

ESCUELA POLITÉCNICA NACIONAL

ESCUELA DE INGENIERÍA

RECONOCIMIENTO DE ROSTROS EN TIEMPO REAL UTILIZANDO UNA RED NEURONAL

**PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
SISTEMAS INFORMÁTICOS Y DE COMPUTACIÓN**

ROMERO ESCUNTAR KARLA MARIANA

DIRECTOR: DR. HUGO BANDA G.

QUITO, SEPTIEMBRE 2006

CONTENIDO

CONTENIDO	I
LISTA DE FIGURAS	III
LISTA DE TABLAS.....	V
RESUMEN.....	VI
PRESENTACIÓN	VII
CAPITULO 1.....	MARCO TEÓRICO
.....	1
1.1 CONCEPTOS FUNDAMENTALES SOBRE IMÁGENES DIGITALIZADAS..	1
1.1.1 PERCEPCIÓN VISUAL	1
1.1.2 IMAGEN DIGITAL	3
1.1.3 DIGITALIZACIÓN DE UNA IMAGEN	7
1.1.4 RESOLUCIÓN DE LA IMAGEN.....	8
1.1.5 RELACIONES BÁSICAS ENTRE PÍXELES	11
1.1.6 TRANSFORMACIONES ESPACIALES	12
1.1.7 COLOR DE LA IMAGEN.....	12
1.1.8 TRANSFORMADAS DE LA IMAGEN.....	15
1.1.9 MEJORA DE LA IMAGEN	24
1.1.10 RESTAURACIÓN DE IMÁGENES.....	30
1.1.11 SEGMENTACIÓN DE IMÁGENES	31
1.1.12 COMPRESIÓN DE IMÁGENES.....	33
1.2 METODOLOGÍAS PARA EL RECONOCIMIENTO DE PATRONES.....	36
1.2.1 RECONOCIMIENTO DE PATRONES [10, 12, 78, 86].....	36
1.2.2 RECONOCIMIENTO DE ROSTROS [47, 76, 91, 92, 90].....	39
1.3 FASES QUE INVOLUCRA EL RECONOCIMIENTO DE PATRONES	43
1.3.1 ADQUISICIÓN DE IMÁGENES	44
1.3.2 PREPROCESAMIENTO	45
1.3.3 EXTRACCIÓN DE CARACTERÍSTICAS	46
1.3.4 RECONOCIMIENTO E INTERPRETACIÓN	47
1.3.5 FASES APLICADAS AL RECONOCIMIENTO DE ROSTROS	47
1.4 REDES NEURONALES	51
1.4.1 DEFINICIÓN	51
1.4.2 ELEMENTOS	51
1.4.3 ESTRUCTURA DE UNA RED NEURONAL ARTIFICIAL.....	52
1.4.4 ARQUITECTURA DE UNA RED NEURONAL ARTIFICIAL.....	53
1.4.5 MECANISMOS DE APRENDIZAJE.....	53
1.4.6 TIPOS DE REDES	55
1.4.7 VENTAJAS DE LAS REDES NEURONALES.....	55
1.4.8 APLICACIONES	56
1.4.9 DISEÑO DE SOLUCIONES CON REDES NEURONALES ARTIFICIALES.....	56

1.5 ARQUITECTURAS MÁS ADECUADAS PARA EL RECONOCIMIENTO DE ROSTROS	58
1.5.1 PERCEPTRÓN MULTICAPAS	60
1.5.2 RED FUNCIÓN BASE RADIAL.....	61
1.5.3 RED KOHONEN.....	63
CAPITULO 2.....	ANÁLISIS Y DISEÑO
.....	65
2.1 REQUERIMIENTOS	66
2.1.1 ÁMBITO	66
2.1.2 ESTUDIO PRELIMINAR.....	66
2.1.3 VISIÓN.....	67
2.1.4 ANÁLISIS DE BENEFICIOS	67
2.1.5 CONCEPTO DE LA SOLUCIÓN	69
2.2 ANÁLISIS	93
2.2.1 ALTERNATIVAS DE DISEÑO CONCEPTUAL DE LA SOLUCIÓN.....	93
2.2.2 MODELO DE ANÁLISIS	94
2.2.3 IDENTIFICACIÓN DE PAQUETES DE ANÁLISIS	95
2.3 DISEÑO DEL SISTEMA	97
2.3.1 MODELO DE DISEÑO	97
2.3.2 IDENTIFICACIÓN DE SUBSISTEMAS Y SUS INTERFACES	139
2.3.3 DISEÑO DE PANTALLAS	141
2.3.4 DISEÑO FÍSICO DE DATOS	142
CAPITULO 3.....	IMPLEMENTACIÓN, PRUEBAS Y EVALUACIÓN
.....	143
3.1 LA HERRAMIENTA MATLAB	143
3.2 IMPLEMENTACIÓN DEL SISTEMA USANDO MATLAB	145
3.2.1 IDENTIFICACIÓN DE COMPONENTES	145
3.2.2 DIAGRAMAS DE COMPONENTES	145
3.3 PRUEBAS	148
3.3.1 PRUEBAS DE UNIDAD	148
3.3.2 PRUEBAS DE INTEGRACIÓN Y DE SISTEMA	149
3.3.3 REALIZACIÓN Y EVALUACIÓN DE PRUEBAS.....	149
3.4 EVALUACIÓN	154
3.4.1 RESULTADOS EXPERIMENTALES	154
CAPITULO 4.....	CONCLUSIONES Y RECOMENDACIONES
.....	162
4.1 CONCLUSIONES	162
4.2 RECOMENDACIONES	164
BIBLIOGRAFÍA	165
GLOSARIO	173
ANEXOS	177

LISTA DE FIGURAS

Figura 1.1. Imagen creada mediante primitivas geométricas	4
Figura 1.2. Transformación de una imagen bitmap a una imagen vectorial	5
Figura 1.3. Imagen de mapa de bits o bitmap.....	6
Figura 1.4. Digitalización de una imagen.....	7
Figura 1.5. Ejemplos de una imagen con diferentes resoluciones (a) a 96 ppp. (b) a 10 ppp y (c) a 5 ppp.	8
Figura 1.6. Ejemplos de una imagen con diferentes profundidades de píxel.....	10
Figura 1.7. Especificación del color (a) escala de matices (b) escala de brillo (para el matiz azul) (c) escala de saturación (para el matiz azul).....	13
Figura 1.8. Síntesis del Color: (a) síntesis aditiva (b) síntesis sustractiva.	14
Figura 1.9. Principio de la Transformada de Hough.	18
Figura 1.10. Modificación del histograma de una imagen (a) imagen inicial y su histograma (b) estiramiento del histograma (c) ecualización del histograma.	26
Figura 1.11. Perfilado de una imagen (a) imagen inicial (b) imagen perfilada una vez (c) imagen sometida a un excesivo perfilado.....	27
Figura 1.12. Filtrado de Wiener para remover ruido en una imagen (a) imagen inicial con presencia de ruido (b) imagen luego de aplicar el filtrado.	30
Figura 1.13. Modelos del rostro humano (a) en base a medidas antropométricas (b) diagrama elástico con nodos en puntos específicos del rostro.	41
Figura 1.14. Fases que involucra el reconocimiento de patrones.....	44
Figura 1.15. Esquema general para el reconocimiento de rostros.....	47
Figura 1.16. Ejemplos de redes con diferentes estructuras: (a) Perceptrón simple (b) Red ART.	53
Figura 1.17. Perceptrón Multicapas (3 capas).....	60
Figura 1.18. Red Función Base Radial.	61
Figura 1.19. Mapa Auto-Organizativo (SOM).	64
Figura 2.1. Diagrama de Casos de uso para FENIXVR – Verificación de Rostros (Parte I: Funcionalidad Básica).....	85
Figura 2.2. Diagrama de Casos de Uso para FENIXVR – Verificación de Rostros (Parte II: Funcionalidad Adicional)	86
Figura 2.3. Diagrama de Paquetes de Análisis.....	96
Figura 2.4. Diagrama de Secuencia - Diseño: Caso de uso <i>Registrar Usuario</i> (Flujo Básico - Imagen existente)	119
Figura 2.5. Diagrama de Secuencia - Diseño: Caso de uso <i>Registrar Usuario</i> (Flujo Básico – Capturar imagen)	122
Figura 2.6. Diagrama de Secuencia - Diseño: Caso de uso <i>Registrar Usuario</i> (Extensión – No se localizó ningún rostro en la imagen)	121
Figura 2.7. Diagrama de Secuencia - Diseño: Caso de uso <i>Verificar Identidad</i> (Flujo Básico - Imagen existente)	128
Figura 2.8. Diagrama de Secuencia - Diseño: Caso de uso <i>Verificar Identidad</i> (Flujo Básico – Capturar imagen)	130
Figura 2.9. Diagrama de Secuencia - Diseño: Caso de uso <i>Verificar Identidad</i> (Extensión – No se localizó ningún rostro en la imagen).....	130
Figura 2.10. Diagrama de Secuencia - Diseño: Caso de uso <i>Entrenar Red Neuronal</i> (Flujo Básico)	132

Figura 2.11. Diagrama de Secuencia - Diseño: Caso de uso Actualizar Parámetros de la Red Neuronal (Flujo Básico).....	134
Figura 2.12. Diagrama de Clases – Capa de Datos	135
Figura 2.13. Diagrama de Clases – Capa de Aplicación (Funcionalidad Básica).....	136
Figura 2.14. Diagrama de Clases – Capa de Aplicación (Funcionalidad Adicional).....	137
Figura 2.15. Diagrama de Clases – Capa de Aplicación (Subsistema Localizar Rostro). 138	
Figura 2.16. Identificación de Subsistemas e interfaces.....	140
Figura 2.17. Diagrama de jerarquía de pantallas (Opciones disponibles para el Operador)	141
Figura 2.18. Diagrama de jerarquía de pantallas (Opciones disponibles - Administrador)	141
Figura 3.1. Subsistema Registrar Usuario	145
Figura 3.2. Subsistema Verificar Identidad	146
Figura 3.3. Subsistema Entrenar Red	146
Figura 3.4. Diagrama de componentes – Subsistemas de implementación.....	147
Figura 3.5. Porcentaje de aciertos y fallos en la localización de rostros	155
Figura 3.6. Ejemplos de localización correcta del área del rostro.....	156
Figura 3.7. Ejemplos de localizaciones erróneas y localizaciones fallidas (no se localizó el rostro presente en la imagen).....	156
Figura 3.8. Porcentajes obtenidos para la verificación de rostros	159
Figura 3.9. Imágenes que forman parte de las tasas FRR y FAR.....	160
Figura 3.10. Imágenes que forman parte de las verificaciones correctas	160

LISTA DE TABLAS

Tabla 1.1. Clasificación de los filtros espaciales y del dominio de la frecuencia	29
Tabla 1.2. Una taxonomía de redes neuronales artificiales	55
Tabla 2.1. Costos y nivel de intrusión de las distintas tecnologías biométricas.....	68
Tabla 2.2. Descripción de los actores que intervienen en los casos de uso.....	74
Tabla 2.3. Descripción del caso de uso Registrar Usuario	77
Tabla 2.4. Descripción del caso de uso Obtener Imagen	78
Tabla 2.5. Descripción del caso de uso Detectar Cámara	79
Tabla 2.6. Descripción del caso de uso Localizar Rostro.....	80
Tabla 2.7. Descripción del caso de uso Verificar Identidad.....	82
Tabla 2.8. Descripción del caso de uso Entrenar Red Neuronal	83
Tabla 2.9. Descripción del caso de uso Actualizar Parámetros de la Red Neuronal.....	84
Tabla 2.10. Descripción de la clase de diseño <i>FENIXVR</i>	98
Tabla 2.11. Descripción de la clase de diseño <i>registrar_usuario_GUI</i>	100
Tabla 2.12. Descripción de la clase de diseño <i>verif_ident_GUI</i>	101
Tabla 2.13. Descripción de la clase de diseño <i>most_img_usu_GUI</i>	102
Tabla 2.14. Descripción de la clase de diseño <i>preferencias_GUI</i>	102
Tabla 2.15. Descripción de la clase de diseño <i>ManejadorRegistro</i>	103
Tabla 2.16. Descripción de la clase de diseño <i>ManejadorDetDisp</i>	104
Tabla 2.17. Descripción de la clase de diseño <i>LocalizacionRostro</i>	104
Tabla 2.18. Descripción de la clase de diseño <i>PatronImagen</i>	105
Tabla 2.19. Descripción de la clase de diseño <i>ManejadorVerificacion</i>	106
Tabla 2.20. Descripción de la clase de diseño <i>ManejadorPresentacionInfUsu</i>	106
Tabla 2.21. Descripción de la clase de diseño <i>ManejadorEntrenamientoRed</i>	107
Tabla 2.22. Descripción de la clase de diseño <i>ManejadorParametros</i>	108
Tabla 2.23. Descripción de la clase de diseño <i>Usuario</i>	108
Tabla 2.24. Descripción de la clase de diseño <i>Imagen</i>	108
Tabla 2.25. Descripción de la clase de diseño <i>Subespacio</i>	109
Tabla 2.26. Descripción de la clase de diseño <i>Preferencias</i>	109
Tabla 2.27. Descripción de la clase de diseño <i>red_rd</i>	110
Tabla 2.28. Descripción de la clase de diseño <i>EigenEspacioFx</i>	110
Tabla 2.29. Descripción de la clase de diseño <i>RostroPromFx</i>	110
Tabla 2.30. Descripción de la clase de diseño <i>ImagenReconstruida</i>	111
Tabla 2.31. Descripción de la clase de diseño <i>AccesoBD</i>	112
Tabla 3.1. Resultado de la realización de los casos de prueba de unidad	151
Tabla 3.2. Resultado de la realización de los casos de prueba de integración y de sistema	153

RESUMEN

Actualmente las tecnologías biométricas han despertado gran interés a nivel mundial, no sólo por constituir una forma natural de reconocer o identificar a los seres humanos sino también por proyectarse como el esquema estándar a ser empleado en un futuro para múltiples aplicaciones relacionadas con la identificación y autenticación de personas. Esto se debe principalmente a que emplean elementos intrínsecos y universales de los seres humanos, sus rasgos físicos y de comportamiento.

FENIXVR es una solución computacional orientada al campo de la biometría, que utiliza el rostro humano como forma de autenticación o verificación. Consiste en una mezcla de técnicas tanto de procesamiento de imágenes como de reducción de dimensionalidad y clasificación de datos, cada una de las cuales tiene diferente objetivo, que en conjunto permiten llevar a cabo todo el proceso de verificación de un rostro de manera automática.

Para la verificación FENIXVR utiliza imágenes estáticas obtenidas ya sea desde una unidad de almacenamiento o capturadas a través de una cámara de video, emplea el Análisis de Componentes Principales o PCA como método para extraer las características más relevantes de los rostros y una red neuronal tipo RBF que es entrenada con dichas características para que ejecute la tarea de verificación.

Para la evaluación del desempeño de esta solución se emplea una base de imágenes creada con este objetivo y como método de evaluación se recurre a la validación cruzada.

PRESENTACIÓN

El reconocimiento de rostros es un área de la biometría que ha sido estudiada desde los inicios de la visión por computador. A pesar de existir otras formas más confiables para la identificación (huellas digitales, escaneo de iris, etc.), el reconocimiento de rostros siempre ha generado gran interés, especialmente por ser poco intrusivo, su implementación requiere una menor inversión que otros sistemas biométricos, además de constituir la forma principal como las personas identifican a otras.

En los últimos años se han desarrollado nuevas formas de reconocimiento de rostros, en gran parte gracias a los avances tecnológicos que han permitido crear algoritmos más complejos, incluso se ha logrado desarrollar sistemas para reconocimiento de rostros en tiempo real, basándose en el gran potencial que tiene el video y en general las herramientas de procesamiento de imágenes con las que se cuenta actualmente. Sin embargo el reconocimiento de rostros a través de un computador tiene hoy por hoy muchas limitaciones y no se ha logrado establecer una aplicación que realice esta tarea de manera confiable bajo diferentes condiciones y circunstancias, tal como lo haría un ser humano.

Actualmente el proceso de reconocimiento de rostros involucra la adquisición de imágenes, la detección del rostro, extracción de característica, y la fase de reconocimiento. En los últimos años el uso de redes neuronales para el reconocimiento de rostros ha sido ampliamente difundido y analizado por su gran poder de generalización, sin embargo esta técnica es sensible a ciertas variaciones, lo que provoca que el nivel de confiabilidad que ofrece no llegue al cien por ciento.

Durante este trabajo se realizará una introducción al procesamiento de imágenes y a los procesos relacionados con el reconocimiento de rostros (Capítulo 1), se analizará y diseñará la solución computacional propuesta (Capítulo 2), se detallarán sus componentes físicos y se evaluará su desempeño (Capítulo 3), finalmente se establecerán conclusiones y recomendaciones (Capítulo 4).

CAPITULO 1. MARCO TEÓRICO

Este capítulo describe algunos conceptos esenciales relacionados al reconocimiento automático de rostros.

La primera sección presenta una introducción al procesamiento digital de imágenes, con el propósito de proveer ciertos fundamentos teóricos necesarios para la comprensión, diseño e implementación de técnicas orientadas al reconocimiento de rostros. La sección 1.2 expone las diversas metodologías que pueden ser aplicadas para el reconocimiento de patrones y para el caso específico de reconocimiento de rostros. La sección 1.3 detalla las fases que involucra el proceso de reconocimiento de un patrón así como también la aplicación de éstas al reconocimiento de rostros. La sección 1.4 resume algunas nociones sobre redes neuronales artificiales. Finalmente, la sección 1.5 presenta un breve análisis de algunas arquitecturas de redes neuronales empleadas frecuentemente para el reconocimiento de rostros.

1.1 CONCEPTOS FUNDAMENTALES SOBRE IMÁGENES DIGITALIZADAS

1.1.1 PERCEPCIÓN VISUAL

La percepción es una característica propia del ser humano mediante la cual adquiere conciencia de la realidad exclusivamente a través de los sentidos.

Si se limita el análisis de las percepciones únicamente al campo visual, se puede decir que es un proceso cognoscitivo de recepción e interpretación de estímulos o impresiones luminosas captadas por los ojos. La recepción implica la aceptación y comprensión de los estímulos captados, en tanto que la interpretación se refiere al análisis y organización racional de dichos estímulos aunque particularizada por la experiencia de cada ser humano, es decir es subjetiva.

