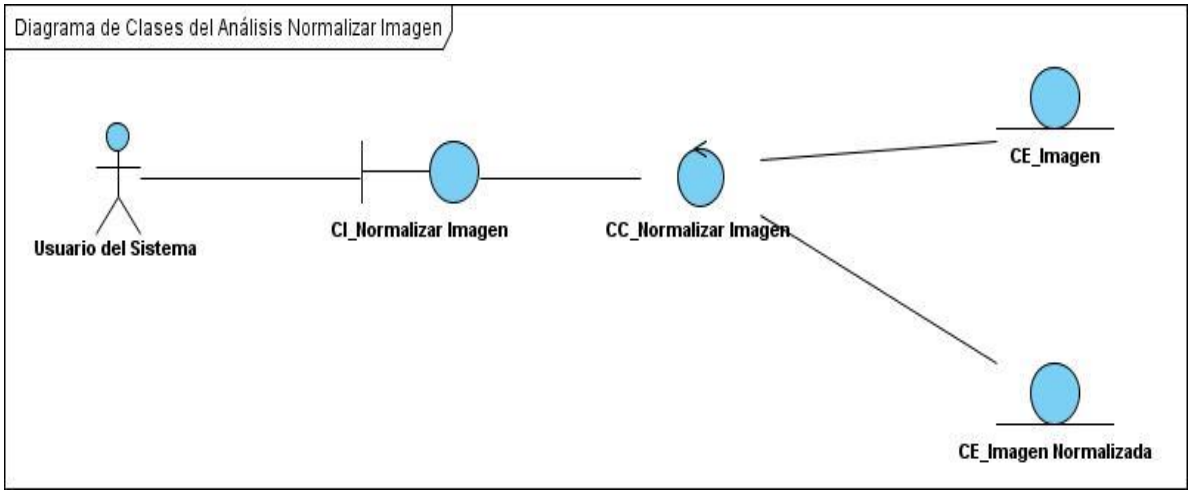

Figura 16.Diagrama de Clases del Análisis Normalizar.

2.7 Diagramas de interacción

Los diagramas de interacción se utilizan para modelar los aspectos dinámicos de un sistema, lo que conlleva modelar instancias concretas o prototípicas de clases interfaces, componentes y nodos, junto con los mensajes enviados entre ellos, todo en el contexto de un escenario que ilustra un comportamiento. En el contexto de las clases describen la forma en que grupos de objetos colaboran para proveer un comportamiento. Se trata de un término genérico que se aplica a varios tipos de diagramas que hacen hincapié en las interacciones entre objetos. Los diagramas interacción pueden ser de dos tipos:

- ✓ Diagramas de Colaboración.
- ✓ Diagramas de Secuencia.

A continuación se muestram los de diagramas colaboracion en epigrafes mas adelante se muestra el diagrama de secuencia.

2.7.1 Diagrama de colaboración

Un Diagrama de Colaboración muestra una interacción organizada basándose en los objetos que toman parte en la interacción y los enlaces entre los mismos (en cuanto a la interacción se refiere).

Una colaboración modela los objetos y los enlaces significativos dentro de una interacción. Los objetos y los enlaces son significativos solamente en el contexto proporcionado por la interacción. Un rol describe un objeto, y un rol en la asociación describe un enlace dentro de una colaboración.

Un diagrama de colaboración muestra los roles en la interacción en una disposición geométrica. Los mensajes se muestran como flechas, ligadas a las líneas de la relación, que conectan a los roles. La secuencia de mensajes, se indica con los números secuenciales que preceden a las descripciones del mensaje. Un uso de un diagrama de colaboración es mostrar la implementación de una operación

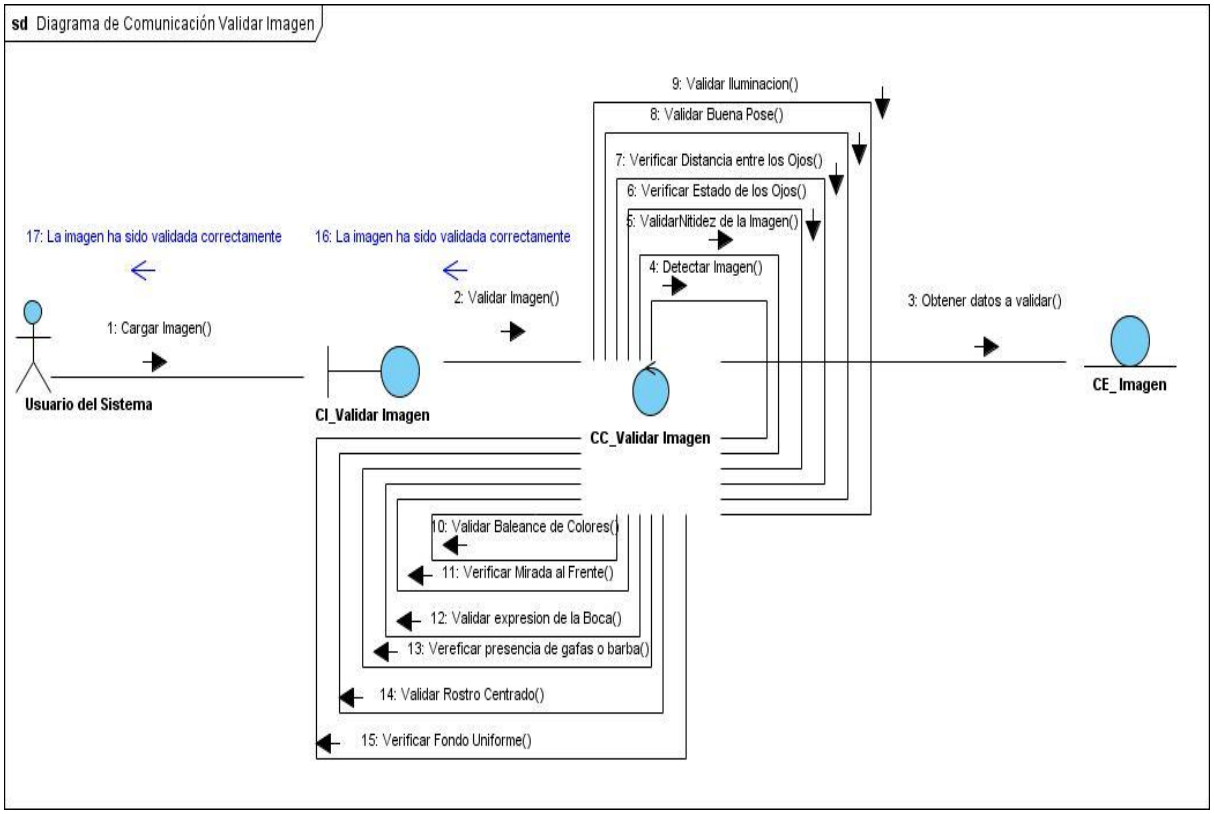

Figura 17.Diagrama de Colaboración Validar Imagen.

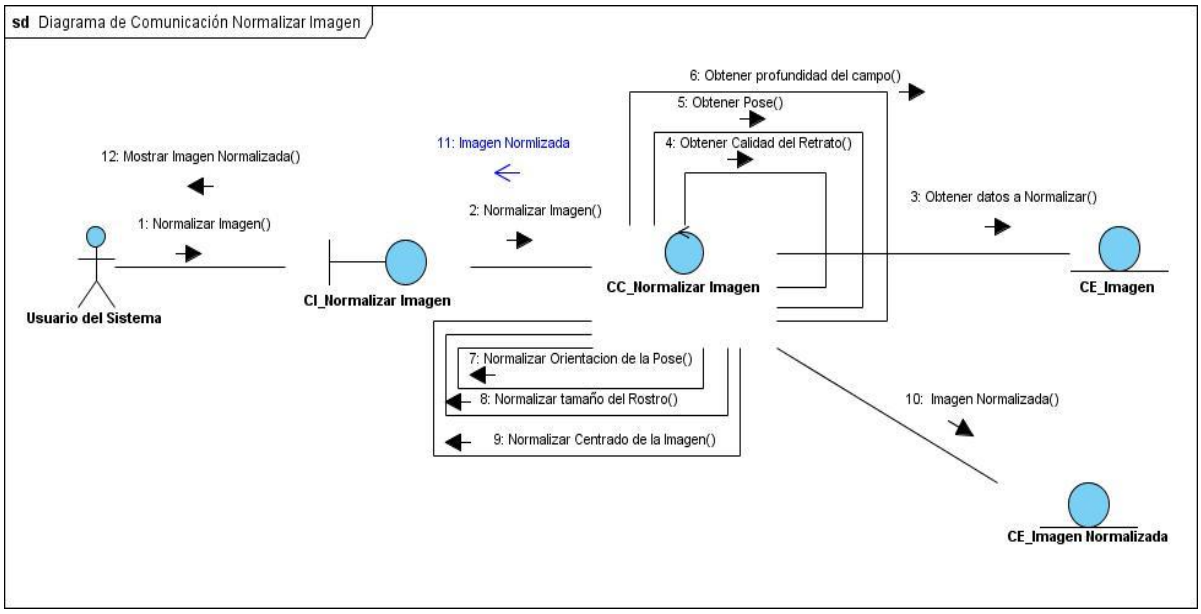

Figura 18.Diagrama de Colaboración Normalizar Imagen.

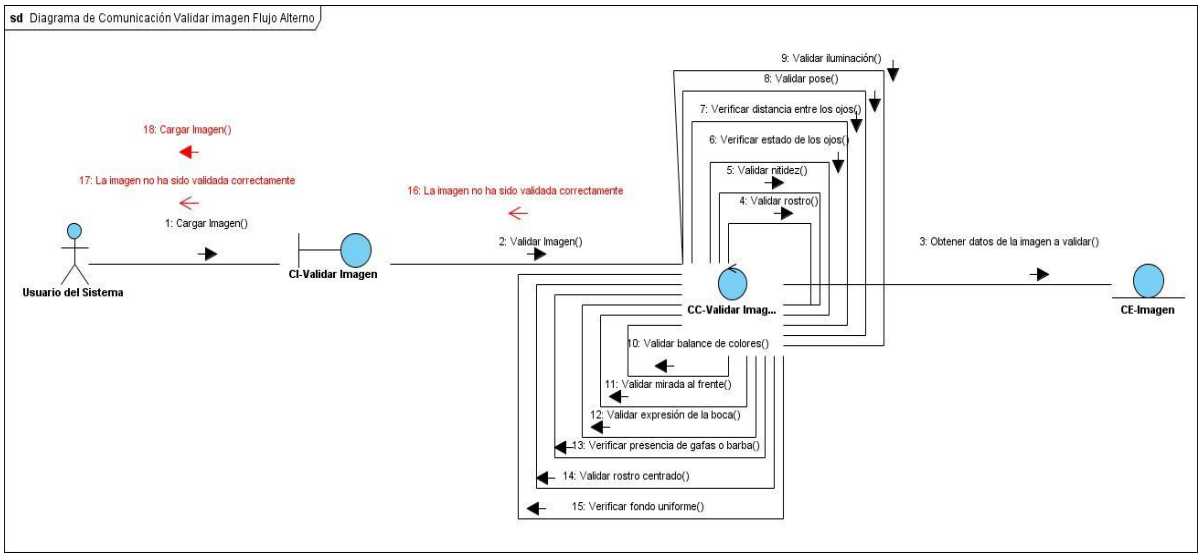

Figura 19.Diagrama de Colaboración Flujo Alterno.

2.8 Diseño

El Modelo de Diseño es una abstracción del Modelo de Implementación y su código fuente, el cual fundamentalmente se emplea para representar y documentar su diseño. Representa a los casos de uso en el dominio de la solución.

El diseño tiene el propósito de formular los modelos que se centran en los requisitos no funcionales y en el dominio de la solución y que prepara para la implementación y prueba del sistema. Pretende crear un plano del modelo de implementación, por lo que

el grueso del esfuerzo está en las últimas iteraciones de elaboración y las primeras de construcción.

El Modelo de Diseño puede contener: diagramas de clases, paquetes, subsistemas, cápsulas, protocolos, interfaces, relaciones, colaboraciones, atributos, las realizaciones de los casos de uso, entre otros que se puedan considerar para el sistema en desarrollo.

2.8.1 Patrones de diseño utilizados

Estos son los patrones que se aplicarán para realizar el diseño de la aplicación que propone la investigación.

Experto en información es el principio básico de asignación de responsabilidades. De este modo obtendremos un diseño con mayor cohesión y así la información se mantiene encapsulada (disminución del acoplamiento).

Creador: El patrón Creador guía la asignación de responsabilidades relacionadas con la creación de objetos, tarea muy frecuente en los sistemas orientados a objetos.

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 bastante fácil darle mantenimiento, entenderla y reutilizarla.

Bajo Acoplamiento: Asignar una responsabilidad para mantener bajo acoplamiento. Las clases deben comunicarse con un número pequeño de clases tanto como sea posible.

Controlador: El patrón controlador es un patrón que sirve como intermediario entre una determinada interfaz y el algoritmo que la implementa. (33)

2.8.2 Patrones de arquitectura utilizados

El patrón de arquitectura propuesto para estructurar el componente de validación y post- procesamiento de imágenes faciales para su uso en documentos oficiales es el de Tuberías y Filtros que pertenece al estilo arquitectónico flujo de datos el cual enfatiza la reutilización y la modificabilidad. Es apropiada para sistemas que implementan transformaciones de datos en pasos sucesivos.

El patrón de arquitectura de tubos y filtros provee una estructura para procesar flujos de datos. Cada paso de procesamiento se encapsula en un filtro. Los datos se pasan usando los tubos entre filtros adyacentes. Recombinado los filtros se pueden construir distintas familias de sistemas relacionados. Para el componente el flujo de datos se comporta de la siguiente manera con la entrada de una imagen la misma pasara por un tubo para la validación del mismo se filtrara la imagen ya validada para que pase

por otro tubo donde será normalizada y como resultado de un nuevo filtro se obtendrá la imagen lista para usar en los documentos oficiales.

Estructura

- Filtros:
  - ✓ unidades de procesamiento en las arquitecturas de tubos y filtros.
  - ✓ enriquece, refina y/o transforma los datos.
  - ✓ el procesamiento se inicia:
  - ✓ el elemento siguiente solicita datos (pull)
  - ✓ el elemento anterior envía datos (push)
  - ✓ el filtro tiene un ciclo interno que solicita datos del filtro anterior y envía datos al siguiente (filtro activo).
  - ✓ los filtros no comparten estado.
  - ✓ los filtros no saben de la existencia o identidad de otros filtros.
- Tubos:
  - ✓ conectan origen-filtro, filtro-filtro, filtro-destino
  - ✓ transferencia de datos con un buffer First-In-First-Out
  - ✓ si dos filtros activos se comunican con un tubo, éste los sincroniza
- Origen de datos:
  - ✓ input del sistema
  - ✓ brinda al sistema una serie de datos con la misma estructura o tipo
  - ✓ ejemplos: archivo de líneas de texto, sensor
  - ✓ pueden también ser activos (push) o pasivos.
- Destino de datos:
  - ✓ a él llegan los resultados del procesamiento del sistema
  - ✓ existen destinos activos (pull) y pasivos

Implementación

- Dividir las tareas en una secuencia de pasos de procesamiento.
  - ✓ los procesos deben ser independientes y ordenados.
- Definir el formato de los datos transmitidos a través de cada pipe.
  - ✓ flexibilidad vs. performance.
- Decidir implementación de cada tubo.
  - ✓ Determinar la implementación de los filtros (activos o pasivos)
- Diseñar e implementar los filtros.
- Diseñar política de errores.
- Instalar el sistema.

Características

- Los tubos no tienen estado interno y simplemente comunican datos entre los filtros.
- Tubos y filtros ejecutan en forma no determinantica sobre los datos hasta que éstos se acaban.
- Restricciones de conexión:
  - ✓ existe una fuente de datos conectada al puerto input de un filtro; existe un destino de datos conectado al puerto output de un filtro

Ventajas

- Simplicidad:
  - ✓ forma limitada de interacción con el ambiente;
  - ✓ la funcionalidad es sólo la composición de funcionalidades más sencillas;
  - ✓ no existen interacciones complejas;
  - ✓ promueve la reutilización y simplifica el mantenimiento;
  - ✓ composición jerárquica:

- una combinación de tubos y filtros puede mostrarse externamente como un único filtro;

  ✓ por la independencia de procesamiento de los filtros, puede hacerse ejecución paralela o distribuida aumentando la disponibilidad y la performance.