El conocer y comprender la estructura y funcionamiento del sistema visual humano ha sido objeto de gran interés en diversas ramas de la ciencia, sin embargo las múltiples investigaciones realizadas no han logrado revelar totalmente el complejo proceso que efectúa el sistema visual. A pesar de esto, la escasa información con la que se cuenta ha contribuido de algún modo al desarrollo de diferentes áreas, como por ejemplo la visión por computador y el procesamiento digital de imágenes.

La visión por computador busca simular la percepción visual. En cuanto al procesamiento digital de imágenes, parte fundamental de la visión por computador, utiliza algunas propiedades y características del sistema visual humano, que es el procesador de imágenes biológico.

La manera en la que se aplica el conocimiento sobre el sistema visual humano en las áreas antes mencionadas, es diversa. Así por ejemplo, existen métodos relacionados a dichas áreas que buscan imitar la percepción visual en ciertos aspectos (tareas perceptuales como percepción de movimiento, de profundidad, textura, entre otras)¹; otros se desarrollan considerando exclusivamente características del sistema visual humano (capacidades y limitaciones) con el objetivo de ser independientes de cualquier aplicación, es decir los requerimientos los fija el sistema visual²; mientras que otros se limitan simplemente a reproducir resultados finales³ ya sea por desconocimiento de cómo se lleva a cabo el proceso en el ser humano o por buscar alternativas que mejoren la calidad de los resultados.

Una breve descripción sobre algunos elementos de la percepción visual se la puede encontrar en el Anexo 1.

¹ Métodos de visión por computador utilizan el modelo HSV para identificar el color de diferentes objetos. Una de las principales razones es que este modelo asemeja la forma como el ser humano percibe el color.

² El formato de compresión de imágenes JPG utiliza técnicas de compresión diseñadas considerando la limitación del sistema visual humano para percibir pequeños cambios en color.

³ Existen muchos métodos utilizados para el reconocimiento de objetos. Sin embargo la forma como estos métodos llevan a cabo esta tarea difiere de la forma como la efectúa el ser humano.

1.1.2 IMAGEN DIGITAL

Una imagen digital es una representación gráfica, generada por computador o creada a través de algún dispositivo de captura (cámara digital, cámara web, escáner, etc.), y que se almacena en forma binaria. Las imágenes digitales pueden ser icónicas (tener diferentes grados de figuración y realismo) o aicónicas (ser abstractas o esquemáticas), bidimensionales o tridimensionales, con o sin movimiento.

Los dispositivos de captura son utilizados comúnmente para obtener imágenes del mundo real. En general toda imagen digital puede ser: ⁴

- Visualizada a través de algún tipo de dispositivo (monitor, cámara digital, etc.)
- Duplicada y copiada con total exactitud tantas veces como se requiera.
- Editada con ayuda de algún software (cortar, pegar, realzar, insertar texto, agregar y suprimir objetos, aplicar efectos especiales, girar y alargar imágenes, etc.)
- Impresa a través de cualquier dispositivo orientado a esa función.
- Distribuida mediante dispositivos de almacenamiento (CD-ROM, DVD, tarjetas de memoria, etc.) o a través de Internet u otro tipo de redes.

La presencia o no de movimiento en la imagen digital permite clasificarla dentro de dos grandes grupos: imagen estática e imagen dinámica.

1.1.2.1 La Imagen Estática

La imagen digital estática es aquella que carece de movimiento, es decir, es una representación de un objeto u objetos, reales o ficticios, en un instante en el tiempo. Este tipo de imagen a su vez se clasifica en dos categorías: imágenes vectoriales e imágenes de mapa de bits o bitmaps.

⁴ Basado en [100].

Adicionalmente existen imágenes compuestas por elementos de las dos clases, se les denomina *metaficheros*.

a. *Imágenes vectoriales*: Se componen de objetos definidos mediante operaciones matemáticas. A estos objetos (líneas, círculos, rectángulos, etc.) también se les conoce como vectores. Cada objeto es una entidad independiente, con propiedades como forma, color, tamaño y posición en el plano; puede tratárselo separado del resto de objetos sin afectarlos, pero si se requiere transformar la imagen completa se asocia a todos los objetos y se aplican los cambios, incluso se pueden hacer rellenos de formas.

Son imágenes que no requieren gran cantidad de espacio para su almacenamiento ya que sólo guardan la información necesaria para generar cada uno de los objetos. También se puede reducir o aumentar su tamaño sin disminuir su calidad, un cambio en tamaño implica el cálculo de la posición y longitud de los objetos pero a diferente escala. Estas características hacen que las imágenes vectoriales sean adecuadas para emplearlas en la ilustración digital y el modelado tridimensional, donde el proceso de creación requiere del tratamiento de objetos por separado y en conjunto, (véase Figura 1.1)

Los programas de diseño asistido por computador (CAD, por sus siglas en inglés) y los programas de gráficos están orientados a la creación y manejo de este tipo de imagen.

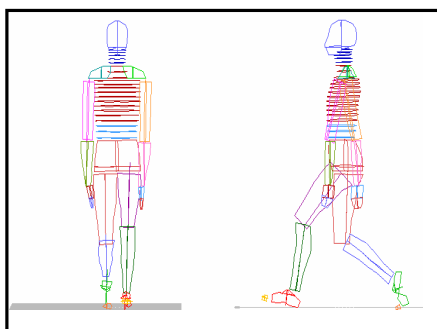


Figura 1.1. Imagen creada mediante primitivas geométricas
Modelado tridimensional para simular el movimiento de un ser humano

El principal inconveniente de las imágenes vectoriales es que no pueden reproducir imágenes fotográficas de un modo eficaz, es decir no logran representarlas con suficiente realismo. En la Figura 1.2 se muestra la transformación de una imagen de mapa de bits a una vectorial, se puede notar al superponerlas que la imagen vectorial no es tan realista como la imagen de mapa de bits.

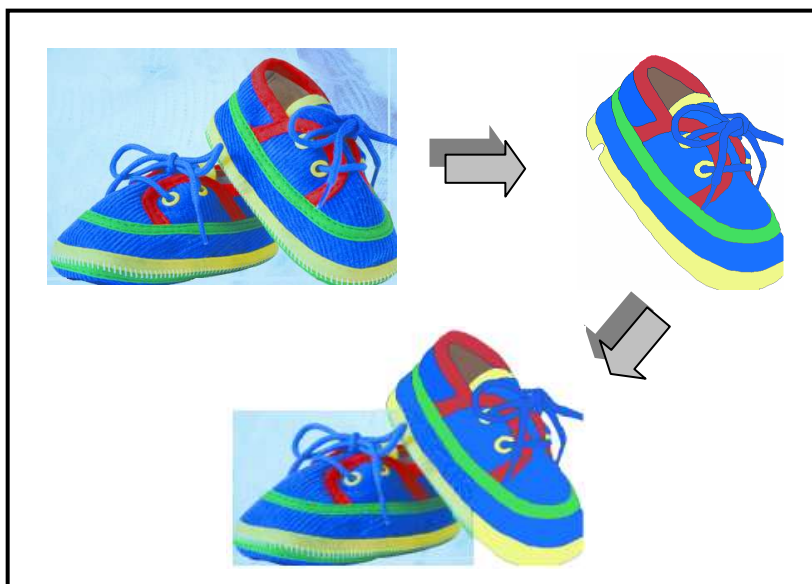


Figura 1.2. Transformación de una imagen bitmap a una imagen vectorial

b. *Imágenes de mapa de bits, bitmaps o imágenes ráster.* Están formadas por matrices de píxeles⁵, cada píxel tiene una posición y color determinado. Este tipo de imágenes permite recoger una amplia gama de colores o tonos, por lo que son adecuadas para representar eficientemente el mundo real. Pueden ser creadas manualmente mediante un programa de pintura o retoque; obtenidas desde documentos con información analógica por medio de escáneres, desde el mundo real por medio de cámaras digitales u otro tipo de dispositivos con similar funcionamiento, y desde cualquier aplicación de software por medio de programas para captura de pantallas.

⁵ Píxel es la abreviatura de la expresión inglesa Picture Element (Elemento de Imagen), se define como la unidad más pequeña o elemento básico de las imágenes compuestas por mapas de bits. No tiene una dimensión concreta.

A diferencia de las imágenes vectoriales, las imágenes bitmap no se pueden desagrupar en los elementos que las forman; cuando son ampliadas se distorsionan es decir pierden calidad o detalle (véase Figura 1.3), y requieren un mayor espacio de almacenamiento ya que deben guardar todos los píxeles que componen la imagen.



Figura 1.3. Imagen de mapa de bits o bitmap se muestra la pérdida de calidad de un área de la imagen cuando se la amplía.

La manera como las imágenes se visualizan en un monitor es a través de píxeles por esta razón tanto imágenes vectoriales como imágenes bitmap se muestran como píxeles. Los programas que manejan imágenes vectoriales convierten las figuras en píxeles para su visualización.

1.1.2.2 La Imagen Dinámica

La imagen dinámica o imagen en movimiento es en realidad un conjunto de imágenes estáticas denominadas cuadros de video que mostrados en secuencia rápida dan la idea de movimiento continuo. La rapidez o velocidad con la que se presenta esta secuencia de imágenes es también conocida como *cuadros por segundo (CPS)* o *frames per second (FPS)*, y está definida de acuerdo a estándares de transmisión, en el caso de PAL (formato usado en Europa) la velocidad de actualización es de 25 cps. y para NTSC (formato usado en América y Japón) es de 30 cps. La calidad de la imagen dinámica depende en gran parte de un método eficiente de compresión que permita reducir la información redundante de los cuadros del video.

1.1.3 DIGITALIZACIÓN DE UNA IMAGEN

Digitalización es el proceso de transformación de una información analógica a digital, esto significa pasar de un espacio continuo a uno discreto.

Para comprender cómo se lleva a cabo este proceso, primero se debe conocer que tipo de información analógica es la que se va a transformar. En el caso de la imagen analógica se la puede definir como una función bidimensional de intensidad de luz $f(x,y)$, donde el valor de f para cada punto de coordenadas (x,y) depende del grado de iluminación de la imagen en ese punto. La imagen analógica se describe en un espacio continuo, es decir f , x e y toman valores continuos.

Por lo tanto la digitalización de una imagen analógica consiste en discretizar la función bidimensional de intensidad de luz $f(x,y)$, esto involucra dos operaciones: el *muestreo de la imagen* que significa dividir la imagen analógica en M filas y N columnas (discretización de las coordenadas espaciales x,y), y la *cuantización* de la amplitud que se refiere a la asignación de un valor discreto (discretización de la intensidad) a cada uno de los $M \times N$ elementos obtenidos en el muestreo (segmentos de la imagen o píxeles). Este proceso se ilustra en la Figura 1.4.



Figura 1.4. Digitalización de una imagen

El resultado final es una matriz formada por píxeles, cada uno de los cuales representa el brillo y color de la imagen en ese punto, es decir se obtiene una imagen digitalizada.

Un ejemplo claro de este proceso es el escaneo de una fotografía, proceso en el que la imagen pasa de un formato analógico a un formato digital.

1.1.4 RESOLUCIÓN DE LA IMAGEN

La resolución de una imagen se define como “el número de píxeles contenidos en la imagen por unidad de longitud. Se mide con píxeles por pulgada (ppp/ppi)”⁶, es decir la cantidad de píxeles que describen a una imagen. Si se tiene por ejemplo una imagen de 300 ppp, en cada pulgada habrá 90.000 píxeles (300 píxeles de ancho x 300 píxeles de alto)⁷.

La calidad de una imagen digital tiene relación directa con la resolución de dicha imagen, ya que a una mayor cantidad de píxeles por pulgada se mostrará mejor los detalles de la imagen, lo que a su vez implica un requerimiento mayor de espacio en disco para su almacenamiento y más tiempo para su edición o impresión. Por el contrario, si una imagen no posee una resolución suficiente tendrá un tamaño de píxel más grande que el deseado y por ende menos detalle, transiciones de color menos suaves y aspecto más dentado. Este efecto puede apreciarse en la Figura 1.5, todas las imágenes tienen el mismo tamaño pero no la misma resolución, por lo que las imágenes con menor resolución necesitan aumentar el tamaño de sus píxeles para llenar toda su área.



Figura 1.5. Ejemplos de una imagen con diferentes resoluciones (a) a 96 ppp. (b) a 10 ppp y (c) a 5 ppp.

⁶ Ver [97].

⁷ Una pulgada equivale a 2,54 cm. Por lo que si se expresa la resolución en centímetros, una imagen de 300 ppp (píxeles por pulgada) tendría 118,11 píxeles por centímetro, es decir 13.950 píxeles aproximadamente en un centímetro cuadrado de la imagen.

1.1.4.1 Dimensiones de Píxel

Son las medidas horizontales y verticales de una imagen expresadas en píxeles. Para obtener estas dimensiones basta conocer el tamaño y resolución de la imagen, únicamente se debe multiplicar el alto o el ancho de la imagen por la resolución de ésta. Por ejemplo, si se tiene una imagen de 12 x 12 pulgadas con una resolución de 300 ppp, sus dimensiones de píxel son 3.600 x 3.600.

Es importante aclarar que el *tamaño de la imagen* es diferente a la resolución de la imagen. Como se explicó anteriormente, la resolución se refiere al número de píxeles que contiene la imagen por cada pulgada, al contrario el tamaño de la imagen se refiere a las dimensiones absolutas (ancho y alto) de la imagen expresadas en unidades de longitud (generalmente expresadas en pulgadas o centímetros).

1.1.4.2 Profundidad del Píxel

La profundidad del píxel determina el número de bits que un píxel puede almacenar, también se la conoce como resolución de bits o profundidad del color. Se encuentra estrechamente relacionada con la nitidez de la imagen digital, ya que a una mayor profundidad se tendrá un mayor nivel de tonos, lo que permite una representación más real de la imagen.

Así un píxel puede tener profundidad de 1, 2, 4, 8, 16, 24 y 32 bits, aunque los valores normalmente utilizados son de 8, 16 o 24.

Un píxel con profundidad de 1 bit puede representar dos tonos (blanco y negro), un píxel con profundidad 8 tiene 256 valores posibles (2^8), uno con profundidad de 24 tiene 16'777.216 valores posibles (2^{24}) en este caso se habla de *color verdadero* debido a que se asemeja a lo que el ojo humano puede distinguir en el mundo real. Para el caso de una profundidad de 32 bits en realidad no se añaden más niveles de tonos

que los definidos con 24 bits, lo que se hace con los últimos 8 bits es indicar el grado de transparencia (desde 0 para totalmente transparente hasta 255 para totalmente visible).

Una profundidad de 1, 2 o 4 bits resulta limitada para imágenes de video debido a la baja calidad que ofrecería. En el caso de las imágenes de video monocromáticas generalmente se trabaja con profundidad de 8, es decir 256 tonalidades de grises que varían gradualmente.

Una imagen a color está representada por una profundidad de 8 y 24 bits; para el caso de 24 bits por lo general se los divide en tres bytes: 8 bits para el color rojo, 8 para el verde y 8 para el azul, para representar otros colores se realizan combinaciones de éstos (RGB).

La Figura 1.6 muestra una imagen con diferentes profundidades de píxel, se puede observar que mientras mayor sea el número de bits, la imagen se presenta más nítida.

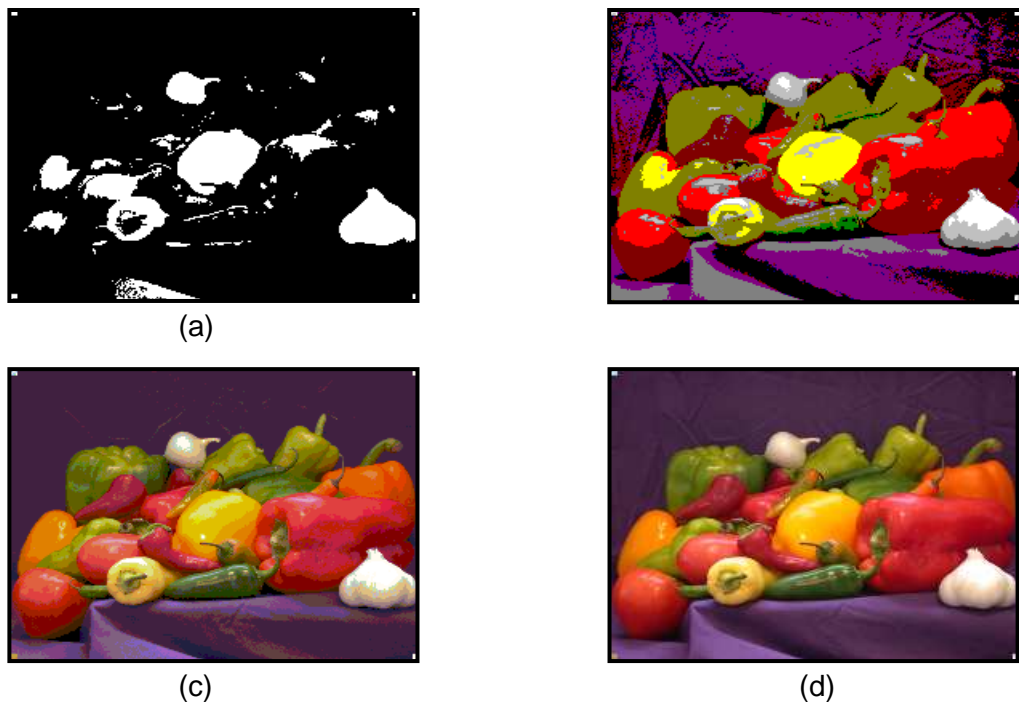


Figura 1.6. Ejemplos de una imagen con diferentes profundidades de píxel
(a) 1 bit (b) 4 bits (c) 8 bits y (d) 24 bits.

1.1.5 RELACIONES BÁSICAS ENTRE PÍXELES

El procesamiento digital de imágenes se refiere a la manipulación de éstas con el objetivo de mejorarlas desde algún punto de vista y con algún fin. Muchas de las operaciones que se pueden o se requieren realizar dentro del procesamiento involucran áreas locales de píxeles que combinadas de diversas maneras permiten obtener un resultado determinado.

A continuación se mencionan algunas de las relaciones básicas entre píxeles así como también ciertas operaciones de tipo aritmético/lógico comúnmente empleadas dentro del procesamiento de imágenes.

a. *Vecinos de un píxel.* Son píxeles que se encuentran ubicados a cierta distancia de éste y que comparten un lado o un vértice con dicho píxel.

b. *Conectividad.* Describe una relación entre dos o más píxeles, su objetivo es establecer los límites o bordes de objetos así como los objetos existentes en una imagen.

c. *Etiquetado de componentes conexas.* Una componente conexa es el conjunto de píxeles en una imagen que están conectados entre sí. El etiquetado de componentes conexas se refiere al proceso de determinar todas las componentes existentes en una imagen y asignar a cada una de ellas una etiqueta distintiva.

d. *Medidas de distancia.* Permiten determinar cuan cerca o lejos se encuentran dos píxeles en una imagen. Algunas de las medidas de distancia comúnmente empleadas son:

- Distancia Euclidiana, Manhattan, Distancia de Tablero de Ajedrez

e. *Operaciones aritmético/lógicas.*

- i. Aritméticas: Adición, sustracción, multiplicación, división.
- ii. Lógicas: AND, OR, Complemento.