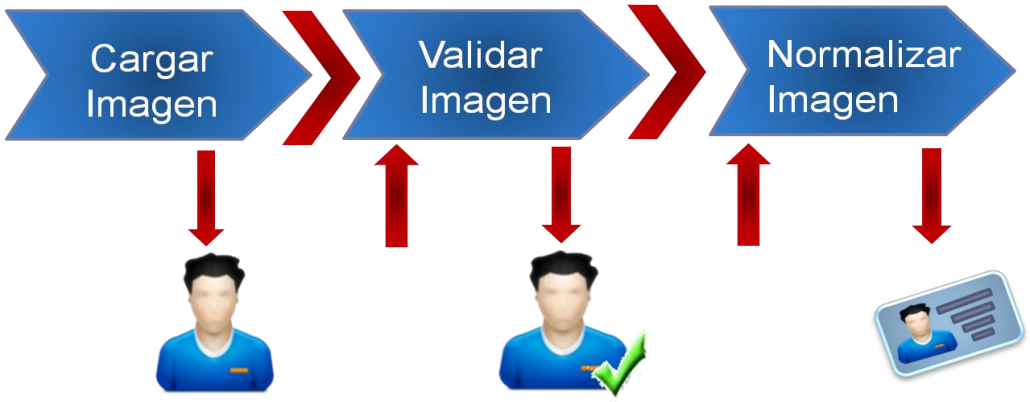

Figura 21. Aquitectura de tuberías y filtros.

2.8.2 Clases del diseño

Una clase del diseño es una abstracción si costuras de una clase o construcción similar en la implementación de sistema. Esta abstracción es sin costuras en el sentido en que, el lenguaje de programación para especificar una clase del diseño es lo mismo que el lenguaje de programación, la visibilidad de los atributos se especifica. Las relaciones entre las clases del diseño, a menudo tienen un significado directo cuando la clase es implementada, los métodos tienen correspondencia directa con los métodos implementados. (34)

2.8.3 Diagrama de clases del diseño

Un diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces en una aplicación. Los diagramas de clases del diseño contienen la siguiente información:

- Clases, asociaciones y atributos.
- Interfaces, con sus operaciones y constantes.
- Métodos.
- Información sobre los tipos de los atributos.
- Navegabilidad.
- Dependencias

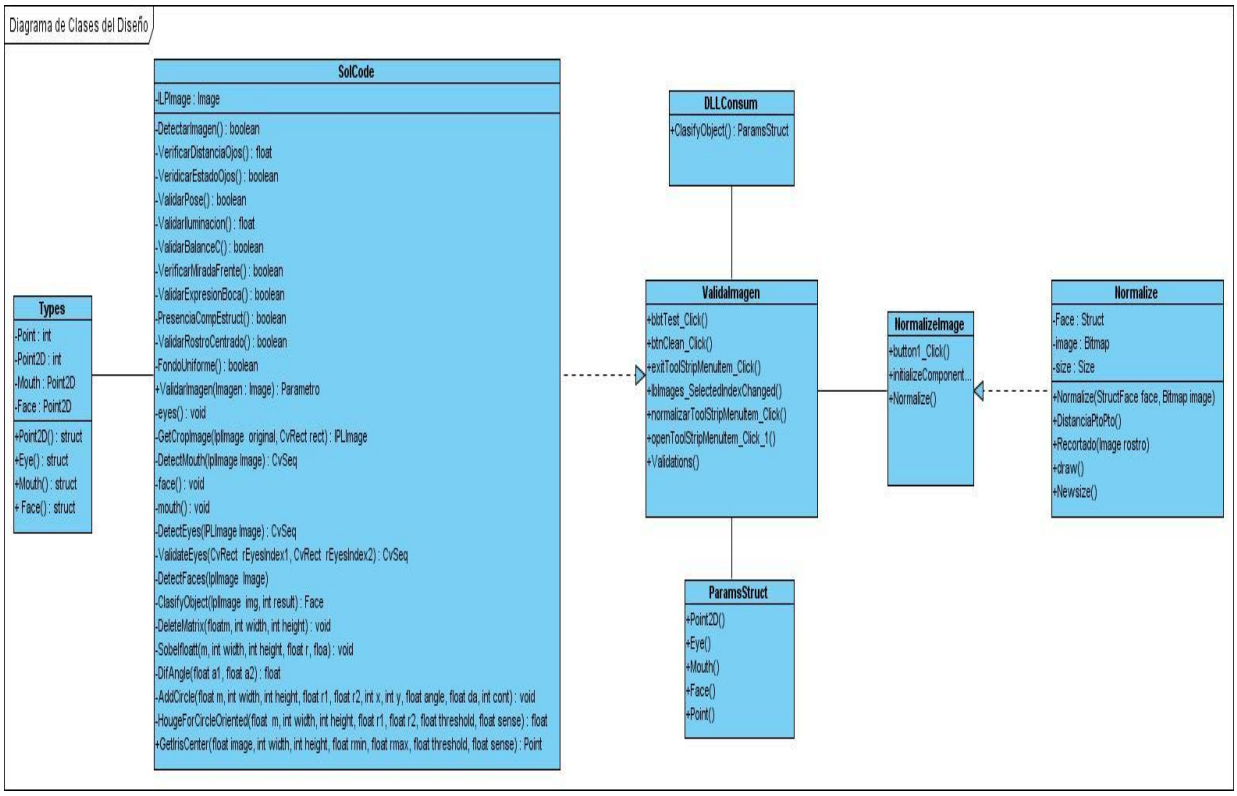

Figura 22.Diagrama de Clases de Diseño.

2.8.4 Descripción de las clases del diseño

A continuación se muestran las descripciones de las clases del diseño, donde se tiene su nombre, tipo, atributó, tipo de atributo, responsabilidad y una breve descripción para la misma.

Nombre: DLLConsum	
Tipo de clase:	
Atributo	Tipo
Responsabilidades	
Nombre	Descripción
ClasifyObject() :ParamsStruct Face	Se consume la dll.

Tabla 7.Descripción de la Clase DLLConsum.

Las demás descripciones se encuentran en el Anexo 1.

2.9Diagrama de secuencia

Un diagrama de secuencia representa una interacción como un gráfico bidimensional. La dimensión vertical es el eje de tiempo, que avanza hacia abajo de la página. La

dimensión horizontal muestra los roles de clasificador que representan objetos individuales en la colaboración. Cada rol de clasificador se representa mediante una columna vertical-línea de vida. Durante el tiempo que existe un objeto, el rol se muestra por una línea discontinua. Durante el tiempo que dura una activación de un procedimiento en el objeto, la línea de vida se dibuja como una línea doble.

Se muestra un mensaje como una flecha desde la línea de vida de un objeto a la del otro. Las flechas se organizan en el diagrama en orden cronológico hacia abajo. Los diagramas de secuencias son buenos para mostrar qué objetos se comunican con qué otros objetos y que mensajes disparan a esas comunicaciones.

Un diagrama de secuencia es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino. (35)

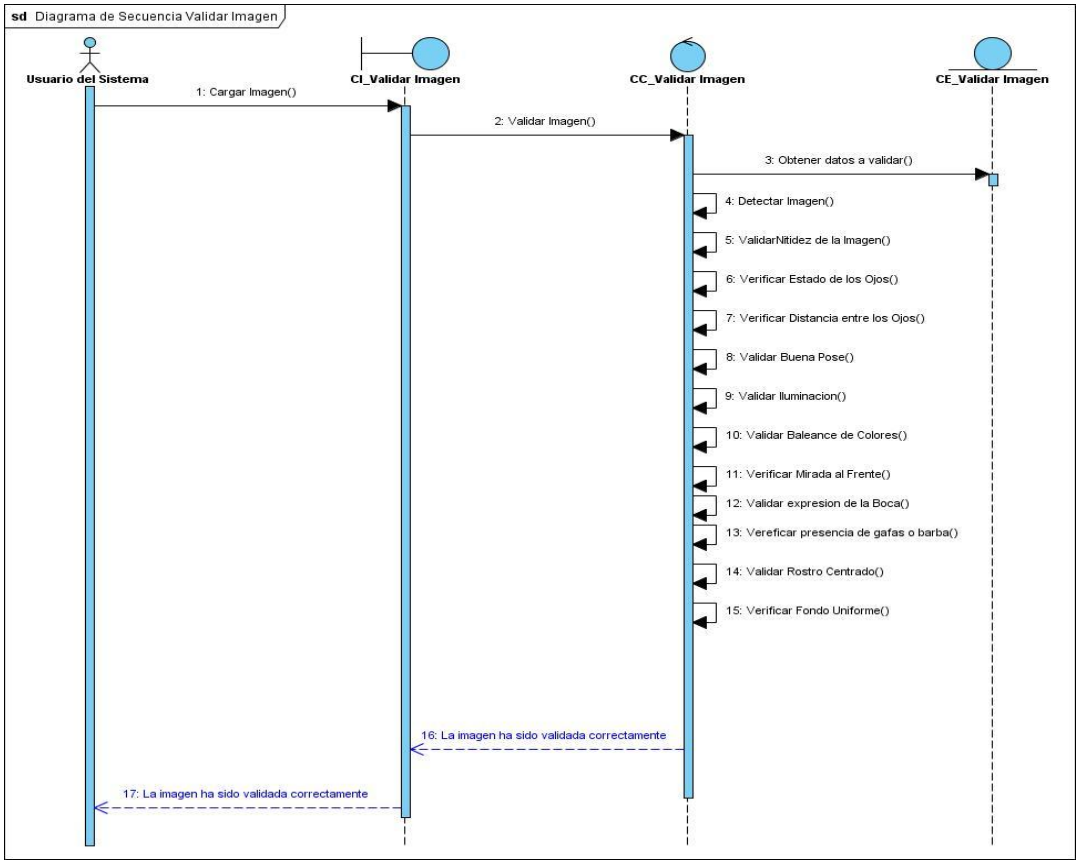

Figura 23.Diagrama de Secuencia Validar Imagen.

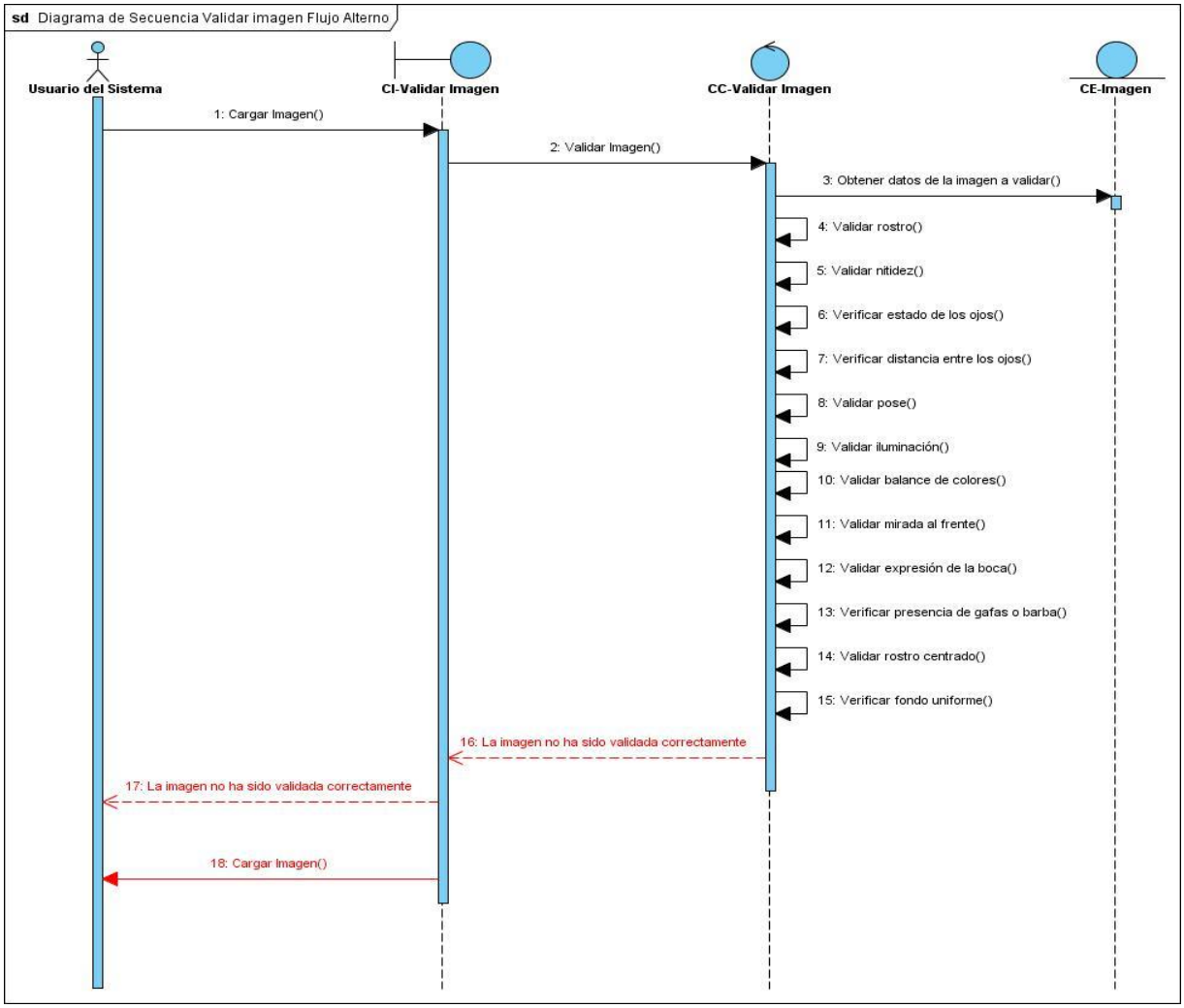

Figura 24.Diagrama de Secuencia Validar Imagen Flujo Alterno.

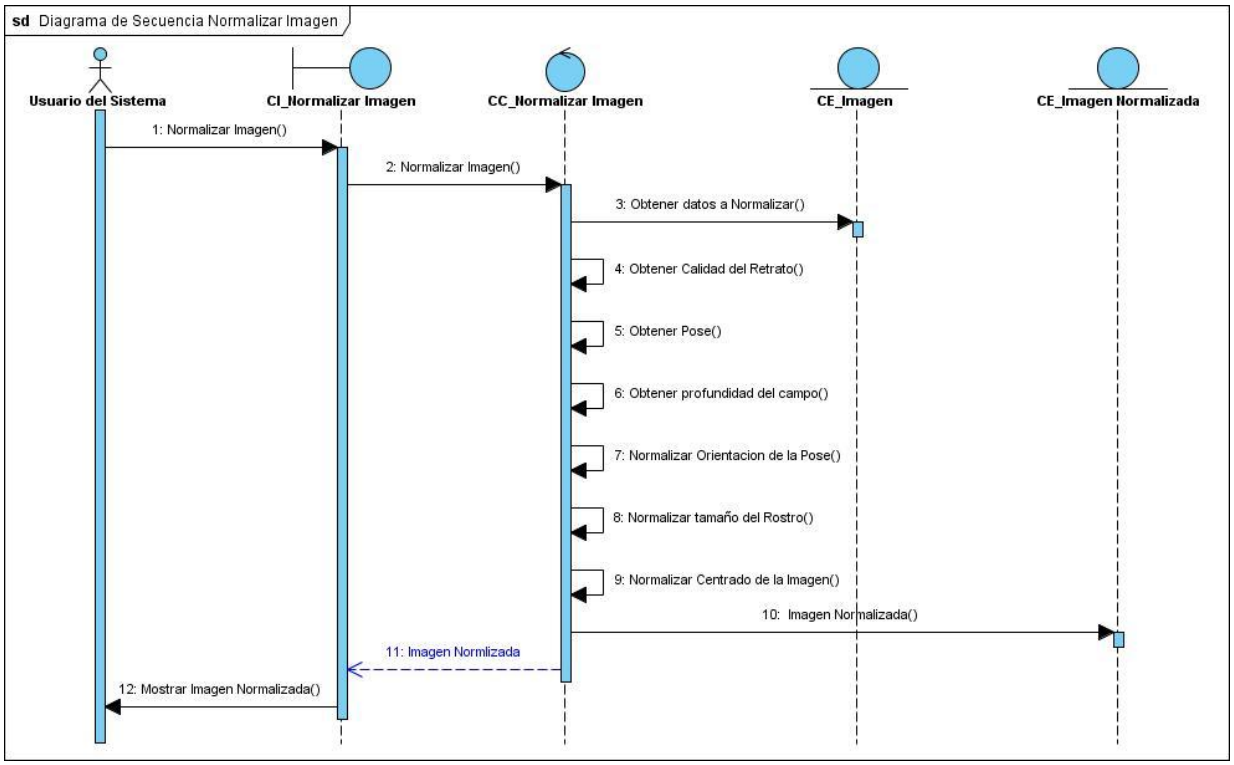

Figura 25.Diagrama de Secuencia Normalizar Imagen.