En el Anexo 3 se describe con mayor detalle las relaciones, distancias y operaciones aritmético/lógicas antes citadas.

1.1.6 TRANSFORMACIONES ESPACIALES

Las transformaciones espaciales realizan un mapeo de los píxeles desde su posición en la imagen inicial hacia una nueva posición en la imagen resultante. A continuación se enumeran algunas de las transformaciones básicas, las mismas que se describen con mayor detalle en el Anexo 4.

a. *Transformación de Escala.* Mediante esta transformación se puede reducir o aumentar el tamaño de una imagen.

b. *Traslación.* Esta transformación geométrica permite el desplazamiento de los píxeles de una imagen en una dirección determinada.

c. *Rotación.* Transformación geométrica que mapea la posición (x_1, y_1) de un píxel en una imagen inicial hacia una posición (x_2, y_2) en una imagen resultante, para esto utiliza un ángulo de rotación y un origen.

d. *Transformación Affine.* Transformación geométrica en la que los valores de los píxeles de la imagen original son mapeados a una imagen final o resultante mediante la aplicación de una combinación de otras transformaciones tales como: traslación, rotación y escalamiento.

e. *Extracción de áreas de la imagen.* Proceso mediante el cual se remueven ciertos píxeles en una imagen ya sea para eliminarlos o utilizarlos como una nueva imagen

1.1.7 COLOR DE LA IMAGEN

La idea del color se relaciona con el proceso de descomposición de la luz. Los colores que el ser humano percibe en un objeto vienen determinados por las longitudes de onda reflejadas por el objeto (luz que incide en éste), por lo tanto el color no es una propiedad intrínseca de los objetos. De lo antes expuesto se puede concluir que dentro del estudio del color, el fenómeno de la luz tiene un papel fundamental ya que sin este fenómeno sería imposible la percepción de color así como de objetos. En el Anexo 2

se detallan ciertas características y definiciones relacionadas a dicho fenómeno.

1.1.7.1 Especificación del Color

Los principales atributos que diferencian a un color de otro son: el matiz, el brillo y la saturación, (véase Figura 1.7).

- a. *El Matiz o tono*: Describe la pureza de un color, también se lo define como la longitud de la onda luminosa o simplemente el color percibido por un observador (verde, azul, rojo, etc.)
- b. *El Brillo, valor o luminosidad*: Se refiere a la intensidad lumínica, es decir a la cantidad de luz que tiene.
- c. *La Saturación*: Se refiere al grado en el que un color puro se diluye con luz blanca, esto significa que indica la concentración de color. Una saturación nula corresponde a una ausencia de color, que es el caso de la escala de grises.

El matiz y la saturación reciben en conjunto el nombre de *cromaticidad o cromaticidad*.

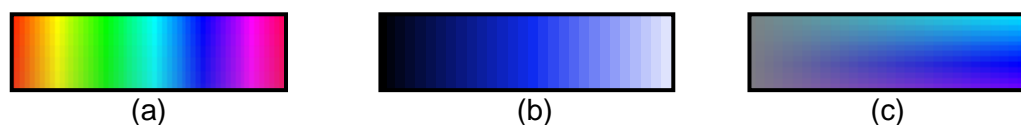


Figura 1.7. Especificación del color (a) escala de matices (b) escala de brillo (para el matiz azul) (c) escala de saturación (para el matiz azul)

1.1.7.2 Síntesis Aditiva y Sustractiva

La síntesis aditiva y la sustractiva son formas que permiten sintetizar o producir color.

- a. La *síntesis aditiva* consiste en mezclar colores agregando luz, de tal forma que los colores resultantes de la mezcla son más luminosos que

los colores iniciales. Los colores primarios de este sistema son el azul, verde y rojo; al mezclarlos se obtienen los colores secundarios que son amarillo, magenta y cyan. Este proceso de adición puede continuar hasta llegar al color blanco, (véase Figura 1.8 (a)).

- b. La *síntesis sustractiva* por su parte genera diferentes colores al restar luz, es decir es el proceso inverso a la síntesis aditiva. Así, los colores fundamentales en este proceso son amarillo, magenta y cyan, quedando como secundarios el azul, verde y rojo; siendo el color negro el resultado final, (véase Figura 1.8 (b)).



Figura 1.8. Síntesis del Color: (a) síntesis aditiva (b) síntesis sustractiva.

1.1.7.3 Modelos de Color

Las propiedades del color se pueden definir matemáticamente mediante modelos, que son representaciones n-dimensionales únicas de los colores a fin de facilitar su especificación en algún formato estándar. Existe un gran número de modelos de color, sin embargo el uso de un modelo u otro está más bien relacionado con el tipo de aplicación al que está orientado. En cuanto al tipo de aplicación se puede decir que existen dos campos dentro de los cuales se encasillan la mayoría de los modelos de color que existen en la actualidad, el área del hardware y el procesamiento de imágenes en color con diversos fines.

Los modelos orientados hacia aplicaciones de hardware que son comúnmente usados son: el RGB utilizado para monitores, cámaras

digitales y escáneres, el CMY utilizado para impresoras de color y el YIQ que es estándar de las emisoras de televisión.

Para el área de procesamiento de imágenes se encuentran el modelo HSV, HSI, L^*a^*b , NRGB, entre otros. Modelos como el RGB también son utilizados en esta área.

El Anexo 5 contiene una breve descripción de algunos de los modelos de color más comúnmente empleados.

1.1.8 TRANSFORMADAS DE LA IMAGEN

Las transformadas son operaciones que convierten imágenes desde una representación hacia otra diferente (una representación matemática alternativa) con el objetivo de hacer más evidente algún tipo de información existente en la imagen. Generalmente luego de una transformación sigue un proceso de umbralización para extraer las características más relevantes de la nueva representación.

De una manera más formal, se puede decir que “el principio de una transformada es convertir una función, perteneciente a un espacio vectorial, en otra función del mismo espacio o de otro espacio, por medio de varias funciones denominadas bases del espacio vectorial.”⁸

A continuación se describen de manera general algunas transformadas que se emplearon como parte del presente trabajo, adjuntamente el Anexo 6 refiere ciertas transformadas utilizadas frecuentemente dentro del procesamiento digital de imágenes, detallando específicamente la forma discreta bidimensional de cada una de ellas, por tal razón no se debe considerar como un análisis completo de dichas transformadas.

⁸ Ver [4].

1.1.8.1 Transformada de Hough

Detectar un objeto en una imagen es una tarea compleja, en la literatura existen muchos métodos para llevar a cabo este proceso [10, 14, 84].

Uno de los métodos más comunes es la correlación⁹ entre la imagen y una máscara que presenta la forma y tamaño del objeto que se busca. Sin embargo la mayor desventaja de este método es que dicha máscara generalmente difiere demasiado del objeto, especialmente por efectos de oclusión, rotaciones, variaciones de escala, entre otros.

Así la transformada de Hough se presenta como un método efectivo para la segmentación de imágenes y reconocimiento de patrones, en especial por su capacidad de detectar objetos sobrepuestos, parcialmente ocluidos o en presencia de ruido.

La transformada de Hough originalmente fue diseñada para detectar líneas rectas y curvas cuyas ecuaciones analíticas sean conocidas, a este método se lo denomina *Transformada Estándar de Hough* y fue propuesto por Paul Hough quien lo patentó en 1962. A través del tiempo se han realizado modificaciones en la forma original de la transformada con el objetivo de reducir el costo computacional que ésta demanda, para incrementar su capacidad de detección bajo condiciones complejas o para ampliar el tipo de patrones que puede detectar. Actualmente existen diversas variaciones de la transformada entre las cuales se pueden mencionar la Transformada de Hough Adaptiva, Transformada de Hough Aleatoria, Transformada de Hough Difusa, Transformada de Hough Generalizada, entre otras.

Dentro del procesamiento de imágenes una de las variaciones que más se emplea es la Transformada de Hough Generalizada [24, 75], fue desarrollada por Richard Duda y Peter Hart en 1972, esta transformada

⁹ Para mayor detalle véase Sección 1.2.1.1

permite detectar objetos cuyo borde no tiene una expresión analítica definida.

Entre las principales aplicaciones de la transformada de Hough se pueden destacar: investigación geográfica de imágenes satelitales, investigación médica en imagenología, control de movimiento de robots por sensores, reconocimiento de patrones, entre otras.

1.1.8.1.1 Principio de la Transformada de Hough

Si se desea detectar en una imagen objetos que tengan una forma determinada se debe primero definir los parámetros que caracterizan a esa forma y luego aplicar la transformada de Hough.

El principio de esta transformada consiste en que para una imagen dada hay un gran número de figuras potenciales que pasan por uno o varios puntos de los contornos o bordes de los objetos de la imagen, y que pueden ser descritas o caracterizadas por los parámetros previamente establecidos (cada figura tendrá una combinación diferente de valores para los parámetros). Por lo tanto el objetivo de la transformada es determinar cuáles de estas figuras potenciales realmente encajan con objetos que existen en la imagen.

Para cumplir este objetivo, la transformada recurre al uso de una estructura o *acumulador* en el cual se registran *votos*. De acuerdo a esta técnica de votos cada píxel o punto que forma parte del contorno de un objeto contribuye con su voto a todas las combinaciones posibles (valores de los parámetros) que describen a las figuras potenciales de las que ese píxel puede formar parte.

Cada combinación está asociada a una celda del acumulador, por lo tanto las celdas con mayor número de votos proporcionan los valores que se deben asignar a los parámetros para que describan a las figuras

que mejor encajan con aquellos objetos de la imagen que tienen una forma similar a la que se busca. Esto se produce porque un mayor número de píxeles de los contornos votaron por esa combinación de valores para los parámetros, (véase Figura 1.9).

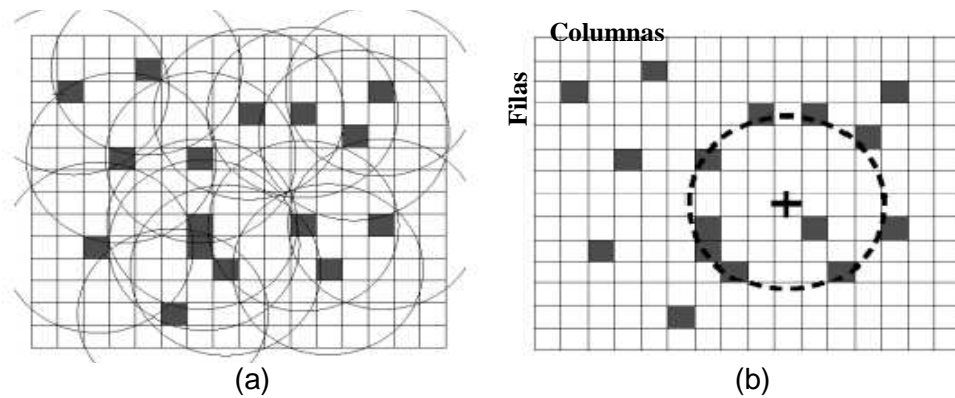


Figura 1.9. Principio de la Transformada de Hough.

Se busca una circunferencia cuyo radio es conocido, los parámetros para los cuales se realiza la votación son las coordenadas (f,c) del centro de la circunferencia. (a) figuras potenciales que pasan por uno o varios puntos de borde (b) figura con mayor número de votos en el espacio de parámetros.

Fuente: YOUNG, D. Lecture 6. 2006. Pág. 5.

El acumulador es una estructura que permite discretizar el *espacio de los parámetros*, este espacio está compuesto por uno o varios acumuladores a los cuales se los denomina también *células acumulador*. En síntesis se puede decir que la transformada de Hough realiza un mapeo desde el espacio de la imagen (dominio espacial) hacia el espacio de los parámetros a fin de detectar dentro de este nuevo espacio picos máximos (mayor votación) que permitan definir los valores de los parámetros que mejor describen al objeto que se busca. Aunque la transformada de Hough presenta grandes ventajas para la detección de patrones, su mayor desventaja está en el crecimiento exponencial de su espacio en función del número de parámetros que representan al patrón buscado, lo que hace que esta transformada sea habitualmente utilizada para patrones que pueden ser definidos con pocos parámetros. Así un patrón con n -parámetros requiere un espacio de parámetros n -dimensional.

El algoritmo básico de la transformada de Hough, sin considerar ningún tipo de optimización es el siguiente:

1. Encontrar todos los puntos de borde o contornos de los objetos presentes en la imagen (Detección de bordes).
2. Para cada combinación posible de los parámetros que definen al patrón.
3. Por cada punto de borde
4. Mapear el punto de borde a una celda en el acumulador, en función de los parámetros previamente definidos.
5. Incrementar dicha posición en el acumulador.
6. Encontrar máximos locales en el acumulador.
7. Si se requiere, mapear cada máximo del acumulador hacia el espacio de la imagen.

El algoritmo antes descrito es sólo una forma general de aplicar esta transformada, no obstante existe en la literatura una gran variedad de algoritmos que emplean diferentes técnicas de optimización [24, 87].

En el Anexo 7 se sugieren algunas consideraciones generales a tener en cuenta cuando se utiliza la transformada de Hough. También se incluye una descripción sobre el uso de esta transformada para la detección de elipses, el principal motivo para analizar este caso particular es su aplicación en áreas como la detección de rostros [56, 77] que para el presente trabajo se considera de interés.

1.1.8.2 Análisis de Componentes Principales (PCA)

El Análisis de Componentes Principales (PCA, por sus siglas en inglés) también es conocido como la transformada Karhunen-Loeve o la transformada de Hotelling.

Su objetivo principal es la reducción de dimensionalidad de un conjunto de datos mediante la extracción de características o componentes principales que retengan tanto como sea posible la información relevante acerca de los datos originales, maximizando la variación entre ellos.

Una de las principales características de esta transformación es que minimiza el error de reconstrucción en función de los componentes utilizados. A los componentes se los denomina como *autovectores* o *eigenvectores* y en conjunto caracterizan la variación entre los datos analizados, de tal manera que cada dato inicial puede ser representado exactamente en términos de una combinación lineal de los eigenvectores o a su vez puede ser aproximado utilizando únicamente los eigenvectores más representativos que son aquellos asociados con los *autovalores* o *eigenvalores* más grandes y por ende cuentan con la mayor variación. Por esta razón se puede decir que la utilidad de un eigenvector para caracterizar la variación entre los datos iniciales está dada por el eigenvalor asociado a éste.

De una manera más formal los eigenvectores se definen como los vectores base del espacio de características y corresponden a las direcciones de máxima variación en el espacio original. Así, la primera dirección corresponde a la máxima variación, la segunda a la siguiente variación más grande y así sucesivamente.

Dentro del procesamiento de imágenes PCA tiene aplicaciones como la compresión, análisis, visualización de imágenes, reconocimiento de patrones, entre otros.

1.1.8.2.1 PCA en el reconocimiento de patrones

Una de las aplicaciones más comunes del PCA dentro del procesamiento de imágenes es el reconocimiento de patrones.

A continuación se explicará brevemente la forma cómo se utiliza la transformada en este campo. En cuanto a la teoría y la base matemática que la respaldan están descritas en detalle en la literatura [68, 15, 49] por lo tanto no serán analizadas dentro del presente trabajo.

El proceso de detección de patrones puede ser difícil de llevar a cabo en espacios de alta dimensionalidad por esta razón se utiliza inicialmente PCA para mapear los datos originales¹⁰ hacia un espacio de menor dimensionalidad denominado espacio de características¹¹.

Luego de esta reducción de dimensionalidad se tiene un conjunto de eigenvectores calculados en función de las imágenes iniciales (datos originales). De tal manera que al utilizar una nueva imagen para saber si es similar o no al conjunto de imágenes iniciales y con cuál de ellas tiene mayor relación se mide la diferencia entre la nueva imagen y el conjunto inicial pero no utilizando los ejes originales ya que sería muy complejo sino a través de los nuevos ejes derivados a partir del PCA (eigenvectores).

Este proceso se justifica en el hecho que PCA presenta a las imágenes originales en términos de las diferencias y similitudes que tienen, por lo que si la nueva imagen posee ciertas características similares con el conjunto entonces se puede concluir que forma parte de éste; del mismo modo, tendrá más diferencias con algunas imágenes que con otras, por lo que con aquella que tenga mayor similitud será con la imagen que esté relacionada (si la similitud alcanza un valor umbral).

Se pueden establecer los siguientes supuestos que sirven como base para el reconocimiento de patrones a través de PCA ¹² :

¹⁰ Conjunto de imágenes que representan a una clase de objetos.

¹¹ Con esta transformación los cálculos se reducen considerablemente, del orden asociado al número de píxeles en las imágenes ($N \times N$) al orden asociado al número de imágenes que forman el conjunto inicial (M), en la práctica $M \ll N \times N$.

¹² Extraído de [73] página 5.

- Imágenes de un objeto particular ocupan una región relativamente pequeña pero delimitada en el espacio de las imágenes.
- Diferentes objetos ocupan diferentes regiones en el espacio de las imágenes.
- Clases enteras de objetos ocupan una región todavía relativamente pequeña pero delimitada en el espacio de las imágenes.

El algoritmo básico para el cálculo de eigenvectores dentro del reconocimiento de patrones se describe a continuación¹³ :

1. Obtener el conjunto de imágenes a partir del cual se va a crear el nuevo espacio. Todas las imágenes deben tener las mismas dimensiones.
2. Codificar cada imagen como un vector columna (se concatenan las columnas de la imagen).
3. Hallar el vector promedio del conjunto de vectores.
4. Normalizar o centrar cada vector del conjunto (restar de cada vector el vector promedio).
5. Agrupar los vectores normalizados en una matriz.
6. Calcular la matriz de covarianza.
7. Calcular los eigenvectores de la matriz de covarianza
 - a. Debido a que la matriz de covarianza es generalmente demasiado grande para el cálculo de los eigenvectores se trabaja con:
 - i. Sea A la matriz obtenida en el paso 5, se calculan los eigenvectores de $A^T A$ (se omite el paso 6).
8. Ordenar los eigenvectores resultantes en forma descendente de acuerdo a los eigenvalores asociados.
9. Calcular los m mejores eigenvectores de la matriz de covarianza a partir de los eigenvectores obtenidos y ordenados en los pasos 7 y 8
 - a. Sea A la matriz obtenida en el paso 5 y v_i los eigenvectores calculados en el paso 7 ya ordenados.