2.10 Conclusiones parciales

En este capítulo se realizaron los diferentes diagramas de clases del análisis y del diseño. Fueron explicados los diferentes patrones de diseño empleados que dan una mayor robustez a la arquitectura de Tuberías y Filtros utilizada en el componente, la cual brinda la posibilidad de ir transformando un flujo de datos en un proceso comprendido por varias fases secuenciales siendo la entrada de cada una la salida de la anterior . Los patrones de diseño empleados son buenas prácticas que a lo largo de la historia han demostrado capacidades de reutilización, extensibilidad y escalabilidad. Cada uno de los patrones utilizados está bien justificado y con la actual investigación se ha podido constatar que su uso ha sido efectivo, garantizando al grupo de desarrollo la posibilidad de extender el componente sin muchas contradicciones y de manera sencilla. Los artefactos generados en este capítulo constituyen una entrada indispensable para la implementación del componente.

Capítulo III Implementación y Pruebas.

3.1 Introducción

En la implementación empezamos con el resultado del diseño e implementamos el sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. Entré los propósitos de la implementación se encuentran. Distribuir el sistema asignado componentes ejecutables a nodos en el diagrama de despliegue. Esto se basa en las clases activas encontradas durante el diseño. Implementar las clases y subsistemas encontrados en el diseño. Probar los componentes individualmente y a continuación integrarlos compilándolos y enlazándolos en uno o más ejecutables.

En el flujo de trabajo de prueba verificamos el resultado de la implementación probando cada construcción, incluyendo tanto construcciones internas como intermedias, así como las versiones finales a ser entregadas a terceros.

En el presente capitulo se expondrán el diagrama de despliegue y componente, además se describen los conceptos relacionados con los mismos, Quedan definidas las pruebas a realizar al componente de validación post- procesamiento de imágenes faciales par uso en documentos oficiales.

3.2 Diagrama de despliegue

El Modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos del cómputo. El modelo de despliegue se utiliza como entrada fundamental en las actividades, de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño yrepresenta una correspondencia entre la arquitectura de software y la arquitectura del sistema (el hardware).

El mismo está compuesto por:

- ✓ Nodos: Elementos de procesamiento con al menos un procesador, memoria, y posiblemente otros dispositivos.
- ✓ Dispositivos: Nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela.
- ✓ Conectores: Expresan el tipo de conector o protocolo utilizado entre el resto de los elementos del modelo.

El que se muestra a continuacion utiliza una computadora en donde estará la aplicación que se encarga de validar y normalizar las imágenes faciales, las cuales pueden ser obtenidas de un dispositivo de captura o un directorio en cual estén previamente almacenadas. Los nodos se relacionan mediante conexiones bidireccionales (en principio) que pueden a su vez estereotiparse. Las conexiones se modelan como asociaciones, con todas las características que implica. En este caso las conexiones entre nodos estarán dadas por una conexión USB para el dispositivo y una FTP para el directorio con las imágenes.

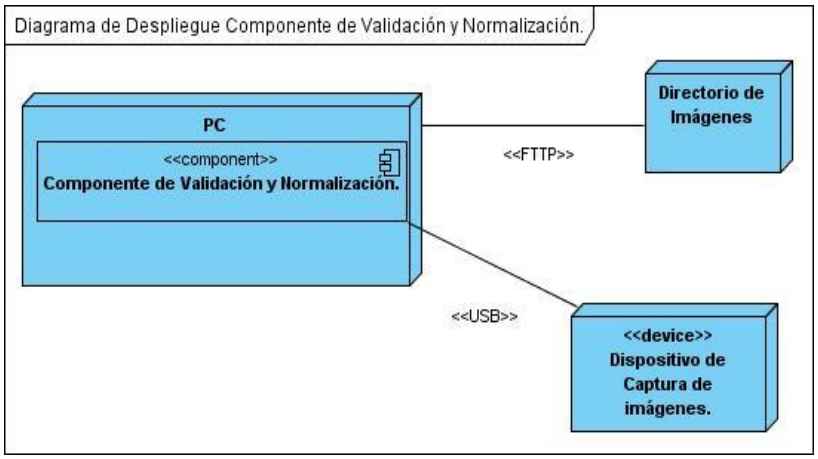

Figura 26.Diagrama de Despliegue.

3.3 Modelo de implementación

El modelo de implementación describe como los elementos del diseño, como las clases, se implementan en términos de componentes, como ficheros de código fuente, ejecutables. El modelo de implementación describe también como se organizan los componentes de acuerdo con los mecanismos de estructuración disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y como dependen los componentes unos de otros. (36)

3.3.1Diagrama de componentes

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Muestran las opciones de realización incluyendo código fuente, binario y ejecutable. Los componentes representan todos los tipos de elementos software que entran en la fabricación de aplicaciones informáticas. Pueden ser simples archivos, paquetes, bibliotecas cargadas dinámicamente. Él diagrama de componentes es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño

El diagrama de componentes describe la descomposición física del sistema de software (y eventualmente, de su entorno organizativo) en componentes, a efectos de construcción y funcionamiento.

Los componentes identifican objetos físicos que hay en tiempo de ejecución, de compilación o de desarrollo, y tienen identidad propia y una interfaz bien definida. Los componentes incluyen código en cualquiera de sus formatos (código, fuente o ejecutable), DLL, imágenes, pero también pueden ser documentos manuales cuando se describen partes no informatizadas de un sistema de información. (37) (38)

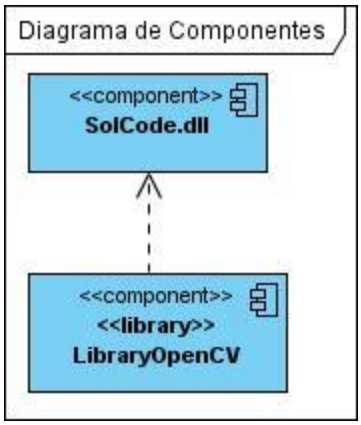

Figura 27.Diagrama de Componentes.

3.4 Pruebas

En el ámbito de la Ingeniería de Software, toda la actividad se guía por la aplicación de técnicas: procedimientos técnicos o de gestión q ayudan en la ejecución, evaluación y mejora de los procesos de desarrollo de software (IEEE, 1990).

Las técnicas más comunes aplicadas en los procesos de prueba tienen el objetivo de seleccionar buenos casos de prueba, estos es , casos que tengan una probabilidad alta de descubrir un error .Tradicionalmente (Myers, 2004; Bézier, 1990) se han considerado dos enfoques complementarios para seleccionar casos de prueba, denominados pruebas de caja blanca y caja negra. Las técnicas de caja blanca, también denominadas pruebas estructurales, utilizan el código fuente del programa, y especialmente su estructura de control, para seleccionar casos de prueba. Por otro lado las técnicas de cajas negras o funcionales, obtienen casos a partir de los requisitos funcionales del programa a probar por lo que no se tiene en cuenta la forma en que se codifica esa funcionalidad, sino que se consideran únicamente entradas y salidas.

Otras de las clasificaciones son técnicas estáticas y dinámicas. En la primera no es necesario ejecutar el software bajo prueba para detectar errores, mientras que en las dinámicas el software siempre es ejecutado por lo que las primeras no son consideradas pruebas sino técnicas de aseguramiento de la calidad. Porque podemos incluir las pruebas de caja blanca y negra como dinámicas según la definición dada anteriormente.

3.4.1 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:

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 porciento el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad. [Pressman, 2000]

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.

Hay 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.

```
void Sobel(float** m, int width, int height, float** r, float** o)
{
int w = width, h = height;
float vx, vy;//1
for (int i = 1; i < h - 1; i++)//2
 {
for (int j = 1; j < w - 1; j++)//3
 {
 vx = m[i - 1][j - 1] + 2 * m[i][j - 1] + m[i + 1][j - 1] -
 (m[i - 1][j + 1] + 2 * m[i][j + 1] + m[i + 1][j + 1]);

 vy = m[i - 1][j - 1] + 2 * m[i - 1][j] + m[i - 1][j + 1] -
 (m[i + 1][j - 1] + 2 * m[i + 1][j] + m[i + 1][j + 1]);
                  r[i][j] = (float)sqrt(vy * vy + vx * vx);
 o[i][j] = (float)atan2f(vy, vx);//4
if (o[i][j] < 0) //5
                  o[i][j] += 3.14159265354 * 2;////6
 }
 }
}//7
```

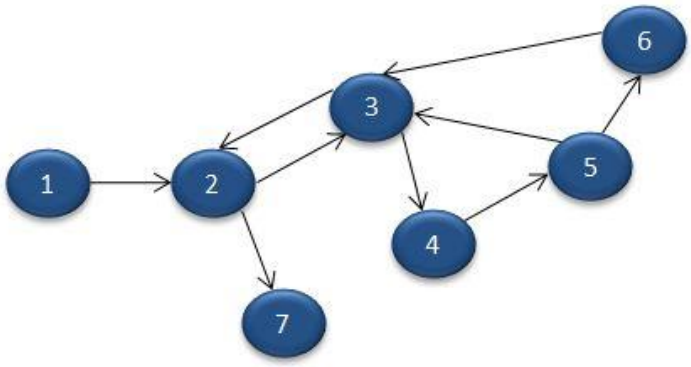

Figura 28.Grafo Funcionalidad: Sobel

Complejidad ciclomática para la funcionalidad Sobel.

- Fórmula1

  V (G) = (A (Aristas) – N (Nodos)) + 2

  V (G) = (9 – 7) + 2 = 4

- Fórmula 2

  V (G) = P (Nodos Predicados) + 1

  V (G) = 3 + 1 = 4

- Fórmula 3

  V (G) = R = 5

  Caminos independientes obtenidos

  1-2-7

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

  1-2-7

  1-2-3-2-7

  1-2-3-4-5-3-2-7

```
float** ConvertToFloatArray(IplImage *img, int ancho, int largo)
{
int height=largo;
int width=ancho;
RgbImage data(img);
float** m = CreateMatrix(width,height);//1
for(int i = 0; i < height; i++ ) //2
      for(int j = 0; j < width; j++ )//3
       m[i][j] = (0.11f * data[i][j].r + 0.59f * data[i][j].g + 0.30f
* data[i][j].b );//4
```

```
return m;//5
}
```

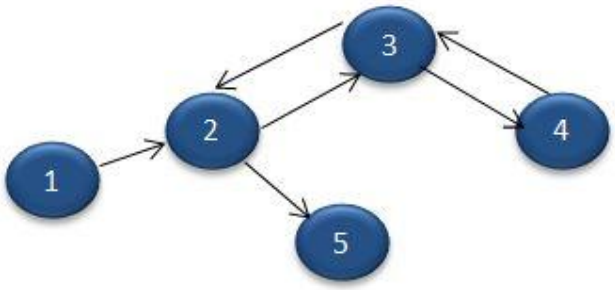

Figura 29.GFrafo Funcionalidad: ConvertToFloatArray

Complejidad ciclomática para la funcionalidad ConvertToFloatArray.

- Fórmula1

  V (G) = (A (Aristas) – N (Nodos)) + 2

  V (G) = (6 – 5) + 2 = 3

- Fórmula 2

  V (G) = P (Nodos Predicados) + 1

  V (G) = 2 + 1 = 3

- Fórmula 3

  V (G) = R = 3

Caminos independientes obtenidos

1-2-5

1-2-3-2-5

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

3.5 Conclusiones parciales

En este capítulo fueron realizados los diagramas de despliegue y el de componente, se definieron las pruebas a realizar al componente y se desarrollaron los casos de prueba correspondientes según el método de caja blanca que fue el seleccionado para la realización de la misma probando así la calidad del componente, ya que el resultado de las pruebas fue satisfactorio.

Conclusiones

Durante el desarrollo de la presente investigación se consulto bibliografía referente al tema de reconocimiento de rostro para lograr tener un mayor conocimiento sobre el reconocimiento de personas a través de patrones biométricos. Se investigaron los sistemas existentes tanto en Cuba como el mundo dando como resultado la necesidad real del desarrollo del componente de validación y post- procesamiento de imágenes faciales para los documentos oficiales. Se seleccionaron las herramientas, metodologías y lenguajes más adecuados para el desarrollo del componente. Para la obtención de los requisitos se tuvo en cuenta las normas propuestas por la Organización de Aviación Civil Internacional (OACI) para los documentos oficiales. Se obtuvieron todos los artefactos de cada flujo de trabajo según lo establecido por RUP.

Se le dio cumplimiento al objetivo fundamentalmente del trabajo ya que se implemento la librería SolCode con la cual se facilitara el proceso de validación y quedaran establecidas mejoras en la calidad de las imágenes faciales. Las pruebas realizadas garantizan que disminuyan en un gran porciento el número de errores existentes y por ende una mayor calidad y confiabilidad, con las misma se demuestra la validez del componente ya que lo resultados obtenidos fueron satisfactorios.

Recomendaciones

- Fortalecer con nuevos criterios el tema de la iluminación en las imágenes
- Continuar con el desarrollo del componente.

Bibliografía

1. **A.J. Goldestein, L.D. Harmon, and A.B. Lesk.** *"Identification of Human faces". .* s.l. : s.l. : Proc. IEE , Vol. 59, No. 5, 748-760,, May 1971.

2. **Kirby, L. Sirovich and M.** *"A Low-Dimensional Procedure for the Chatecterization of Human Faces". .* s.l. : s.l. : J. Optical Soc. Am. A, . : s.l. : J. Optical Soc. Am. A, , 1987.

3. **M. A Turkand, A.P Pentland.** *"Face Recognition Using Eigenfaces". .* s.l. : s.l. : Proc IEEE., 586-591, 1991.

4. **P. J. Phillips, P. J. Flyn, T. Scruggs, K. W. Boyer, J. Chang, K. Hoffman, J. Marques, J. Min and W. Wored.** *"Overview of the Face Recognition Grand Challenge".* San Diego : Proc. Computer Vision and Pattern Recognition Conference, 2005.

5. **D. M. Blackun, J. M. Bone and P.J. Phillips.** *"Facial Recognition Vendor Test".* Febrero 2001.

6. **Internacional, Organizacion de Aviacion Civil.** *"Documentos de viaje de lectura mecánica. ".* s.l. : Parte 3, volumen 1., 2008.

7. **González, Wintz.** *"Procesamiento digital de imágenes.".* 1996.

8. **Sons, John Wiley.** *"Image processing: principles and applications.".* 2005.

9. **Zksoftware.** Zksoftware. [En línea] http://www.11298.tradebig.com..

10. [En línea] www.identix.com..

11. [En línea] http://www.l1id.com/pages/17..

12. [En línea] www.cenatav.co.cu..

13. **Dra. Isneri Talavera Bustamante, Ing. Jorge Luis Rodriguez Hierrezuelo.** *Reconocimiento de Patrones.".* La Habana : s.n., 2008.

14. **Albi, Denns.** *"BMW utiliza reconocimiento de rostro para personalizar los coches ".* Marzo 18, 2008.

15. Sistema de Acceso por Reconocimiento facial . [En línea] http://www.solostocks.com.co..

16. **Laurent Debrauwer, Fien Van der Heyde.** *"UML2 iniciación y ejemplos corregidos".*

17. [En línea] http://www.visual-paradim.com.