¹³ Basado en [74].

$u_i = Av_i$, donde u_i son los m mejores eigenvectores de la matriz de covarianza.

10. Normalizar los eigenvectores del paso 9 ($\|u_i\| = 1$).

11. Seleccionar el número de eigenvectores que se van a utilizar (eigenvectores que corresponden a los eigenvalores más grandes), el conjunto de eigenvectores seleccionados constituye el espacio de características o eigenespacio.

Una vez calculados los eigenvectores la fase de reconocimiento de patrones mediante PCA involucra:

a. *Representación*: Una imagen puede ser representada como una combinación lineal de los eigenvectores, asignando a cada uno de éstos un peso que especifica en qué grado el eigenvector está presente en la imagen. Se dice que el cálculo de estos pesos (vector de pesos) es la proyección de la imagen en el eigenespacio.

b. *Reconocimiento*: Finalmente el reconocimiento de patrones propiamente dicho se lleva a cabo comparando los vectores de pesos de las imágenes que forman el conjunto inicial con el vector de pesos de una nueva imagen. Existen muchos métodos para realizar esta comparación, los mismos que van desde la utilización de medidas de distancia hasta el uso de redes neuronales. Estos métodos se explicarán más adelante.

El Anexo 8 es un fragmento del documento "*Eigenfaces for Recognition*" de *Matthew Turk* y *Alex Pentlad*, donde se presenta el uso de PCA para reconocimiento de rostros (pasos para calcular los eigenvectores, para proyectar imágenes en el eigenespacio, para realizar el reconocimiento de rostros utilizando los vectores de pesos y la aplicación de PCA para detección de rostros.)

1.1.9 MEJORA DE LA IMAGEN

El mejorar una imagen tiene como objetivo resaltar o corregir determinados aspectos de la misma, estos aspectos están en función del tipo de aplicación para el que se requiere dicha imagen y de las condiciones bajo las que se adquiere la imagen. Por esta razón las técnicas utilizadas para mejora o realce no pueden ser aplicadas de la misma manera en todos los casos de procesamiento de imágenes. Así una técnica que favorezca a un tipo de imagen puede no ser útil en otro tipo de imagen.

A continuación se expondrán algunas de las técnicas más generales, sin embargo es importante mencionar que existe una gran variedad de técnicas, muchas de ellas han sido desarrolladas para casos particulares.

1.1.9.1 Modificación del Histograma

El histograma de una imagen muestra la cantidad de píxeles que hay en ésta por cada valor de luminosidad, donde el extremo izquierdo del histograma representa al color negro y continúa hasta el extremo derecho que representa al color blanco.

La utilidad del histograma radica en que proporciona una descripción global de la apariencia de la imagen por lo que puede ser utilizado como punto de partida para eliminar ciertas tonalidades, mejorar el contraste, obtener información sobre el brillo en la imagen, ente otros. Además una de las mayores ventajas de trabajar sobre el histograma en lugar de trabajar sobre la imagen, es la reducción de la carga computacional ya que se pasa de una estructura bidimensional con $M \times N$ valores a una unidimensional con 2^n valores (n es el valor asociado con la profundidad del píxel de una imagen).

Las técnicas relacionadas con la modificación del histograma están enfocadas especialmente a mejorar la visualización de una imagen.

Existen dos técnicas muy comunes para la modificación de histogramas:

- *Estiramiento del Histograma*: Es la forma más simple de modificación de un histograma. Consiste en buscar una función lineal que ajuste el rango de valores inicial (valores presentes en la imagen) hacia un nuevo rango.

Sea $(NI_{\min} - NI_{\max})$ el rango de valores inicial, se transforma a $(NF_{\min} - NF_{\max})$ tal que $NI_{\max} = NF_{\max}$ y $NI_{\min} = NF_{\min}$. El resto de valores que conforman el rango inicial son transformados en otros valores según la transformación lineal utilizada.

Un caso particular de esta técnica, es el *estiramiento localizado*, que se aplica de la misma manera pero no a todo el rango de valores sino a un sub-rango determinado, incluso se pueden aplicar diferentes estiramientos localizados con diferentes funciones lineales en diferentes áreas del histograma.

- *Ecuación del Histograma*: Consiste en redistribuir las intensidades de los píxeles de la imagen de manera uniforme en todo el rango posible del histograma. Esto significa que a diferencia de la técnica de estiramiento donde sólo se consideran los valores máximo y mínimo, la ecualización considera la frecuencia o número de píxeles en cada valor para modificar el histograma de manera que tienda a ser más parejo. La ecualización del histograma aumenta el contraste en la imagen. De igual manera existe un caso particular de esta técnica, la *ecualización adaptativa*, que divide a la imagen en pequeñas áreas y para cada una de éstas realiza una ecualización del histograma asociado.

La Figura 1.10 muestra los cambios que sufre una imagen al modificar su histograma. En (a) la distribución de los píxeles se concentra en un rango de valores muy pequeño por lo que la imagen no tiene mayor contraste. En (b) y (c) se puede observar que al modificar el histograma por

cualquiera de los métodos se logra mejorar el contraste de la imagen, sin embargo la modificación realizada mediante la ecualización en este caso produce mejores resultados.

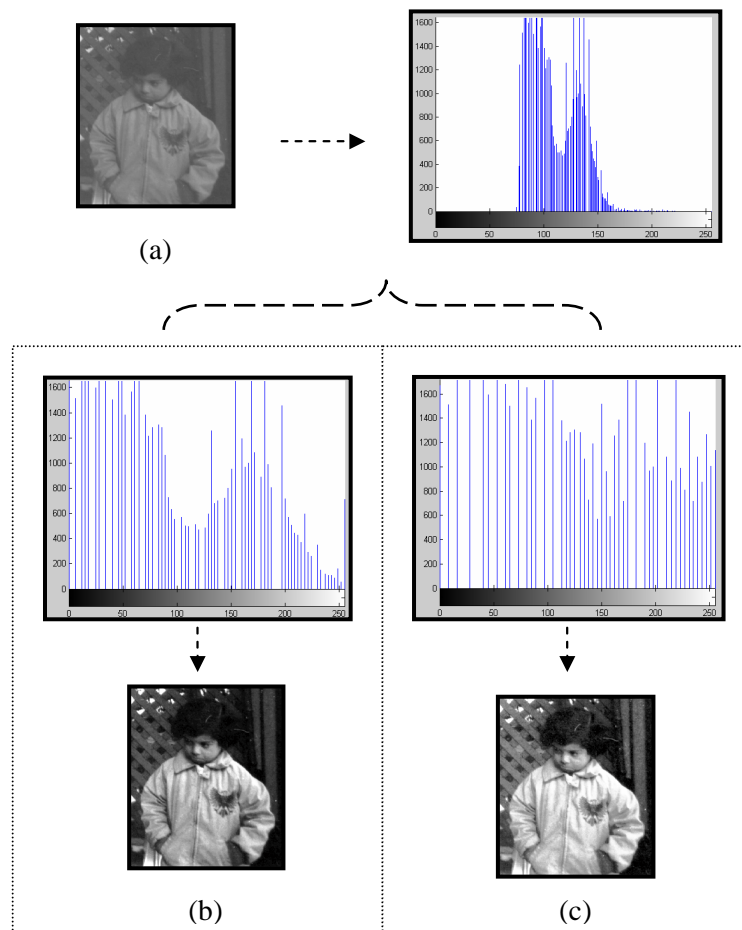


Figura 1.10. Modificación del histograma de una imagen (a) imagen inicial y su histograma (b) estiramiento del histograma (c) ecualización del histograma.

1.1.9.2 Sustracción y Promediado de imágenes

Son operaciones aritméticas que se pueden efectuar sobre las imágenes. A continuación se describe cada una de ellas:

- La *sustracción* en imágenes se encuentra detallada en el Anexo 3. Esta técnica suele ser utilizada para detectar movimiento o mostrar un cambio producido en una escena. Por ejemplo, si se toma una imagen de

una habitación vacía, luego se toma otra imagen de la misma habitación pero añadiendo un objeto. El resultado será que sólo las áreas que son diferentes entre las dos imágenes aparecen de forma más notoria en la imagen resultante, en este caso será el objeto agregado el que aparece de forma evidente, mostrando el cambio producido en la escena.

- La otra operación aritmética consiste en *promediar* un conjunto de imágenes a fin de obtener una imagen final en la que prevalezcan las características comunes entre las imágenes del conjunto. Esta técnica es muy utilizada para la reducción de ruido cuando se cuenta con una secuencia de imágenes.

1.1.9.3 Perfilado

Esta técnica remarca los contornos de los objetos presentes en una imagen con el fin de hacerlos más perceptibles, para esto incrementa el contraste entre los píxeles provocando que los bordes brillantes se hagan uniformemente más brillantes y los bordes oscuros uniformemente más oscuros.

El uso excesivo del perfilado en una imagen puede provocar efectos como la amplificación de ruido, contornos poco naturales e incluso que la imagen se segmente completamente, (véase Figura 1.11).



Figura 1.11. Perfilado de una imagen (a) imagen inicial (b) imagen perfilada una vez (c) imagen sometida a un excesivo perfilado

1.1.9.4 Filtrado Espacial y Filtrado en el Dominio de la Frecuencia

El objetivo del filtrado es mejorar o suprimir detalles presentes en una imagen. De acuerdo al espacio en el que se aplica se puede clasificar en filtrado espacial y filtrado en el dominio de la frecuencia.

a. Los *filtros espaciales* se aplican directamente a la imagen y no a una transformada de ésta. Al ser una operación que se lleva a cabo directamente en la imagen su resultado estará en función de los valores de los píxeles, por lo tanto se puede definir al filtrado espacial como una técnica mediante la cual el valor de cada píxel en una imagen es modificado de acuerdo a los valores de los píxeles que le rodean¹⁴ para así obtener cierto resultado deseado. Es entonces el tipo de resultado que se desea obtener el factor que condiciona el tipo de filtro que se va a utilizar. Así se pueden definir dos clases de filtros espaciales: filtros suavizantes y filtros de realce.

b. Los *filtros en el dominio de la frecuencia* son aquellos que se aplican en una transformada de la imagen, en este caso en el dominio de la frecuencia que como se ha mencionado anteriormente resulta de la aplicación de la transformada de Fourier sobre una imagen.

Una vez que se ha realizado la transformación hacia el dominio de la frecuencia resulta más intuitiva la selección de componentes de alta y baja frecuencia, siendo éstos la base para diferenciar los tipos de filtros dentro de esta clase: filtros paso bajo y filtros paso alto.

La Tabla 1.1 presenta una síntesis de los tipos de filtros antes mencionados, describiéndolos brevemente:

¹⁴ El filtrado espacial emplea como filtros máscaras o ventanas, que son pequeñas distribuciones bidimensionales (de por ejemplo 3x3), en las que los valores de los coeficientes determinan la naturaleza del proceso (tipo de resultado). La ventana o filtro se centra en el píxel cuyo valor va a cambiar en función de los píxeles cercanos (píxeles que se encuentran dentro del área que cubre la máscara). [50]

Filtrado Espacial (*)	
<i>Filtros Suavizantes</i>	<ul style="list-style-type: none"> ○ Filtros Paso Bajo: Atenúan las frecuencias altas mientras mantienen las bajas y medias.
(efecto de difuminado o para eliminar ruido)	<ul style="list-style-type: none"> ○ Filtro de Media: Su objetivo es la reducción de ruido sin difuminar los contornos, para esto reemplaza el valor de cada píxel por la mediana de los valores de los píxeles cercanos a éste (el número de píxeles considerados depende del tamaño del filtro utilizado).
<i>Filtros de Realce</i>	<ul style="list-style-type: none"> ○ Filtros de Paso Alto: Buscan desaparecer las frecuencias bajas y dejar sólo los componentes de alta frecuencia.
(resaltar características de una imagen aumentar el contraste)	<ul style="list-style-type: none"> ○ Filtros de Enfatización: Realzan las frecuencias altas de una imagen sin perder la contribución de las frecuencias bajas. ○ Filtros de Derivadas: Se basan en la derivación de la imagen y son utilizados para la detección de bordes. La forma más común para calcular la derivada en un punto de la imagen es el cálculo de su gradiente. Entre los filtros creados con este objetivo se pueden mencionar: Laplaciano, Robert, Prewitt, Sobel, etc.
Filtrado en el dominio de la Frecuencia	
<i>Dominio de la frecuencia</i>	<ul style="list-style-type: none"> ○ Filtros Paso Bajo: Atenúan un rango específico de componentes de alta frecuencia, con el objetivo de difuminar o suavizar.
(Transformada de Fourier)	<ul style="list-style-type: none"> ○ Filtros Paso Alto: Atenúan los componentes de baja frecuencia sin modificar la información existente en los componentes de alta frecuencia. Su objetivo es el realce de la imagen.

Tabla 1.1. Clasificación de los filtros espaciales y del dominio de la frecuencia

(*) Si se consideran las variaciones de los niveles de gris entre los píxeles de una imagen en función a su distancia, se puede establecer que las variaciones rápidas (bordes y detalles muy marcados) están asociadas con componentes de alta frecuencia, en tanto que las transiciones suaves se asocian con componentes de bajas frecuencias.

1.1.10 RESTAURACIÓN DE IMÁGENES

La restauración busca recuperar la información original de una imagen conociendo las causas de la degradación que ésta ha sufrido; es en base a esas causas que se debe seleccionar la técnica más adecuada para la restauración.

Las causas de degradación son muy variadas sin embargo se pueden agrupar en: artefactos de movimiento, distorsiones geométricas, problemas de enfoque y contaminación por ruido.

Las técnicas empleadas para el proceso de restauración son muy diversas, sin embargo en el presente trabajo sólo se considerará la técnica de *filtro de Wiener* por ser un filtro clásico para restauración de imágenes además de ser utilizado como parte de este trabajo.

1.1.10.1 Filtro de Wiener

Intenta minimizar el error cuadrático medio entre la imagen restaurada y la imagen inicial, (véase Figura 1.12). Si no se conoce cómo es el ruido que afecta a la imagen se lo puede calcular a partir de una imagen sin contenido o se lo puede estimar (gaussiano, ruido blanco, etc.).

Este filtro es generalmente aplicado en el dominio de la frecuencia.

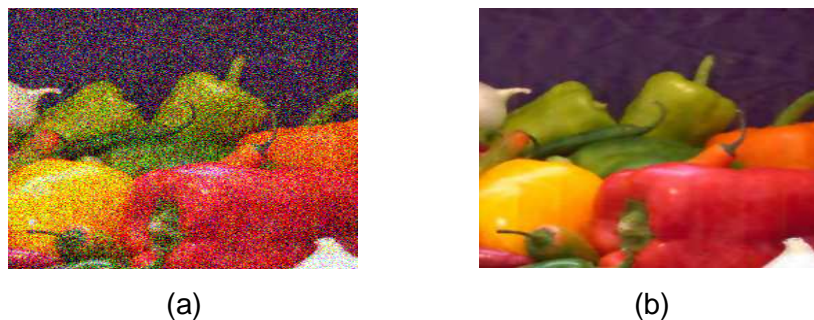


Figura 1.12. Filtrado de Wiener para remover ruido en una imagen (a) imagen inicial con presencia de ruido (b) imagen luego de aplicar el filtrado.

1.1.11 SEGMENTACIÓN DE IMÁGENES

La segmentación es un proceso que permite extraer información de una imagen; consiste en dividir a dicha imagen en diferentes regiones o áreas¹⁵ de acuerdo a un criterio que está dado en función de lo que se busca en la imagen, con esto se trata de separar las regiones de interés para posteriormente someterlas a un análisis o simplemente presentarlas.

Se la considera como una de las etapas iniciales dentro del procesamiento de imágenes y también una de las más difíciles. De una correcta segmentación de la imagen dependerán en gran medida los resultados alcanzados al final del procesamiento.

Existen diferentes técnicas para segmentar imágenes y la selección de una o varias de éstas depende del tipo de aplicación a la que esté orientado el procesamiento. Sin embargo se pueden mencionar técnicas tales como: detección de bordes, detección de regiones, umbralización, agrupamiento o clustering, segmentación de texturas, morfología matemática, redes neuronales, etc. Estas y otras técnicas se encuentran ampliamente detalladas en la literatura [1, 22] por lo que sólo se mencionarán algunas de las más utilizadas.

1.1.11.1 Detección de bordes

Un borde se define como la frontera entre dos regiones. Generalmente diferentes regiones presentan diferentes niveles de gris o diferentes colores, por lo tanto una manera de detectar a los píxeles que forman parte del borde o contorno de un objeto es examinando los píxeles vecinos para determinar diferencias entre éstos (cambios bruscos entre los valores de los píxeles).

¹⁵ Una región o área en una imagen se define como un grupo de píxeles conectados que tienen propiedades similares.

Apoyándose en este principio se han creado operadores detectores de contornos basados en máscaras o filtros. En la Tabla 1.1 se mencionan algunos de estos detectores.

Debido a la presencia de artefactos en las imágenes como por ejemplo ruido, los píxeles de borde detectados no siempre caracterizan completamente a los contornos de los objetos, ya que se presentan con discontinuidades. Por esta razón es muy común utilizar luego de la detección de bordes una técnica de *enlazado y detección de límites* con el objetivo de unir píxeles de borde en contornos bien definidos y fácilmente representables. Entre las técnicas que cumplen este objetivo se encuentran: seguimiento de contornos, unión de bordes por búsqueda heurística, Transformada de Hough, entre otras.

1.1.11.2 Detección de regiones

Esta forma de segmentación se basa en similitudes entre píxeles vecinos para agruparlos dentro de regiones. Los píxeles son asignados a una región o componente según algún criterio que los diferencia del resto de la imagen. Todos los píxeles que corresponden a una región son etiquetados para indicar que pertenecen a dicha región.

1.1.11.3 Umbralización

La umbralización como técnica de segmentación es una manera rápida, sencilla y conveniente de separar las regiones de interés de las demás regiones en una imagen. Para esto se define un rango de valores (*intervalo umbral*) y se compara el valor de cada píxel en la imagen con este rango, todos los píxeles que estén dentro del rango son etiquetados como parte de la región de interés.

La forma más sencilla de etiquetar píxeles en un proceso de umbralización es cambiar su valor. Cuando están dentro del rango se

cambia su valor a uno y cuando están fuera del rango se cambia su valor a cero o viceversa. Al final se tiene una imagen binaria en la que se diferencian claramente las regiones de interés.