18. [En línea] http://www.visual-paradim.com.

19. [En línea] http://mundolibre10.blogspot.com/2010/03/rational-rose-enterprise-edition-suite.html ..

20. **Barahona, Jesus M. Gonzalez.** *"Intoduccion al software libre".* s.l. : Madrid : Grupo de Sistemas y Comunicaciones, ESCET, Universidad Rey Juan Carlos.

21. [En línea] http://plataformaeclipse.com/..

22. [En línea] http://netbeans.org/..

23. [En línea] http://www.slideshare.net/soreygarcia/net-framework-981946.

24. [En línea] http://es.wikipedia.org/wiki/Microsoft_.NET.

25. *Selecting a development aproach.* 2008.

26. **Jacobson, Booch, Rumbaugh.** *"Proceso unificado en el desarrollo de software " .*

27. **Agam, Gady.** *"Introduction to programming with Open CV".* Illinois : s.n., Enero 27, 2006.

28. **Gary Bradski, Arian Kaebler.** *"Learning Open CV ". .* 2008.

29. **Proenza, Y.** *"'Introduccion al modelo conceptual".* s.l. : La Habana , Universidad de Ciencias Informaticas., 2005.

30. **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.

31. —. *Flujo de trabajo Captura de requisitos. Modelo de Negocio".* s.l. : Ciudad de la Habana, Universidad de Ciencias Informáticas. ,2004.

32. —. *Flujo de trabajo Captura de requisitos. Modelo de Negocio". .* s.l. : Ciudad de la Habana, Universidad de Ciencias Informáticas, 2004.

33. **Erich Gamma, Richard Helm, Ralph Jonson y John Vlissides Gamma, E. Helm, R. Jonson, R. & Vlissides, J..** *Design Patterns: Elements of Reusable Object-Oriented Software".* s.l. : Addison Wesley , 1995.

34. **Ivar Jacobson, Grady Booch, James Rumbaugh Pearson.** *"Proceso Unificado de desarrollo de software".* s.l. : s.l. : Educación S. A. pag 120, 2006.

35. **Ivar Jacobson, Grady Booch, James Rumbaugh Pearson.** *"Proceso Unificado de desarrollo de software.".* s.l. : . s.l. : Educación S. A. , 2000.

36. —. *"Proceso Unificado de desarrollo de software".* s.l. : .". s.l. : Educación S. A.pag 257.

37. **Falgueras, Benet Campderrich.** *Ingeniería de Software .*

38. **Ivar Jacobson, Grady Booch, James Rumbaugh.** *"Proceso Unificado de desarrollo de software".* s.l. : Pearson Educación S. A. 2000 .

39. **Ivar Jacobson, Grady Booch, James Rumbaugh Pearson.** *"Proceso Unificado de desarrollo de software".* s.l. : Educación S. A. 2000.

Glosario de Términos

Rasgos conductuales: son aquellos que se soportan sobre características de la conducta del ser humano tales como: pulsaciones del teclado, discurso, dinámica de la firma, etc.

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

NFL: La National Football League (en español: Liga Nacional de Fútbol Americano).

Minería de Datos: La minería de datos (DM, Data Mining) consiste en la extracción no trivial de información que reside de manera implícita en los datos.

Fotograma: Se denomina a cada una de las imágenes impresionadas químicamente en la tira de celuloide del cinematógrafo o bien en la película fotográfica; por extensión también se llama de ese modo a cada una de las imágenes individuales captadas por cámaras de video y registradas analógica o digitalmente.

BD: base datos

USB: El Universal Serial Bus (bus universal en serie) o mejor conocido como Conductor Universal en Serie (CUS), abreviado comúnmente USB, es un puerto que sirve para conectar periféricos a un ordenador.

Ethernet: es un estándar de redes de computadoras de área local con acceso al medio por contienda CSMA/CD. ("Acceso Múltiple por Detección de Portadora con Detección de Colisiones"), es una técnica usada en redes Ethernet para mejorar sus prestaciones.

TCP/IP: Protocolo de Control de Transmisión (del inglés Transmission Control Protocol.) y Protocolo de Internet Es un protocolo de comunicación del nivel de transporte orientado a conexión.

IBM: International Business Machines o IBM (conocida coloquialmente como el Gigante Azul) es una empresa multinacional estadounidense que fabrica y comercializa herramientas, programas y servicios relacionados con la informática.

ECMA: Ecma International es una organización internacional basada en membrecías de estándares para la comunicación y la información.

ISO: La Organización Internacional de Normalización, es el organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales a excepción de la eléctrica y la electrónica.

Ruby: lenguaje interpretado de propósito general inspirado en Python, Perl, Smalltalk, entre otros.

PHP: lenguaje interpretado diseñado para la creación de páginas web dinámicas.

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

GHz: Frecuencia de vibraciones eléctricas (ciclos) por segundo. Abreviado un Hz es igual a un ciclo por un segundo.1 GHz equivale 10^9 Hz.

RAM: Es un tipo de memoria de ordenador a la que se puede acceder aleatoriamente; es decir, se puede acceder a cualquier byte de memoria sin acceder a los bytes precedentes. La memoria RAM es de tipo de memoria más común en ordenadores y otros dispositivos como impresoras.

MB: El Megabyte es una unidad de cantidad de datos informáticos. Es múltiplo del byte u octeto, que equivale a 10^6 bytes.

Pixel: Un píxel o pixel, plural píxeles (acrónimo del inglés picture element, "elemento de imagen") es la menor unidad homogénea en color que forma parte de una imagen digital.

OEM: En inglés equipment manufacturer u OEM (en español "fabricante de equipamiento original") es una empresa que fabrica productos que luego son comprados por otra empresa y vendidos bajo la marca de la empresa compradora.

IDE :Un entorno de desarrollo integrado (IDE) (también conocido como entorno de diseño integrado o entorno de depuración integrada) es una aplicación de software que ofrece servicios integrales a los programadores de computadoras para el desarrollo de software.

Ada: Es un lenguaje de programación orientado a objetos y fuertemente tipado de forma estática. Es un lenguaje multipropósito, orientado a objetos y concurrente, pudiendo llegar desde la facilidad de Pascal hasta la flexibilidad de C++.

ANSI C: Es un estándar publicado por el Instituto Nacional Estadounidense de Estándares (ANSI), para el lenguaje de programación C.

Python: Es un lenguaje de programación de alto nivel cuya filosofía hace hincapié en una sintaxis muy limpia y que favorezca un código legible.

Smalltalk: Es un lenguaje de programación que permite realizar tareas de computación mediante la interacción con un entorno de objetos virtuales. Metafóricamente, se puede considerar que un Smalltalk es un mundo virtual donde viven objetos que se comunican mediante el envío de mensajes.

Delphi: es un entorno de desarrollo de software diseñado para la programación de propósito general con énfasis en la programación visual. En Delphi se utiliza como lenguaje de programación una versión moderna de Pascal llamada Object Pascal.

Anexos

Anexo 1 Descripción de las clases del diseño.

Nombre: Types	
Tipo de clase:	
Atributo	Tipo
-Point	int
-Point2D	int
-Mouth	Point2D
-Eye	Point2D
-Face	Point2D
Responsabilidades	
Nombre	Descripción
Point():struct	Devuelve un punto(x,y).
Point2D (): struct	Devuelve un punto(x,y).
Mouth (): struct	Devuelve dos puntos: pointTop (x,y) y pointDown (x,y)
Eye (): struct	Devuelve dos puntos: pointTop (x,y) y pointDown (x,y)
Face(): struct	Devuelve una estructura rostro

Tabla 8.Descripción de la Clase Types.