La mayor dificultad de esta técnica es precisamente el definir el rango de valores correcto que permita aislar dichas regiones.

Otra forma de umbralización es la *adaptativa*, en la cual se examinan regiones pequeñas de la imagen y para cada una de ellas se establece un intervalo umbral.

1.1.12 COMPRESIÓN DE IMÁGENES

Las técnicas desarrolladas para la compresión se fundamentan en el hecho de que las imágenes poseen información redundante, por lo tanto su objetivo es disminuir el número de bits que requieren las imágenes con la menor pérdida de calidad que sea posible. Estas técnicas se dividen en dos grandes grupos: compresión sin pérdidas y compresión con pérdidas.

1.1.12.1 Compresión sin pérdidas

También conocida como compresión reversible. Estas técnicas permiten la reconstrucción completa de la imagen comprimida a su forma original, sin pérdida de información. El nivel de compresión que se puede obtener aplicando estas técnicas es muy bajo y depende de la redundancia de los datos de la imagen.

Algunos ejemplos de técnicas de compresión sin pérdidas son: codificación de Huffman o codificación de longitud variable, codificación predictiva sin pérdidas, interpolación jerárquica, pirámide de diferencias, codificación de planos de bits, auto regresión multiplicativa, codificación de línea y codificación de área.

1.1.12.2 Compresión con pérdidas

También conocida como compresión irreversible. En este tipo de técnicas la imagen que se reconstruye varía de la imagen original ya que existe una pérdida de información.

Permiten obtener un alto nivel de compresión, por lo que son las más utilizadas para el tratamiento digital de las imágenes. Sin embargo su éxito depende en gran medida de lograr que la distorsión causada por la pérdida de cierta información no pueda ser distinguida por el ojo humano.

Dentro de la compresión con pérdidas existen diferentes técnicas tales como: compresión mediante transformadas (PCA, Transformada Discreta Coseno, Wavelets, etc.), cuantificación umbral, codificación predictiva con pérdidas, LOT (*Lapped Orthogonal Transform*), codificación en sub-bandas, cuantificación vectorial, quadrees, compresión mediante fractales, compresión mediante redes neuronales, entre otros.

1.1.12.3 Estándares de Compresión

Los estándares permiten definir ciertas especificaciones técnicas o criterios que serán aplicados con el objetivo de compartir e intercambiar información entre diferentes aplicaciones sin que existan problemas de compatibilidad. Se han dividido de acuerdo al tipo o naturaleza de la imagen, existen estándares para imágenes estáticas y estándares para imágenes dinámicas (video).

A continuación se mencionan los diferentes tipos de estándares que existen tanto para imágenes estáticas como para video, citando dentro de cada clase algunos ejemplos representativos.

Si se requiere una descripción más detallada de algunos de éstos estándares, se la puede encontrar en el Anexo 9.

1.1.12.3.1 Estándares de Compresión de Imágenes Estáticas

Existen diferentes estándares de compresión para las imágenes estáticas. Hay estándares para las imágenes bitmap y estándares para las imágenes vectoriales, también existen estándares para los metaficheros (híbridos):

- Formatos bitmap: TIFF, JPEG, GIF, PNG, PCX, BMP, RAW, PICT, PSD, TGA, entre otros.
- Formatos vectoriales: DXF, CDR, entre otros.
- Formatos híbridos: CGM, WMF, WPG.

1.1.12.3.2 Estándares de Compresión de video

En una imagen dinámica el contenido de ésta varía de acuerdo al movimiento, por lo que es necesario aprovechar la redundancia espacial¹⁶ y redundancia temporal¹⁷ para obtener niveles significativos de compresión.

Por ejemplo, para el caso de cuadros contiguos en los que no existe movimiento, la secuencia de éstos puede codificarse como una sola imagen e incluir solamente algunos bits para indicar que esa imagen se debe repetir. Si por el contrario, existe movimiento, se codifica la información que varía entre un cuadro y otro.

En muchos casos la información de un mismo cuadro puede ser reducida para así evitar redundancia.

Algunos de los estándares de compresión de video que se pueden mencionar son: Norma H.261, MPEG, Norma H.263 y Norma H.264.

¹⁶ Información que puede ser eliminada o reducida en el propio cuadro.

¹⁷ Similitud entre cuadros contiguos.

1.2 METODOLOGÍAS PARA EL RECONOCIMIENTO DE PATRONES

1.2.1 RECONOCIMIENTO DE PATRONES [10, 12, 78, 86]

Un patrón es un conjunto de descriptores o características que definen a un objeto determinado. Los patrones que describen objetos de una misma clase presentan características similares en tanto que aquellos que describen objetos de diferentes clases poseen diferentes características.

Por lo tanto el reconocimiento de patrones no es más que la clasificación de éstos ya sea en base a un conocimiento a priori (clasificación supervisada)¹⁸ o mediante su agrupamiento con otros patrones (clasificación no supervisada)¹⁹.

Este proceso es de gran utilidad en aplicaciones como recuperación de información, minado de datos, biometría, bioinformática, análisis de imágenes, entre otras.

De forma general se puede decir que existen tres metodologías básicas utilizadas para el reconocimiento de patrones:

- De correspondencia
- Estadística
- Sintáctica o estructural

En la actualidad las redes neuronales artificiales también son consideradas como otra forma de reconocer patrones, especialmente por su habilidad de aprender relaciones complejas, de adaptarse y de ser tolerantes a fallos. En esta sección no se considerará a las redes neuronales ya que serán analizadas en secciones posteriores.

¹⁸ Para la clasificación se utiliza un conjunto de entrenamiento formado por patrones de los que se conoce previamente a qué clase pertenece cada uno de ellos. Estas clases constituyen el universo de objetos.

¹⁹ Se refiere a encontrar grupos de patrones similares, sin tener información previa sobre las clases que existen. Para esto se emplean métodos de agrupamiento o clustering.

1.2.1.1 De Correspondencia

Cada clase está caracterizada mediante un patrón prototipo o un conjunto de prototipos. La clasificación se lleva a cabo al comparar un patrón desconocido con los prototipos de las clases existentes.

Una de las mayores ventajas de este enfoque es su simplicidad, sin embargo tiende a ser poco confiable cuando las clases forman nubes dispersas y están muy cercanas a otras clases.

Entre las técnicas que se basan en esta metodología se pueden mencionar:

- *Clasificador de distancia mínima:* Cada clase está representada por un prototipo. Por lo tanto para determinar a qué clase pertenece un nuevo patrón, se debe establecer cuál es el prototipo más próximo a éste, una manera de establecer el grado de proximidad es a través de medidas de distancia, esto significa que la menor distancia está relacionada con la mejor correspondencia.
- *Adaptación por correlación:* También se basa en la comparación de un patrón a clasificar con los patrones prototipo de cada clase existente. En este caso la comparación se realiza mediante la correlación, valores altos de correlación indican una buena correspondencia entre el nuevo patrón y un prototipo. Este método no es práctico cuando existen rotaciones arbitrarias o cuando un patrón es demasiado grande.

Como se menciona en el Anexo 6, la correlación también puede ser calculada mediante la FFT para reducir complejidad.

1.2.1.2 Estadística

Esta metodología se fundamenta en la estadística y en la teoría de probabilidad. Busca clasificar los patrones utilizando un conjunto de características extraídas y apoyándose en un modelo estadístico que

describa el proceso asociado a la generación de dichos patrones (variabilidad de los patrones de una clase, medidas de similitud entre patrones, etc.)

Los patrones se representan como vectores numéricos de dimensión n , de tal forma que un patrón es un punto en un espacio n -dimensional de características, por lo tanto patrones de objetos similares serán puntos cercanos en dicho espacio.

El reconocimiento de patrones por métodos estadísticos presenta dos etapas:

- *Etapas de entrenamiento*: Se entrena al clasificador con un conjunto de datos de entrenamiento (en una etapa previa se diseña el extractor de características para representar a los patrones de entrada).
- *Etapas de reconocimiento*: El clasificador previamente entrenado recibe como entrada un vector de características de un patrón desconocido y lo asigna a una de las clases existentes.

Entre las técnicas basadas en esta metodología se pueden mencionar: reglas de decisión, criterios de error mínimo, análisis de clusters, etc.

1.2.1.3 Sintáctica o Estructural

El reconocimiento de patrones a través de esta metodología ha sido ampliamente utilizado en las últimas dos décadas en especial en problemas prácticos como reconocimiento de caracteres, análisis de texturas, reconocimiento de huellas, reconocimiento de voz, entre otros.

De acuerdo a esta metodología un patrón puede ser descompuesto en subpatrones más simples, cada uno de los cuales puede a su vez ser descompuesto en nuevos subpatrones y así sucesivamente. A los subpatrones más simples se los denomina *primitivas*, de tal manera que un patrón puede ser descrito como una combinación de primitivas

(cadena, matriz, árbol, grafo, etc.) basándose en ciertas reglas sintácticas. Así se busca construir una gramática que describa la estructura del universo de objetos²⁰.

El reconocimiento de patrones a través de esta metodología está compuesto de dos etapas:

- *Etapas de entrenamiento:* Se refiere a la selección y relación de primitivas, inferencia gramatical y la construcción de autómatas a partir de los patrones de entrenamiento.
- *Etapas de reconocimiento:* Involucra el preprocesamiento, segmentación, reconocimiento de primitivas, construcción de la representación del patrón y el análisis sintáctico para el nuevo patrón (patrón que se desea clasificar).

Entre las técnicas que utilizan este tipo de metodología se pueden mencionar: clasificadores sintácticos, emparejamiento de grafos, etc.

1.2.2 RECONOCIMIENTO DE ROSTROS [47, 76, 91, 92, 90]

El reconocimiento de rostros es un área que forma parte del reconocimiento de patrones. En los últimos años ha cobrado un gran interés especialmente por la amplia gama de aplicaciones que tiene en distintos campos tales como seguridad, vigilancia, tarjetas inteligentes, etc.

A continuación se describirán las metodologías actualmente utilizadas para el reconocimiento de rostros citando algunas técnicas representativas de cada una de éstas. Sin embargo es importante mencionar que si bien en los últimos años se ha logrado un gran avance en el área de reconocimiento de rostros desarrollando técnicas más robustas que buscan

²⁰ De acuerdo a Anil [10] en el reconocimiento sintáctico de patrones se deduce una analogía formal entre la estructura de los patrones y la sintaxis de un lenguaje. Los patrones se pueden ver como frases que pertenecen a un lenguaje, las primitivas como el alfabeto de ese lenguaje y las oraciones se generan de acuerdo a una gramática. La gramática asociada a cada clase de patrón se debe inferir del conjunto de datos de entrenamiento.

solucionar problemas como cambios en iluminación, rotaciones, oclusiones, entre otros; todavía no se puede hablar de métodos para reconocimiento de rostros que resulten altamente confiables y tolerantes a diferentes condiciones o circunstancias de manera que asemejen el proceso de reconocimiento que lleva a cabo un ser humano. Por lo tanto algunas metodologías presentan mejores resultados bajo unas condiciones y otras bajo condiciones diferentes, lo que permite concluir que la selección de una u otra metodología y de las técnicas empleadas, está basada en los requerimientos específicos de la aplicación para la que se realiza la tarea de reconocimiento.

Las metodologías para el reconocimiento de rostros han sido divididas de acuerdo al tipo de imágenes utilizadas, así existen metodologías para reconocimiento de rostros a partir de imágenes estáticas y metodologías para reconocimiento de rostros mediante video.

1.2.2.1 Metodologías para Reconocimiento de Rostros a partir de imágenes estáticas

1.2.2.1.1 Métodos basados en características geométricas

Estos métodos están orientados a la construcción de modelos del rostro humano a partir de características geométricas (características invariantes) que permitan establecer diferencias faciales entre un rostro y otro, (véase Figura 1.13).

Los modelos pueden ir desde un conjunto de medidas antropométricas faciales tales como distancia ínter ocular, separación entre ojos y nariz, separación entre boca y nariz, tamaño de la boca, tamaño de los ojos, alto o ancho del área del rostro, entre otras medidas; hasta modelos más complejos como diagramas elásticos compuestos por nodos en puntos definidos en el rostro (ojos, nariz, contornos, etc.) [13, 18, 83].

Como se puede notar uno de los pasos esenciales en estos métodos es la detección de características locales (ojos, boca, nariz, etc.), debido a que son la base para construir el modelo del rostro. Para llevar a cabo la detección de estas características, cada una es considerada como un nuevo patrón (con sus propias particularidades) y se puede emplear cualquiera de las metodologías para reconocimiento de patrones descritas en la sección anterior. Una de las metodologías más comúnmente utilizadas con este fin es la de correspondencia.

Una vez construido el modelo del rostro, métodos estadísticos o redes neuronales, generalmente son empleados para determinar la identidad de la persona en base al modelo creado y sus componentes.

Los métodos basados en características geométricas suelen ser menos afectados por cambios de iluminación en las imágenes lo que constituye una ventaja, sin embargo presentan menor tolerancia a cambios de expresión facial y la construcción de los modelos puede ser difícil y costosa a nivel computacional.

Entre las técnicas más representativas están: métodos puramente geométricos, arquitecturas de enlace dinámico como por ejemplo EBG, entre otras.

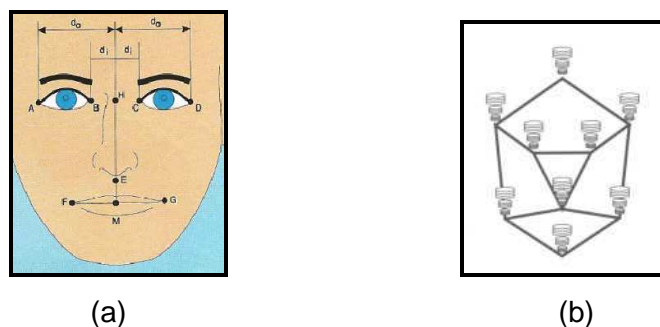


Figura 1.13. Modelos del rostro humano (a) en base a medidas antropométricas (b) diagrama elástico con nodos en puntos específicos del rostro.

Fuente: XU, X. 2005.

1.2.2.1.2 *Métodos holísticos o basados en la imagen*²¹

Son métodos conceptualmente relacionados con el uso de plantillas, tratan de identificar los rostros usando representaciones globales. Es decir, consideran a la imagen del rostro como un todo por lo tanto no tratan de analizar su contenido sino que extraen características relevantes de la región completa.

Una vez que han sido extraídas las características del área del rostro, al igual que en los métodos basados en características geométricas se puede utilizar métodos estadísticos o redes neuronales para llevar a cabo el reconocimiento en sí.

Este tipo de métodos son utilizados habitualmente en imágenes de baja resolución.

Entre las técnicas más representativas están: Eigenfaces (PCA), métodos bayesianos (PCA), Fisherfaces (LDA), Support vector machine, ICA, decisión probabilística basada en redes neuronales, técnicas basadas en transformadas (Fourier, Wavelets, Transformada Discreta Coseno, etc.), entre otras.

1.2.2.1.3 *Métodos híbridos*

Combinan aspectos de los métodos anteriores. Si se considera que los métodos holísticos y los basados en características geométricas tienen ventajas y desventajas con respecto al otro, el combinarlos puede lograr mejores resultados [25, 34, 45].

Algunas de las técnicas o métodos híbridos son: Eigenfaces modulares (PCA a nivel del rostro y de áreas específicas), análisis híbrido de características locales (LFA híbrido), entre otras.

²¹ Véase ejemplos en [15, 19, 31, 33, 44, 54, 61, 67, 89].

1.2.2.2 Metodologías para Reconocimiento de Rostros mediante video

El reconocimiento de rostros en imágenes dinámicas tiene algunas características diferentes al realizado a partir de imágenes estáticas.

Existen nuevos elementos que pueden ser utilizados: uso de movimiento para detección, modelación de rostros en tres dimensiones, análisis de información adicional (por ejemplo audio), entre otros.

Puesto que el presente trabajo se enfoca en el reconocimiento de rostros a partir de imágenes estáticas, sólo se mencionarán algunas de las metodologías utilizadas para reconocimiento mediante video:

- Métodos para imágenes estáticas
- Métodos multi-modales
- Métodos espacio-temporales

Estos y otros métodos se encuentran descritos en la literatura.

1.3 FASES QUE INVOLUCRA EL RECONOCIMIENTO DE PATRONES

El reconocimiento de patrones en imágenes digitalizadas se lleva a cabo en varias fases o etapas, (véase Figura 1.14), las mismas que se pueden establecer de forma general independientemente del tipo de patrón que se busque y son:

- Adquisición de imágenes
- Preprocesamiento
- Extracción de características
- Reconocimiento e interpretación

No todas las fases antes mencionadas deben necesariamente formar parte del proceso de reconocimiento de un patrón, su inclusión depende del tipo de patrón y del objetivo de la aplicación para la que se realice el procesamiento; de igual manera las técnicas o métodos empleados dentro de cada una de estas fases están sujetas a los mismos factores.

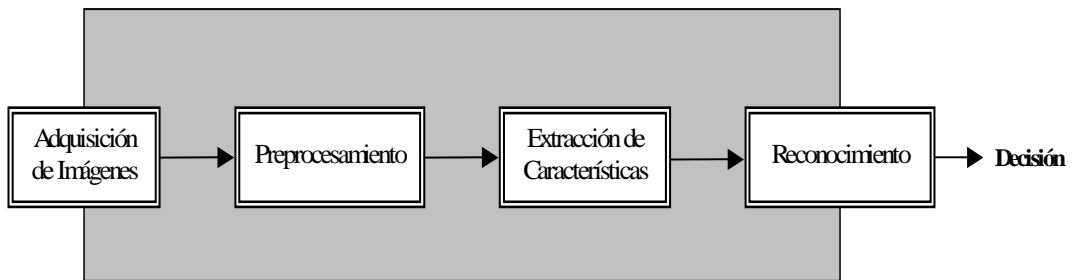


Figura 1.14. Fases que involucra el reconocimiento de patrones

Finalmente se debe mencionar al *conocimiento sobre el dominio del problema* como un elemento adicional que está presente en el reconocimiento de cualquier patrón y que de alguna forma orienta la manera de ejecutar cada fase y la comunicación entre éstas.

Este conocimiento es tan variado que puede ir desde un rango umbral que permita segmentar objetos considerando su color o el tamaño aproximado que debe tener un objeto, hasta una base de imágenes que representen a un conjunto de clases de objetos, así como también el conocimiento previo de cómo debe ser el resultado.

1.3.1 ADQUISICIÓN DE IMÁGENES

En esta fase se adquiere la imagen o conjunto de imágenes que van a ser utilizadas para el reconocimiento. Existen algunos factores que se deben considerar dentro de esta fase:

- *Tipo de dispositivo de captura:* El tipo de dispositivo estará en función de la naturaleza del patrón y de los requerimientos de la aplicación. Así por

ejemplo si se busca realizar una verificación estática de firmas se puede adquirir la imagen de la firma digitalizada mediante un escáner, por el contrario si se requiere detectar movimiento en una habitación se debe recurrir al uso de una cámara. De igual manera si la imagen adquirida no ha sido digitalizada por el dispositivo, se requerirá utilizar adicionalmente un dispositivo para conversión analógico-digital.

o *Condiciones de la adquisición:* Este factor está muy ligado a las características del dispositivo seleccionado y a los requerimientos de la aplicación y puede influir en la selección de una u otra técnica de procesamiento en las fases posteriores. Se deben definir factores como por ejemplo: profundidad del píxel, tipo de imagen, formato, resolución de la imagen, uso de fondos complejos en la imagen o imágenes que contienen sólo el objeto a reconocer, factores de iluminación, número de imágenes requeridas, entre otros.

1.3.2 PREPROCESAMIENTO

El utilizar las imágenes tal como han sido obtenidas resulta en la mayoría de los casos poco conveniente, esto se debe a que durante la adquisición pueden ocurrir alteraciones que afecten a la imagen (ruido, errores de digitalización, exceso o poca iluminación en la imagen, etc.). Otra razón puede ser el requerir eliminar partes de la imagen no deseadas.

De ahí que la fase de preprocesamiento busca adaptar la imagen a ciertas condiciones para facilitar su tratamiento posterior. Como parte del preprocesamiento se pueden aplicar técnicas para mejorar, restaurar, segmentar, rotar, ampliar, reducir, extraer áreas de la imagen, entre otras.

Además de las técnicas ya mencionadas, en la literatura existen técnicas novedosas, que han sido creadas para resolver algún tipo de problema bajo condiciones específicas.

1.3.3 EXTRACCIÓN DE CARACTERÍSTICAS

La fase de extracción de características tiene como objetivo reducir la dimensionalidad de los datos de entrada (imágenes) mediante la extracción de información discriminativa.

Si se considera que el número de características en una imagen puede ser tan grande como el número de píxeles que contiene²², es lógico pensar que en este contexto muchas de las características no representan información relevante o están altamente correlacionadas. Por lo tanto la extracción de ciertas características permite representar a las imágenes iniciales de una manera más óptima sin pérdida de información relevante.

Es en esta nueva representación en donde se fundamenta la capacidad de un clasificador de distinguir entre diferentes patrones y clases, consecuentemente en esta fase se deberá considerar dos aspectos: determinar cómo se va a extraer las características y una vez extraídas seleccionar qué características permitirán discriminar de mejor manera a un patrón de otro, considerando también que el número de características seleccionadas influirá en la cantidad de operaciones a efectuar en la fase de reconocimiento (considerar confiabilidad y tiempo de procesamiento).

Adicionalmente la extracción de características implica el almacenamiento de dos tipos de datos: las imágenes preprocesadas y las nuevas representaciones (vectores de características).

Para la extracción de características existen técnicas como por ejemplo: PCA, ICA, transformada de Fourier, transformada Discreta Coseno, Wavelets, etc. En cuanto a la selección de características existen técnicas heurísticas, selección mediante algoritmos genéticos, búsqueda exhaustiva, búsquedas secuenciales, entre otras.

²² Por ejemplo en una imagen de 100 x 100 píxeles el número de características sería 10 000

1.3.4 RECONOCIMIENTO E INTERPRETACIÓN

En esta etapa se realiza el reconocimiento en sí, para esto se utiliza cualquiera de las metodologías para reconocimiento de patrones antes descritas. El resultado de la fase de reconocimiento está sujeto a una interpretación, es decir se le da un significado; por ejemplo si se trata de reconocer una huella dactilar, el proceso de reconocimiento etiquetará al patrón asociado a ésta dentro de una clase y la interpretación le dará un significado, en este caso la identidad de la persona a la pertenece la huella. Finalmente la interpretación produce una decisión que está en relación con el objetivo de la aplicación y el tipo de patrón (conceder o negar accesos, diagnosticar enfermedades, pronosticar cambios climatológicos, etc.).

1.3.5 FASES APLICADAS AL RECONOCIMIENTO DE ROSTROS

La mayoría de métodos para reconocimiento de rostros aplican las fases antes mencionadas utilizando el esquema expuesto en la Figura 1.15.

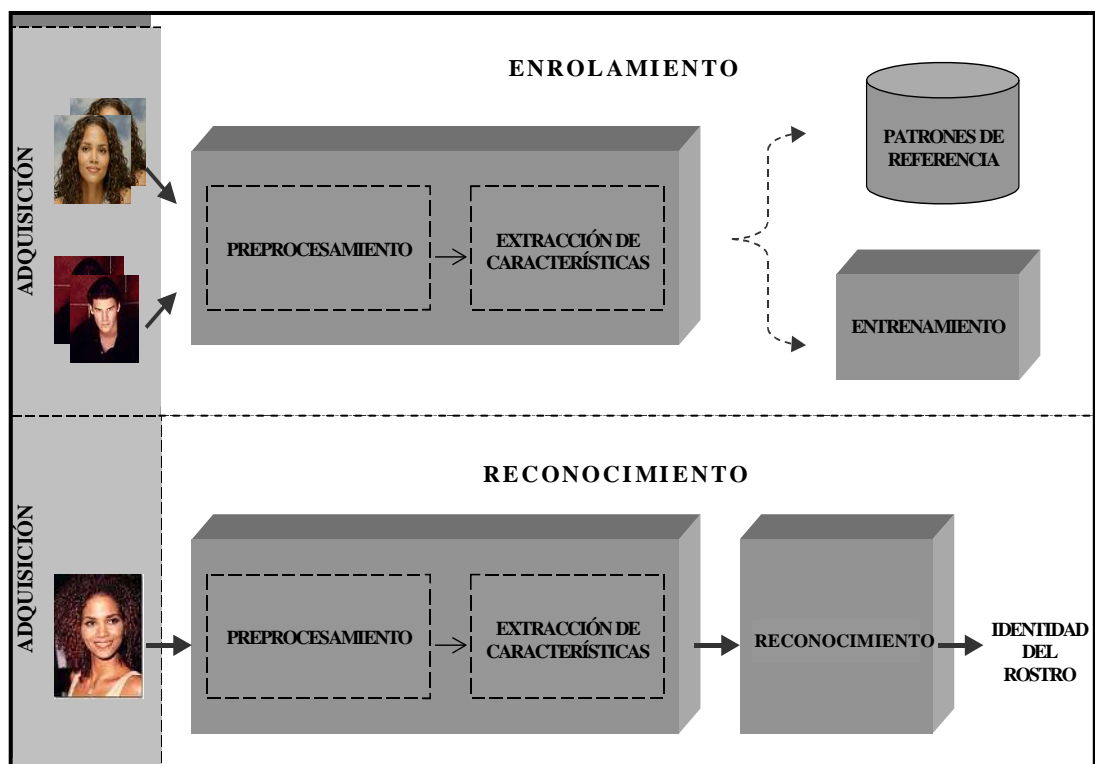


Figura 1.15. Esquema general para el reconocimiento de rostros

1.3.5.1 Enrolamiento

Durante el enrolamiento se generan los vectores de características o patrones de referencia que permitirán identificar a los diferentes usuarios²³. Dependiendo del método seleccionado el proceso final del enrolamiento puede consistir en almacenar los patrones de referencia para compararlos durante la etapa de reconocimiento con patrones desconocidos o bien puede ser el entrenamiento de un clasificador con dichos patrones.

A continuación se describen sólo aquellas fases que requieren cierta puntualización para el caso específico de reconocimiento de rostros:

a. *Adquisición de Imágenes*: Durante el enrolamiento se recopila un conjunto de imágenes que representan a los usuarios (conjunto de entrenamiento). Por cada usuario puede existir una o varias imágenes, si son varias imágenes éstas deben ser lo más diferente posible entre sí para lograr una mejor generalización.

b. *Preprocesamiento*: El realizar o no ciertas tareas en esta fase depende del grado de automatización que se desee dar al proceso de reconocimiento de rostros, existen algoritmos para reconocimiento *completamente automáticos* y otros que son *parcialmente automáticos*.

En los dos tipos de algoritmos se puede requerir operaciones de preprocesamiento como las referidas en la Sección 1.3.2 con el objetivo de *normalizar* las imágenes en cuanto a iluminación, orientación del rostro, tamaño del rostro, eliminación de ruido, etc.

La diferencia radica en que los algoritmos *parcialmente automáticos* reciben como entrada una imagen y algún otro tipo de información que indica la ubicación de cada rostro en dicha imagen (información como las coordenadas de los centros de los ojos, coordenadas del vértice de la

²³ Personas que formarán parte del universo de rostros conocidos (clases que lo constituyen).

nariz, etc.). Esto significa que se conoce a priori la ubicación de los rostros, por lo tanto el preprocesamiento en este caso consiste en extraer de la imagen inicial las áreas correspondientes a los rostros (así se elimina información innecesaria como fondos u otros objetos) y si se requiere se las normaliza.

En cuanto a los *algoritmos completamente automáticos*, sólo reciben una imagen y deben realizar un proceso conocido como detección de rostros [88, 58, 28] para determinar la presencia de rostros en la imagen y su ubicación. Luego que han detectado los rostros los extraen de la imagen y continúan con el proceso de reconocimiento.

En sí la detección de rostros es un tipo específico de reconocimiento de patrones, en el que existen sólo dos clases o grupos dentro de los que se puede clasificar a un patrón, el grupo que representa rostros humanos y el que representa al resto de objetos. Las técnicas para detección de rostros se dividen en: basadas en características (utilizan información de color, movimiento, contornos, etc.), basadas en la imagen (transformadas de la imagen, métodos estadísticos, redes neuronales, etc.) y las técnicas híbridas.

La localización de rostros es un caso particular de la detección, se usan las mismas técnicas pero se tiene como información adicional el número de rostros existentes en la imagen (usualmente uno), en otras palabras se conoce a priori el número de rostros a detectar [70, 71, 69].

En síntesis se puede decir que en los algoritmos completamente automáticos la detección de rostros constituye una tarea adicional dentro de la fase de preprocesamiento, de la cual dependerá en gran medida los resultados finales.

c. *Entrenamiento*: A esta fase también se la denomina de aprendizaje, ya que mediante alguna forma de entrenamiento el clasificador utilizado

“aprende” los patrones que describen a los rostros que integran el conjunto de entrenamiento. Esto permitirá que durante la etapa de reconocimiento el clasificador pueda discriminar lo mejor posible a un patrón de otro.

La forma de entrenamiento dependerá del clasificador utilizado, algunos ejemplos de estos clasificadores son: redes neuronales artificiales, árboles de decisión, clasificador de bayes, algoritmos genéticos, algoritmo de vecinos cercanos, Support vector machine, entre otros.

1.3.5.2 Reconocimiento

Se realiza el reconocimiento de un rostro utilizando como herramienta al clasificador previamente entrenado o a los patrones de referencia almacenados.

A continuación se describen sólo aquellas fases que requieren cierta puntualización ya que difieren de alguna manera con las detalladas en el enrolamiento o no forman parte de éste:

a. *Adquisición de Imágenes:* Se obtiene una imagen del rostro que se desea reconocer. En ocasiones en la literatura se hace uso del término conjunto de prueba, que se refiere a un conjunto de imágenes que se utiliza para probar el grado de confiabilidad del algoritmo aplicado para el proceso de reconocimiento, por lo tanto fuera del ámbito de desarrollo este conjunto no es requerido.

b. *Reconocimiento:* Se deben distinguir dos formas de aplicar esta fase:

- Identificación. También conocida como reconocimiento. En este caso, es el resultado de la comparación entre patrones o el clasificador quien asigna mediante la interpretación, la posible identidad al rostro presentado. La decisión de asignar una u otra identidad generalmente se basa en la aplicación de valores umbrales.

- Verificación. También conocida como autenticación. Esta forma requiere que además de la imagen de un rostro se proporcione una identificación asociada a éste, de tal manera que al finalizar el proceso de reconocimiento se verifica que la identidad suministrada realmente corresponda con la del rostro analizado. Es decir, es un proceso de validación entre la información proporcionada y el resultado obtenido mediante el proceso de reconocimiento.

1.4 REDES NEURONALES

1.4.1 DEFINICIÓN

Las redes neuronales artificiales o RNA son sistemas para procesar información inspirados en la estructura y funcionamiento del cerebro humano. Son capaces de almacenar conocimiento que han adquirido a través de un *proceso de adaptación* y utilizarlo con capacidad de generalización, razón por la cual se dice que simulan características propias del ser humano como memorizar y asociar hechos.

Las unidades de procesamiento que las componen están altamente interconectadas y por analogía con el cerebro humano se las denomina *neuronas*, cada neurona recibe varias señales de entrada y envía una única señal de salida.

1.4.2 ELEMENTOS

Los elementos de una RNA en términos generales se pueden definir como:

- a. *Unidad de proceso o neurona*: Unidad básica del sistema. Existen tres clases de neuronas: entradas (reciben señales desde el entorno), salidas (envían la señal fuera de la red) y ocultas (sus entradas y salidas se encuentran al interior de la red).

b. *Estado de Activación*: Son los estados de la red en un momento determinado, cada neurona tiene su propio estado de activación. Existen dos estados: reposo y excitado.

c. *Función de salida o de transferencia*: Cada neurona tiene una función de salida asociada cuya tarea es transformar el estado actual de activación en una señal de salida.

Existen cuatro funciones de transferencia típicas, las cuales definen distintos tipos de neuronas: función escalón, lineal y mixta, sigmoideal y gaussiana.

d. *Conexiones*: Se denominan *sinapsis*, cada una de éstas tiene asociado un peso que es el responsable de que la red adquiera conocimiento. Las señales que llegan a las sinapsis son las entradas a la neurona, le proveen información sobre el estado de activación de todas las neuronas con las que se conecta. Cuando las sinapsis tienen peso positivo la señal de entrada puede excitar a la neurona, si las sinapsis tienen asociado un peso negativo, la señal inhibe a la neurona.

e. *Función o regla de activación*: Es una función que combina las entradas con el estado actual de la neurona para producir un nuevo estado de activación.

Las funciones que existen con este objetivo son: función escalón, lineal y mixta, sigmoideal y gaussiana.

f. *Regla de aprendizaje*: Ya que el conocimiento en una red se encuentra en los pesos asociados a las sinapsis, un proceso de aprendizaje implica el ajuste o modificación iterativa de estos pesos. Por lo tanto, la regla de aprendizaje define cómo se realizan éstos ajustes.

1.4.3 ESTRUCTURA DE UNA RED NEURONAL ARTIFICIAL

La estructura de una red neuronal está dada por la forma cómo se conectan las neuronas:

a. *Propagación hacia delante*: Ninguna salida de las neuronas es entrada de neuronas del mismo nivel o de niveles anteriores, esto significa que no existen uniones de retroalimentación, (véase Figura 1.16 (a))

b. *Propagación hacia atrás*: Las salidas de las neuronas pueden ser entradas de neuronas del mismo nivel, de niveles anteriores o incluso de la misma neurona. Existe retroalimentación, (véase Figura 1.16 (b))

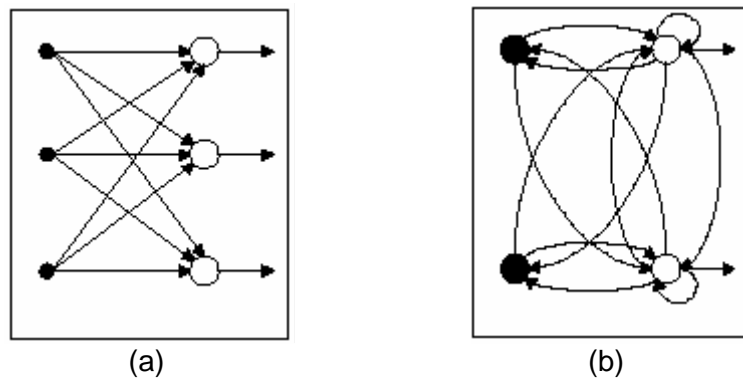


Figura 1.16. Ejemplos de redes con diferentes estructuras: (a) Perceptrón simple
(b) Red ART.

Fuente: Romero, A.; España.

1.4.4 ARQUITECTURA DE UNA RED NEURONAL ARTIFICIAL

En general los parámetros más importantes para definir la arquitectura de una red neuronal son: el número de capas, número de neuronas por capa, grado y tipo de conexiones entre neuronas. De acuerdo a estas características existen dos tipos de redes:

- Monocapa: Está compuesta por una sola capa de neuronas.
- Multicapa: La red esta compuesta por varias capas.

1.4.5 MECANISMOS DE APRENDIZAJE

Son los mecanismos por los cuales las redes modifican sus pesos hasta encontrar un valor adecuado, basándose en la información recibida como entrada. Cuando los valores de los pesos permanecen estables se dice que

una red ha aprendido, la forma de aprendizaje es algo característico del tipo de red.

Los mecanismos de aprendizaje se pueden clasificar de acuerdo a dos criterios:

- Las reglas de aprendizaje utilizadas durante el funcionamiento habitual de la red (no existe una fase de entrenamiento por separado a la de operación, la red aprende mientras realiza una tarea) o las que implican la desconexión de la red (la red es primero entrenada y luego utilizada para la tarea para la que se le entrenó, es decir existen dos fases).
- Las reglas de aprendizaje supervisado y las de aprendizaje no supervisado.

1.4.5.1 Aprendizaje Supervisado

Requiere de un agente externo que establezca la salida que debería generar la red para una entrada determinada. Los pesos de la red serán modificados hasta que se aproximen a la salida deseada (minimizar el error). En este grupo se encuentran el algoritmo de aprendizaje por corrección de errores, algoritmo de aprendizaje por refuerzo, algoritmo de aprendizaje estocástico, entre otros.

El *refuerzo* es una variante de este tipo de aprendizaje, donde no se le proporcionan a la red las salidas deseadas sino una crítica de corrección de salida.

1.4.5.2 Aprendizaje No Supervisado

La red no recibe ninguna información sobre salidas esperadas para ajustar los pesos de sus conexiones, por lo tanto la salida generada proviene de un proceso de auto-organización de la información hasta que

la red llegue a un estado considerado estable. Dentro de este tipo de aprendizaje está el algoritmo de aprendizaje hebbiano y el algoritmo de aprendizaje competitivo.