Nombre: SolCode	
Tipo de clase: DLL	
Atributo	Tipo
-thresholdDiffYRectEyesUpper	define
-thresholdDiffYRectEyesUnder	define
-thresholdDiffYRectEyesMouth	define
-thresholdDiffCenterXMouthEyes	define
-thresholdMinAreaMouth	define

-thresholdMinAreaFace	define
-thresholdRostroCentrado	define
-thresholdIris	define
-thresholdOjos	define
-thresholdBuenaPose	define
-noError	define
-Error	define
-EyesNotFound	define
-FaceNotFound	define
-MouthNotFound	define
Responsabilidades	
Nombre	Descripción
ClasifyObject(int (result, char direction[], char cascade_face[], char cascade_eye[], char cascade_mouth[]):Face WINAPI	Devuelve el rostro con todos sus componentes.
BalancedeColores(IplImage * img): int	Verifica el balance de colores.
BuenaPose(Point centroizq,Point centroder, int y1L,int y1R): int	Verifica la pose del titular.
ComprobarOjosRojos (IplImage * img, Point c): int	Verifica que no existan ojos rojos.
GetCropImage(IplImage* original,CvRect rect): IplImage	Recorta una imagen.
DetectMouth(IplImage * img, char cascade_mouth[]):CvSeq	Encuentra la posición de la boca.
DetectEyes(IplImage * img, char cascade_eye[]): CvSeq	Encuentra la posición de los ojos
ValidateEyes(CvRect * rEyesIndex1,	Verifica que los ojos encontrados sean

CvRect * rEyesIndex2):int	los correctos
DetectFaces(IplImage * img, char cascade_face[]):CvSeq	Detecta el rostro.
Buenalluminacion(IplImage *frameResize): int	Verifica la iluminación.
RostrooCentrado(IplImage * imagen,CvPoint pt1,CvPoint pt2): int	Verifica que el rostro del titular este centrado
DeleteMatrix(float** m, int width, int height): void	Elimina en una matriz.
CreateMatrix(int width, int height): float	Crea una matriz
Sobel(float** m, int width, int height, float** r, float** o): void	Detecta los bordes en la imagen.
DifAngle(float a1, float a2): float	Define el ángulo del iris.
AddCircle(float** m, int width, int height, float r1, float r2, int x, int y, float angle, float da, int cont): void	Adiciona el círculo al iris encontrado.
HougeForCircleOriented(float** m, int width, int height, float r1, float r2, float threshold, float sense): float	Aplica transformada de Houge para saber la localización del iris.
GetIrisCenter(float** image, int width, int height, float rmin, float rmax, float threshold, float sense): Point	Devuelve el punto donde se encuentra el centro del iris.
ConvertToFloatArray(IplImage *img, int ancho, int largo): float	Convierte a float un arreglo.
MiradaFrente(Point punto,int ancho,int largo): int	Verifica que la Mirada del titular este al frente

Tabla 9.Descripción de la Clase SolCode.

Nombre: ParamsStruct	
Tipo de clase:	
Atributo	Tipo
Responsabilidades	
Nombre	Descripción
Point2D():Struc	Devuelve un punto(x,y).
Eye():struct	Devuelve un punto(x,y).
Mouth():struct	Devuelve dos puntos: pointTop (x,y) y pointDown (x,y)
Face():struct	Devuelve dos puntos: pointTop (x,y) y pointDown (x,y)

Tabla 10.Descripción de la Clase ParamsStruct.

Nombre: ValidarImagen	
Tipo de clase: Interfaz	
Atributo	Tipo
Responsabilidades	
Nombre	Descripción
exitToolStripMenuItem_Click(object sender, EventArgs e):void	Salir de menú.
openToolStripMenuItem_Click_1(object sender, EventArgs e):void	Carga la imagen.
normalizarToolStripMenuItem_Click(object sender, EventArgs e)void	Llama a la interfaz de normalizar.
btnClean_Click(object sender, EventArgs e):void	Limpia el listbox.
Validations(ParamsStruct.Face face): void	Verifica los resultados de los métodos.
lbImages_SelectedIndexChanged(object sender, EventArgs e):void	Selecciona la imagen a validar y dibuja la detección del rostro.

Tabla 11.Descripción de la Clase ValidarImagen.

Nombre: NormalizeImage	
Tipo de clase: Controladora	
Atributo	Tipo
-face	ParamsStruct.Face
-image	Bitmap
-size	Size
Responsabilidades	
Nombre	Descripción
NormalizeImage(ParamsStruct.Face face, Bitmap image)	Constructor
draw()Bitmap	Dibuja un rectángulo
Newsize():Size	Calcula el Nuevo tamaño de para la imagen
DistanciaPtoPto(ParamsStruct.Point2D inicial, Point final): double	Calcula distancia entre los puntos de la localización del rostro.

Tabla 12.Descripción de la Clase NormalizeImage.

Nombre: Normalize	
Tipo de clase: Interfaz	
Atributo	Tipo
-image	Bitmap
-face	ParamsStruct.Face
- size	Size
Responsabilidades	
Nombre	Descripción
Normalize(Bitmap image, ParamsStruct.Face face)	Constructor
button1_Click(object sender, EventArgs e): void	Llama al método de la clase controladora que normaliza la imagen.

Tabla 13.Descripción de la Clase Normalize.

Anexo 2 Revisión de requisitos.

La revisión es uno de los mejores métodos para la validación. Con la misma se logra descubrir una gran cantidad de defectos en los requisitos, reducir el costo del desarrollo y reduce el tiempo de pruebas. Las revisiones de requisitos consisten en una o varias reuniones planificadas, donde se intenta confirmar que los requisitos poseen los atributos de calidad deseados. Las reuniones de revisión se realizan habitualmente siguiendo un procedimiento compuesto por siete pasos:

Preparar el plan de la revisión. Este plan debería incluir al menos lo siguiente: Las tareas a realizar (esto es, las que se indican a continuación), la planificación temporal y las personas que participarán en la revisión.

Distribuir los documentos a revisar. Habitualmente, el único documento a revisar será el documento de especificación. Junto con este documento, también es necesario remitir a los participantes en la revisión todos aquellos documentos que ayuden a comprender adecuadamente el documento de especificación.

Preparar la reunión. Esta tarea es muy distinta, dependiendo de quien la realice: o Para el analista promotor de la reunión, esta tarea es fundamentalmente logística .Básicamente consiste en reservar una sala donde realizar la revisión, solicitar los materiales que sean necesario (desde papel y lápiz a cañones de proyección), preparar justificantes del tiempo empleado por el personal participante6 (en caso de que sea preciso), etc. Pará los analistas participantes, la preparación de la reunión es, probablemente, las más importante de las tareas de revisión. Durante esta tarea se deben leer cuidadosamente los documentos recibidos y anotar todas aquellos defectos (o cosas que parezcan defectos) con la finalidad de ponerlos de manifiesto durante la reunión posterior.

Realizar la reunión de revisión. Básicamente, El formato de esta reunión puede ser muy diverso: Puede oscilar desde una total falta de control7 hasta grupos muy formalizados y sujetos a protocolos de actuación.

Identificar los defectos y acciones a realizar. La lista de defectos y acciones recomendadas es el documento final obtenido en las revisiones de requisitos.

Realizar las correcciones que sean precisas a los documentos revisados. El analista promotor de la reunión (esto es, el analista encargado del documento de especificación) debe evaluar y, si lo estima conveniente, llevar a cabo, las acciones recomendadas que han surgido de la reunión de revisión.

Informar de las modificaciones realizadas a los participantes en la reunión. Una vez que los defectos en la especificación han sido subsanados, debería enviarse un breve informe de las tareas realizadas, y una copia corregida de los documentos de especificación, a los participantes en la reunión para su visto bueno.

Se realizaron varias reuniones con el jefe de proyecto siguiendo los pasos antes mencionados logran corregir lo erros encontrados.

Anexo 3 Matriz de Trazabilidad

CU / **RF**	Carga Imagen	Validar Imagen	Normalizar Imagen
Detectar imagen.	X	X	
Validar nitidez de la imagen.		X	
Verificar estado de los ojos.		X	
Verificar distancia entre los ojos.		X	
Validar buena pose.		X	
Validar iluminación.		X	
Validar balance de colores.		X	
Verificar mirada al frente.		X	
Validar expresión de la boca.		X	
Verificar presencia de espejuelos, barba o bigote.		X	
Validar rostro centrado.		X	
Verificar fondo uniforme.		X	
Obtener calidad del retrato.			X
Obtener pose.			X
Obtener profundidad del campo.			X
Normalizar orientación de la pose.			X
Normalizar tamaño del rostro.			X
Normalizar centrado de la imagen.			X

Tabla 14.Matriz de Trazabilidad.