1.4.6 TIPOS DE REDES

Como se ha visto las redes pueden clasificarse de acuerdo a diferentes aspectos. Sin embargo de manera general se habla de redes con propagación hacia delante (sin realimentación) y con propagación hacia atrás (con realimentación). Considerando este criterio la Tabla 1.2 muestra algunos de los tipos de redes más conocidos.

Propagación hacia delante	Propagación hacia atrás
Perceptrón simple	Redes Kohonen
Perceptrón Multicapa	Redes de Hopfield
Función de Base Radial	Redes ART

Tabla 1.2. Una taxonomía de redes neuronales artificiales

1.4.7 VENTAJAS DE LAS REDES NEURONALES

Las redes neuronales tienen ciertas características que pueden ser consideradas como ventajas al momento de seleccionar este tipo de sistemas:

- *Aprendizaje adaptativo:* Las redes neuronales aprenden a llevar a cabo ciertas tareas mediante un entrenamiento con ejemplos, siendo capaces de adaptarse a nuevas condiciones.
- *Auto-organización:* Organizan la información que reciben durante el aprendizaje y crean su propia representación, esto permite la generalización a partir de los datos de entrada.

- *Tolerancia a fallos*: Son robustas frente a perturbaciones externas, si se produce un fallo en un número pequeño de neuronas, aunque la red puede ser afectada continúa realizando su función con cierta degradación. Esta tolerancia se debe principalmente a la memoria distribuida, deslocalizada y a la redundancia de información que tiene la red.

1.4.8 APLICACIONES

Las redes neuronales son utilizadas para resolver problemas que son muy complejos para los sistemas convencionales ya sea porque no tienen una solución algorítmica o ésta es muy difícil de implementar. También se las utiliza para aquellos problemas que el ser humano no tiene dificultad en resolverlos pero que para las computadoras sí presentan un grado de dificultad, en general son problemas asociados a destrezas del ser humano.

Entre las principales aplicaciones que tienen las redes neuronales se pueden citar: reconocimiento de patrones, predicción, recuperación de información distorsionada o incompleta, optimización de tareas, diagnóstico automático, procesamiento de señales, compresión de datos, etc.

1.4.9 DISEÑO DE SOLUCIONES CON REDES NEURONALES ARTIFICIALES

Gracias a las ventajas que las redes neuronales artificiales ofrecen, su utilización para la resolución de problemas en diversas áreas se ha extendido. Por esta razón es importante definir a breves rasgos cómo se debe llevar a cabo el proceso para incorporar redes neuronales en la resolución de un problema.

El diseño de soluciones con redes neuronales artificiales abarca los siguientes pasos: ²⁴

²⁴ Basado en las notas del curso *Sistemas Inteligentes*, Dr. Hugo Banda (EPN - 2002).

- Seleccionar una arquitectura de red adecuada para el problema que se desea solucionar.
- Entrenar la red utilizando un conjunto de entrenamiento. Esto implica adicionalmente seleccionar un algoritmo de aprendizaje apropiado y crear el conjunto de entrenamiento de manera que represente el problema a resolver.
- Probar la capacidad de la red para resolver el problema. Se utiliza un conjunto de prueba que tiene datos diferentes al de entrenamiento.
- Evaluar el funcionamiento de la red con nuevos datos, esto se denomina *generalización*.

Como se puede observar el diseño de soluciones con redes neuronales requiere la selección de una arquitectura apropiada, el desarrollo o selección de un buen algoritmo de aprendizaje y la creación de un conjunto de datos de entrenamiento correcto (que represente el problema).

Cada uno de estos factores influirá en cierta medida en la capacidad de la red para resolver el problema para el que fue entrenada, dichos factores requieren la intervención humana, no obstante el proceso de aprendizaje en sí, no es un aspecto en el que se pueda intervenir ya que es intrínseco a la red neuronal (caja negra).

El definir correctamente aquellos factores que requieren la intervención humana es una tarea difícil y en ocasiones requiere gran cantidad de tiempo. Por ejemplo la selección de una arquitectura de red es en muchos casos consecuencia de un proceso de prueba y error, ya que los tipos de redes con los que se cuenta actualmente no son universales, es decir funcionan bien para ciertas tareas y para otras no.

1.5 ARQUITECTURAS MÁS ADECUADAS PARA EL RECONOCIMIENTO DE ROSTROS

En la actualidad existen diversos algoritmos para reconocimiento de rostros que emplean redes neuronales, muchos de ellos con resultados promisorios [17, 43, 65]. Por esta razón se podría pensar que el comparar los resultados de los experimentos o investigaciones realizadas con dichos algoritmos, podría ser una base para seleccionar la arquitectura de red más adecuada para el reconocimiento de rostros.

Sin embargo si se analiza detenidamente algunos de los algoritmos propuestos, se puede observar que existe una gran disparidad en los datos empleados por dichos algoritmos.

Existen algoritmos que emplean redes neuronales para las fases de entrenamiento y reconocimiento, es decir sólo como clasificadores; en estos casos las redes reciben datos adquiridos, preprocesados o extraídos de diferentes maneras (de acuerdo a la metodología utilizada por cada algoritmo). Mientras que otros algoritmos también las utilizan para detectar el área del rostro, filtrar las imágenes y extraer las características más relevantes, es decir las redes también se encargan del preprocesamiento y extracción de características (elaboran sus propias representaciones).

Muchos algoritmos utilizan sólo una red neuronal, en tanto que otros utilizan *ensamble de redes*²⁵ o *mezclas de expertos*²⁶ con el objetivo de formar arquitecturas más complejas. Existen algoritmos que utilizan conjuntos de entrenamiento o de prueba reducidos (pocas imágenes), otros utilizan un número mayor de imágenes, algunos trabajan con imágenes adquiridas en una sola sesión, otros con imágenes adquiridas en diferentes momentos, algunos utilizan bases de rostros modelos, otros crean sus bases, etc.

²⁵ Varias redes neuronales entrenadas para una misma tarea.

²⁶ Se refiere a entrenar redes para diferentes subtareas y sólo un número pequeño o incluso sólo una red es usada para el proceso de reconocimiento, basándose en los resultados de las otras redes

Considerando lo antes expuesto, se puede concluir que no se cuenta con datos totalmente estandarizados que sirvan como base para definir qué arquitectura es la más adecuada para el reconocimiento de rostros.

Además como se mencionó en la sección anterior, los tipos de redes que actualmente existen pueden funcionar bien bajo ciertas condiciones y no bajo otras.

Así la selección de una arquitectura de red neuronal es una tarea mas bien de tipo experimental que puede auxiliarse de las investigaciones previas así como también del soporte teórico sobre RNA con el que se cuenta actualmente.

En cuanto a la definición de las arquitecturas más adecuadas para el reconocimiento de rostros es todavía un problema sujeto a investigación.

Por lo tanto, el siguiente análisis se centrará en los tipos de redes más frecuentemente utilizados para el reconocimiento de rostros, específicamente como clasificadores (para entrenamiento y reconocimiento), mencionando sus principales características, algunas de las cuales podrían justificar de cierta manera su uso en la resolución de este tipo de problemas.

La familia de redes más frecuentemente utilizada para el reconocimiento de patrones es la de propagación hacia delante, en especial el Perceptrón Multicapas y las redes Función Base Radial. Otro tipo muy difundido en este campo es la red Kohonen o Mapa Auto-Organizativo (SOM).

En el área de reconocimiento de rostros estos tipos de redes también han tenido gran aceptación, por lo tanto serán descritas a continuación. No obstante existen otros algoritmos que utilizan diferentes arquitecturas con el objetivo de probar su capacidad para resolver este problema.

1.5.1 PERCEPTRÓN MULTICAPAS

La red perceptrón multicapas (MLP, por sus siglas en inglés) es del tipo propagación hacia adelante, utiliza aprendizaje supervisado. Está compuesta por tres clases de capas: una capa de entrada, una de salida y una o varias capas ocultas, (véase Figura 1.17).

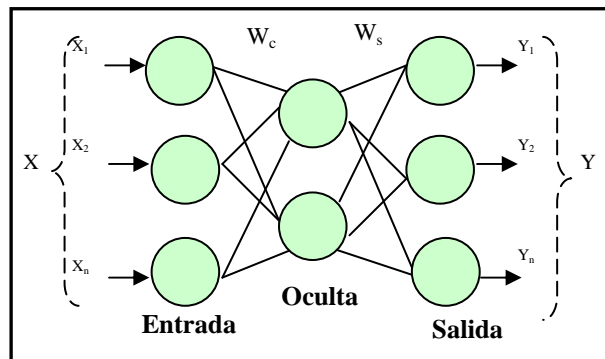


Figura 1.17. Perceptrón Multicapas (3 capas)

En general las redes MLP no utilizan más de tres capas, ya que esta arquitectura es suficiente para generar regiones de decisión arbitrarias ²⁷.

El algoritmo de aprendizaje comúnmente utilizado para este tipo de red es el algoritmo de retro-propagación, que tiene como objetivo propagar los errores producidos en la capa de salida hacia las capas anteriores para así modificar los valores de los pesos.

Entre las principales características que se pueden mencionar acerca de este tipo de red están:

- Es un aproximador universal, esto significa que puede ser usada en aplicaciones donde no se tiene mucho conocimiento sobre la relación entre las entradas y las salidas.
- La presencia de una o más capas escondidas facilita que pueda resolver tareas complejas, sin embargo esto puede provocar que el proceso de aprendizaje sea más difícil y lento.

²⁷ Clasificaciones arbitrarias o complejas

- Construye aproximaciones globales, es decir es capaz de generalizar en regiones del espacio de entrada donde existen pocos datos de entrenamiento (extrapolar).
- Esta red es más rápida durante la etapa de operación que otras redes.

1.5.2 RED FUNCIÓN BASE RADIAL

Las redes función base radial (RBF, por sus siglas en inglés) en su modelo clásico tienen una arquitectura rígida de tres capas: entrada, oculta y salida, (véase Figura 1.18). Su entrenamiento es solamente hacia delante, tienen un aprendizaje no supervisado desde la capa de entrada hacia la capa oculta y un aprendizaje supervisado desde la capa oculta hacia la capa de salida por lo que se considera que son redes *híbridas*.

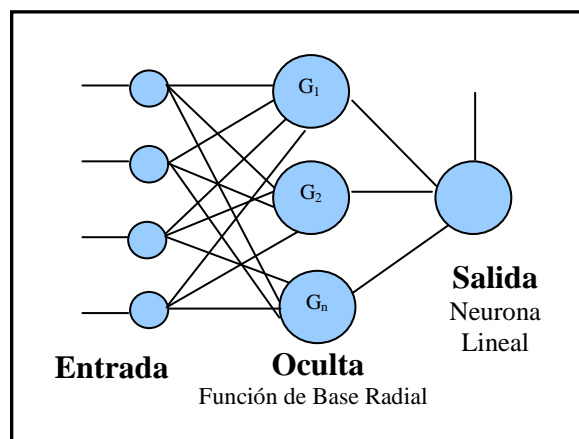


Figura 1.18. Red Función Base Radial.

Construyen sus modelos con funciones de activación que son diferentes para la capa oculta y para la capa de salida. La capa oculta aplica una transformación no lineal (funciones radiales) desde el espacio de entrada hacia su propio espacio; existen diferentes tipos de funciones no lineales siendo la Gaussiana la más utilizada. En cuanto a la capa de salida, ésta es activada a través de funciones lineales.

La capa oculta está compuesta por unidades radiales, cada una de éstas contiene una función base radial que se define por un centro y una

amplitud, así cada unidad o neurona modela una respuesta gaussiana. Esta estructura permite que no se requiera más de una capa oculta para modelar cualquier tipo de función. La activación de una neurona oculta está determinada por la distancia entre el patrón de entrada y el centro asociado a dicha neurona considerando la función radial. Por lo tanto, si una entrada es diferente a las proporcionadas durante el entrenamiento, esa entrada estará lejos del centro por lo que la salida podría ser muy baja o no ocurrir (la neurona no se activa). En el caso contrario, si la entrada es similar significa que debe estar cerca del centro, esto provoca que la neurona se active y reconozca al patrón de entrada. Esta característica hace de este tipo de red muy útil para el reconocimiento de patrones.

La red RBF requiere de la estimación durante el proceso de entrenamiento de parámetros como los centros y las desviaciones estándar (amplitud de la campana de Gauss) de las funciones radiales, lo que implica que se puede modelar a la red para un desempeño adecuado para un caso específico, pero también implica el conocer la forma más apropiada para definir dichos parámetros. En el caso de los centros, éstos deben reflejar el agrupamiento natural de los patrones de entrada, de tal manera que cada entrada forme parte de un grupo asociado a un centro, esa entrada estará más cerca de ese centro que de cualquier otro centro. En cuanto a la desviación estándar, ésta define el área de influencia de la neurona en el espacio de entrada.

Si las gaussianas son muy estrechas la red no interpolará puntos cercanos por lo que perderá la capacidad de generalización, en tanto que si las gaussianas son demasiado amplias la red perderá los detalles finos.

Entre las características más relevantes de este tipo de red están:

- Se la considera un aproximador universal.
- Es un mal extrapolador lo que puede ser útil para problemas de clasificación (presenta baja tasa de falsos positivos).

- Posee la propiedad de mejor aproximación ²⁸.
- Tiene una topología compacta, no utiliza más de una capa oculta. Sin embargo usualmente requiere mayor número de neuronas que otros tipos de redes como por ejemplo las MLP ²⁹.
- Requiere mucho menos tiempo de entrenamiento que otras redes, pero tiende a ser en ocasiones más lenta en la etapa de operación.

1.5.3 RED KOHONEN

La red Kohonen o mapa auto-organizativo es una red del tipo propagación hacia delante, que utiliza un algoritmo de aprendizaje no supervisado basado en la similitud o vecindad.

Este tipo de red está compuesta por dos capas de neuronas: capa de entrada y capa de salida, siendo la capa de salida la responsable de procesar la información y formar a partir de ésta un mapa, (véase Figura 1.19). Se basa en la organización de neuronas en función de las características de los patrones de entrada.

El proceso de entrenamiento se puede describir como: se presenta un patrón a la red, la capa de entrada transmite a la capa de salida esta información. En esta capa las neuronas compiten unas con otras, hasta que sólo una de ellas queda activada y el resto no, la neurona que queda activada es aquella cuyo vector de pesos es el más parecido respecto al patrón. Este proceso se repite para todo el conjunto de patrones de entrenamiento, siempre identificando la neurona de salida ganadora por patrón. Al final se logra que cada neurona responda frente a un determinado patrón y a patrones similares debido a su capacidad de generalización, de tal manera que el mapa que forma la red está

²⁸ Mejor aproximación significa que para cualquier función que se quiera aproximar existe una “configuración” de una red RBF que resulta el método que mejor aproxima a la función. [50]

²⁹ Debido a que las unidades radiales (neuronas ocultas) sólo responden a regiones relativamente pequeñas del espacio de entrada, se requiere un mayor número de unidades radiales mientras mayor sea el número de entradas.

compuesto por las neuronas ganadoras que representan a las clases existentes. Así, cuando se presenta a la red un patrón desconocido por ésta, sólo una de las neuronas de salida o un grupo de vecinas³⁰ se activa, lo que indica la clase a la que pertenece el patrón.

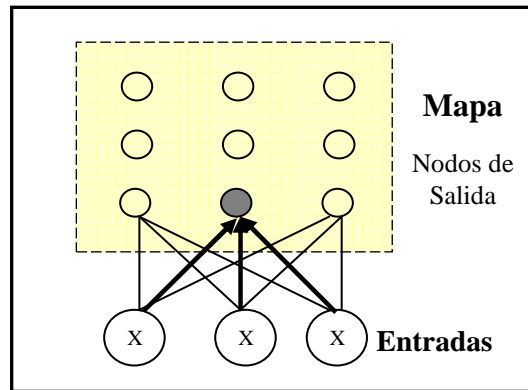


Figura 1.19. Mapa Auto-Organizativo (SOM).

En síntesis las redes Kohonen buscan asociar cada neurona de la capa de salida a un grupo de patrones de entrada con cierta semejanza, generando clases que permitirán la clasificación de los patrones de entrada.

Entre las características más relevantes de este tipo de red están:

- Las neuronas pueden tener múltiples etiquetas de clases al mismo tiempo, lo que permite un mapeo de “uno a muchos”.
- Representa el contenido de cada clase de una forma no lineal gracias a su propiedad de ordenamiento topológico. Esto es muy útil cuando existe un gran número de clases, donde generalmente la transición entre éstas es casi continua.
- Modela y analiza vectores de datos experimentales complejos.
- Requiere una considerable cantidad de tiempo para su entrenamiento.

³⁰ Este tipo de red logra que neuronas próximas se asocien a patrones similares

CAPITULO 2. ANÁLISIS Y DISEÑO

Este capítulo describe las etapas que forman parte de la definición y diseño de la solución computacional que se desea plantear, dentro del contexto exploratorio teórico experimental de este proyecto, para el problema de reconocimiento automático de rostros.

El enfoque seleccionado para el desarrollo de la solución computacional es el sugerido por Microsoft® Solutions Framework versión 3.0 (MSF, por sus siglas en inglés), que es un marco de trabajo compuesto por seis modelos orientados a cubrir diferentes áreas involucradas en el desarrollo de un proyecto tecnológico. Al ser un marco de trabajo y no una metodología, MSF provee un esquema flexible de estos modelos, lo que significa que se pueden ajustar de acuerdo al contexto de cada proyecto; en este sentido MSF permite crear diferentes estrategias de desarrollo. Dentro de estos modelos se encuentra el *Modelo de Procesos* que propone un ciclo de vida de desarrollo basado en algunos conceptos de los modelos cascada y espiral, de tal manera que plantea dividir al proceso en varias iteraciones cortas a fin de poder planificar y comprobar el progreso tanto de cada iteración como del proceso en general, logrando un mayor control durante el desarrollo del proyecto.

Otra característica importante de MSF es que se centra en la arquitectura de la solución presentando diferentes perspectivas de ésta: conceptual, lógica y física. De esta manera provee una visión clara, estructural e integral de la solución, proporcionando un modelo completo que permite visualizar y entender desde diferentes niveles de abstracción, cómo los componentes individuales de dicha solución están interconectados, sin perder la trazabilidad entre niveles.

Estas características hacen que MSF sea el enfoque más adecuado para el desarrollo de esta solución computacional. Adicionalmente se va a utilizar UML como lenguaje de modelado, en tanto que la herramienta para el modelado visual de las vistas y notaciones derivadas de UML será la herramienta case-OO Rational Rose 2002 Enterprise Edition.

La información generada es el resultado de la consolidación de diversos artefactos creados durante el proceso de desarrollo, en varias iteraciones.

2.1 REQUERIMIENTOS

Para realizar una adecuada especificación de requerimientos primero se debe entender el problema que se desea resolver, enfocarse en el entorno de éste y lo que implica resolverlo. Una vez que se ha entendido el problema se debe plantear la visión de la solución, que servirá de guía durante el proceso de desarrollo. Luego se establecen los posibles beneficios que derivaría la solución y finalmente se crea el concepto de la solución dentro del cual se definen los requerimientos o características que ésta debe tener.

2.1.1 ÁMBITO

El presente proyecto es de carácter exploratorio teórico experimental por lo tanto se centra en el problema de reconocimiento automático de rostros como tal, esto significa que no se orienta hacia un área de aplicación específica del reconocimiento de rostros (control de accesos, cajeros automáticos, vigilancia, etc.).

2.1.2 ESTUDIO PRELIMINAR

El estudio preliminar se compone de dos elementos: en primer lugar se analizan los antecedentes y el estado actual de la temática para el ámbito establecido, esto permite entender cuál ha sido el proceso por el que ha atravesado el reconocimiento de rostros hasta llegar a la situación actual destacando algunos avances logrados dentro de este campo. Como segundo elemento se identifica el problema a resolver considerando su entorno y las limitaciones existentes. El contenido de esta sección se encuentra descrito en detalle en el Anexo 10.

2.1.3 VISIÓN

Desarrollar una solución computacional que realice el reconocimiento automático de rostros bajo condiciones específicas, con un nivel de confiabilidad adecuado, minimizando el porcentaje de reconocimientos erróneos.

2.1.4 ANÁLISIS DE BENEFICIOS

De acuerdo con el Grupo Internacional de Biometría – IBG³¹, el mercado de las tecnologías biométricas crece rápidamente. El área de reconocimiento de rostros se presenta como la segunda área dentro de la biometría que proyecta mayores ingresos en los próximos años, después de las huellas digitales.

Según el análisis efectuado por IBG en su *Reporte 2004-2008 del Mercado Biométrico e Industria*,³² el reconocimiento de rostros de acuerdo a proyecciones para el año 2008 reportará un ingreso de \$802 millones de dólares que corresponde al 17.28% del total de ingresos proyectados para ese año por tecnologías biométricas (siendo el segundo tipo de tecnología que más ingresos alcanza luego del reconocimiento de huellas).

Mientras que en el *Reporte 2006-2010 del Mercado Biométrico e Industria*³³ efectuado por la misma firma, se muestra que el reconocimiento de rostros como tecnología biométrica tiene el 19% del mercado biométrico actual y sólo es superado por el reconocimiento de huellas que tiene un 43.6% del mercado. Adicionalmente en este último reporte se proyecta que para el año 2010 los ingresos totales de la industria biométrica serán de aproximadamente \$5,749 millones de dólares.

³¹ El Grupo Internacional de Biometría (IBG) es una firma fundada en 1996 orientada a la investigación biométrica, consultoría e integración de servicios. Tiene una amplia experiencia en las diferentes tecnologías biométricas (implementación de sistemas, estándares, evaluación de tecnologías, etc.).

³² Referencia Secundaria *IEEE Spectrum*.

³³ Referencia Secundaria [40].

Además de esto, el reconocimiento automático de rostros presenta dos grandes beneficios: costos bajos y bajo nivel de intrusión, (véase Tabla 2.1)

- a. *Costos bajos*: Es una de las tecnologías biométricas de más bajo costo ya que no requiere de dispositivos complejos que involucren una gran inversión. Muchos sistemas de reconocimiento de rostros utilizan únicamente una cámara web para capturar la imagen y un computador para realizar el proceso de reconocimiento, por lo que el costo del equipo requerido es inferior al requerido por otras tecnologías biométricas (huellas digitales, iris, entre otros).
- b. *Bajo nivel de intrusión*: Únicamente requiere la imagen del rostro de una persona por lo que el grado de colaboración que se necesita es mínimo.

Tecnología Biométrica	Nivel de Intrusión	Costo	Hardware Utilizado
Huellas Digitales	Bajo	Bajo	Especial, barato
Reconocimiento de Rostros	Ninguno	Bajo	No especializado, barato
Geometría de la mano	Ninguno	Alto	Especial, costo medio
Reconocimiento de Voz	Ninguno	Bajo	No especializado, barato
Escaneo Iris	Ninguno	Alto	Especial, caro
Escaneo Retina	Alto	Alto	Especial, caro
Reconocimiento de Firma	Ninguno	Bajo	Especial, costo medio

Tabla 2.1. Costos y nivel de intrusión de las distintas tecnologías biométricas

Fuente: Información extraída de NCSC. BIOMETRIC COMPARISON CHART. <http://ctl.ncsc.dni.us/biomet%20web/BMCompare.html>

En síntesis el reconocimiento automático de rostros es una opción real que puede ser implementada con un costo bajo, mínimo nivel de intrusión y que tiene una gran proyección de uso en los próximos años.

2.1.5 CONCEPTO DE LA SOLUCIÓN

2.1.5.1 Meta

La solución computacional a desarrollarse tendrá como propósito localizar un rostro en una imagen, extraer sus características más relevantes y realizar la verificación de la identidad de dicho rostro en el menor tiempo posible con la ayuda de una red neuronal artificial previamente creada y entrenada para esta tarea.

2.1.5.2 Objetivos

- Realizar el preprocesamiento de la imagen obtenida para así eliminar dentro de lo posible cualquier artefacto que pueda influir en el proceso de localización de un rostro o de verificación de una identidad.
- Localizar y extraer el rostro contenido en la imagen.
- Normalizar la nueva imagen en cuanto a tamaño y a orientación del rostro.
- Extraer las características que contengan la información más discriminante de un rostro.
- Identificar una arquitectura de red neuronal apropiada y realizar su entrenamiento mediante vectores de características asociados a un conjunto de usuarios.
- Realizar la verificación de la identidad de un rostro en el menor tiempo posible, empleando la red neuronal artificial antes entrenada.

2.1.5.3 Alcance

La solución computacional (a) será capaz de verificar la identidad de un usuario, para esto deberá recibir como entradas una imagen que contenga el rostro del usuario y una clave. La imagen comprenderá el

rostro del usuario visto de frente considerando variantes como el uso de lentes, diferentes cortes de cabello o estilos de peinado, uso de barba y distintas expresiones faciales (variación moderada); *(b)* también se encargará del proceso de registro o enrolamiento de nuevos usuarios. Cada registro se realizará de manera independiente y requerirá al menos una imagen del nuevo usuario o hasta un máximo de cinco imágenes, así como los datos personales de éste (nombre, apellido y género) y una clave. Todo usuario tendrá una clave única que lo identificará pues esta servirá posteriormente para el proceso de verificación. De igual manera, las imágenes proporcionadas para el registro deberán tener las mismas características que las descritas para la verificación; *(c)* empleará técnicas para eliminar ruido y compensar la iluminación en la imagen que reciba como entrada para el registro de un nuevo usuario o para la verificación de identidad, luego de esto efectuará la localización y extracción del rostro en la imagen *(d)* realizará la extracción de las características más relevantes de un rostro; *(e)* permitirá crear y entrenar a la red neuronal que se utilizará para la verificación de identidad. La red será entrenada con información de los usuarios registrados, por lo tanto la solución computacional verificará previamente que exista un número adecuado de usuarios registrados; *(f)* permitirá la modificación de ciertos parámetros relacionados con la creación y entrenamiento de la red neuronal; *(g)* permitirá obtener imágenes estáticas que estén almacenadas (disco, medios extraíbles, etc.) o capturarlas a través de una cámara digital, tanto para el registro de usuarios como para la verificación; *(h)* ofrecerá la posibilidad de guardar una copia de la imagen que reciba como entrada y también una copia de la misma imagen pero con el área del rostro detectada; *(i)* permitirá actualizar claves de usuario; y *(j)* elaborará un reporte de los usuarios registrados.

Por otro lado la solución computacional *(a)* no estará orientada a la gestión de usuarios por lo tanto no se podrá realizar la actualización de los datos personales o de las imágenes de los usuarios así como tampoco se podrá eliminar usuarios registrados; *(b)* no podrá detectar

múltiples rostros en una imagen ni verificar varias identidades a la vez; (c) no localizará rostros ni verificará identidades en imágenes dinámicas (video), esto significa que no empleará este tipo de imágenes; (d) no se encargará de operaciones como rotación, calibración o configuración de propiedades de la cámara digital utilizada para la captura de imágenes.

2.1.5.4 Supuestos

- Para la captura de imágenes se asume la existencia de una cámara web o de algún otro tipo de cámara digital que utilice la interfaz genérica de Windows para dispositivos de video (dispositivos con controladores WDM o VFW).
- La cámara debe estar correctamente instalada y funcionando adecuadamente, además debe proporcionar imágenes de buena calidad (el rostro en la imagen se debe distinguir claramente y con suficiente detalle, sin degradación del color).
- Se asume que cualquier imagen utilizada cumplirá las características descritas como parte de las restricciones del proyecto.
- Se asume que la plataforma de ejecución es Windows empleando bajo este entorno Microsoft Access 2000 para la base de datos y Matlab en su versión 7.0 como herramienta de desarrollo, con las librerías, toolbox y los paquetes necesarios, instalados previamente para el correcto funcionamiento de la solución computacional.
- Se asume que el hardware utilizado cumple con los requerimientos necesarios para la correcta ejecución de la solución computacional, especialmente en relación a tiempos de respuesta.
- Se asume que la herramienta de desarrollo proveerá el soporte necesario en cuanto a funcionalidad para llevar a cabo el proyecto.

2.1.5.5 Restricciones y Dependencias

A continuación se recogen ciertas restricciones y dependencias que deben ser consideradas durante el diseño e implementación de la solución computacional:

a. El reconocimiento de rostros está estrechamente relacionado a la calidad de las imágenes utilizadas por lo que se deben establecer ciertas características específicas y restricciones en torno a éstas:

- La imagen debe abarcar el rostro del usuario completo y visto de frente. La rotación de la cabeza debe ser menor a +/- 10 grados de la posición frontal hacia cualquier dirección (arriba, abajo, izquierda, derecha).
- La imagen puede contener variantes como el uso de lentes, diferentes cortes de cabello o estilos de peinado, uso de barba o variaciones moderadas en las expresiones faciales (no contorsiones faciales extremas). No se admite el uso de elementos que cubran parcial o completamente alguna región del rostro.
- Durante la captura de imágenes, el usuario debe ver de frente a la cámara. No se admiten imágenes en las cuales los ojos del usuario estén cerrados, se encuentren cubiertos por el cabello o no se logren ver claramente.
- Cada imagen debe contener sólo el rostro de una persona, se debe evitar imágenes que muestren varios rostros aunque se encuentren parcial o completamente visibles.
- La iluminación debe estar distribuida equitativamente en cada lado del rostro, por lo que no debe existir sombras visibles o una marcada iluminación en cualquier dirección. De igual manera debe existir la iluminación suficiente para apreciar completamente el rostro, no se admiten imágenes en las que alguna parte del rostro no sea visible a causa de falta de iluminación.

- Si el usuario utiliza lentes, éstos deben ser transparentes. Si se produce reflejo de luz sobre los lentes, el usuario debe quitárselos.
 - Durante la captura de una imagen, la distancia entre la cámara y el rostro del usuario debe ser entre 30 a 60 cm. No se admiten imágenes en las que el alto del rostro sea menor al 50% del alto total de la imagen.
 - Las imágenes deben tener fondos simples, se debe evitar fondos con varios objetos y colores a la vez.
 - Las imágenes no deben ser borrosas o mal enfocadas, tampoco deben presentar degradación en los colores por efectos del dispositivo de captura.
 - Se trabajará con imágenes a color (24 bits), formatos: bmp y jpg.
 - Se recomienda que la resolución de imagen no sea menor a 160 x 190, para una resolución superior a 640 x 480 se requerirá mayor tiempo de procesamiento. Por lo que se sugiere una resolución de 320 x 240.
- b. Para la captura de imágenes no se contempla el uso de dispositivos diferentes a los que utilicen controladores WDM o VFW .
- c. El desempeño de la solución computacional estará sujeto a las limitaciones que presentan los métodos para reconocimiento automático de rostros que actualmente existen y que pueden ser implementados para el presente proyecto (considerando limitaciones tecnológicas), así como a la calidad de las imágenes proporcionadas.
- d. Existen restricciones relacionadas con la herramienta de desarrollo (Matlab 7.0), limitaciones de la versión utilizada.

2.1.5.6 Especificación de Casos de uso

2.1.5.6.1 Definición de Actores

La Tabla 2.2 muestra dos tipos de actores, los actores primarios que son quienes pueden recurrir a los servicios de la solución computacional para cumplir un objetivo y los actores secundarios que son requeridos por la solución computacional para alcanzar algún objetivo de un actor primario.

ACTORES PRIMARIOS	
Operador	
Descripción	Representa a cualquier persona que interactúe con la solución computacional ya sea para registrar un nuevo usuario, entrenar la red neuronal, verificar una identidad o para modificar parámetros de creación y entrenamiento de la red neuronal. Este actor no puede efectuar operaciones de administración.
Administrador	
Descripción	Tiene la facultad de ejecutar funciones de administración existentes en la solución computacional (actualizar claves de usuarios y generar el reporte de usuarios registrados), además puede realizar las mismas tareas que un Operador. Requiere contar con perfil de administrador del equipo o dominio de acuerdo a la autenticación Windows.
ACTORES SECUNDARIOS	
Dispositivo de Captura	
Descripción	Representa a cualquier cámara digital que utilice la interfaz genérica de Windows para dispositivos de video (cámaras digitales que utilicen controladores WDM o VFW).

Tabla 2.2. Descripción de los actores que intervienen en los casos de uso

A continuación se detallan los casos de uso que componen la funcionalidad básica de la solución computacional. El Anexo 12 detalla los casos de uso restantes (funcionalidad adicional).

2.1.5.6.2 Estudio de Casos de uso

CASO DE USO :		REGISTRAR USUARIO
Actor:		Operador (O)
Propósito:		Registrar un nuevo usuario.
Descripción:		
Este caso de uso permite llevar a cabo el proceso de registro de un nuevo usuario (obtener la información requerida y luego almacenarla). Para registrar a un usuario se requiere de al menos una imagen de su rostro hasta un máximo de cinco imágenes, adicionalmente se requieren ciertos datos (nombre, apellido, clave y género).		
Precondición:		Ninguna
Secuencia Normal: (eventos)	Acción del Actor	Respuesta de la Solución Computacional
	1. Este caso de uso inicia cuando Operador solicita registrar un nuevo usuario.	2. Presenta las tareas asociadas al registro de un nuevo usuario.
	3. Selecciona la forma de obtener una imagen que contiene el rostro del nuevo usuario (cargar una imagen ya existente o capturarla con una cámara digital).	4. <i>Se realiza el caso de uso Obtener Imagen.</i>
		5. Muestra la imagen obtenida y permite al Operador solicitar la localización del rostro en dicha imagen.
	6. Solicita localizar un rostro en la imagen obtenida.	7. Verifica que el número de imágenes del nuevo usuario que ya se han agregado para el registro es menor a cinco.
		8. <i>Se realiza el caso de uso Localizar Rostro.</i>
		9. Pregunta al Operador si desea agregar el rostro localizado al conjunto de imágenes del nuevo usuario.
	10. Confirma que desea agregar el rostro localizado a las imágenes del nuevo usuario.	11. Si el Operador ha agregado la primera imagen del nuevo usuario, genera un conjunto de imágenes sintéticas a partir de ésta y le permite al Operador ingresar los datos del usuario y solicitar almacenar la información para completar el registro.
	12. Agrega el rostro localizado al conjunto de imágenes que pertenecen al nuevo usuario.	

	Puede repetir pasos 3-10 hasta obtener el número de imágenes que desee para el registro del nuevo usuario (máximo 5 imágenes).	
	13. Ingresar los datos del nuevo usuario.	14. Validar los datos.
	15. Solicitar que se almacene la información del nuevo usuario.	16. Validar nuevamente los datos y verificar que no existe un usuario previamente registrado con esa clave.
		17. Almacenar la información del nuevo usuario.
		18. Informar al Operador que el proceso de registro ha finalizado con éxito. El caso de uso termina.
Poscondiciones:	Se ha almacenado la información correspondiente al nuevo usuario por lo tanto éste se encuentra registrado.	
Extensiones:	Paso	Acción
	5	a. Si el Operador no obtuvo una imagen. 1. S: Permanece en espera de una nueva solicitud por parte del Operador.
	8	a. Si el número de imágenes es igual o mayor a cinco. 1. S: Informa al Operador que no se puede agregar más imágenes para el registro. 2. S: El caso de uso continúa (paso 13).
	9	a. Si no se localizó ningún rostro en la imagen. 1. S: Informa de esta situación al Operador indicándole la posible razón por la que no se encontró un rostro en la imagen. 2. S: Permanece en espera de una nueva solicitud por parte del Operador.
	10	a. Si el Operador no desea agregar el rostro localizado a las imágenes del nuevo usuario. 1. S: Permanece en espera de una nueva solicitud por parte del Operador y no agrega la imagen.
	14	a. Si los datos ingresados son incorrectos. 1. S: Solicita al Operador que verifique los datos ingresados y le permite modificarlos. 2. S: El caso de uso continúa.
	16	a. Si los datos ingresados son incorrectos. Análogo a 14a.

	16	<p>b. Si existe un usuario previamente registrado con la clave ingresada.</p> <ol style="list-style-type: none"> 1. S: Solicita al Operador que verifique la clave y le permite modificarla. 2. S: El caso de uso continúa (paso 13).
	18	<p>a. Si no se logró completar la tarea de almacenamiento</p> <ol style="list-style-type: none"> 1. S: Informa de esta situación al Operador y le permite volver a solicitar que la información sea almacenada. 2. S: El caso de uso continúa (paso 15).
	3, 6, 10, 13, 15	<p>a. Si el Operador solicita cancelar el proceso de registro.</p> <ol style="list-style-type: none"> 1. S: Cancela todas las acciones realizadas previamente. El caso de uso termina.

Tabla 2.3. Descripción del caso de uso Registrar Usuario

CASO DE USO :		OBTENER IMAGEN (inclusión)
Actor:	Operador (O)	
Propósito:	Obtener una imagen desde la fuente especificada por el usuario.	
Descripción:		
<p>Permite obtener la imagen estática que se va a utilizar para el registro de un nuevo usuario o para la verificación de una identidad desde una de las siguientes fuentes: desde cualquier unidad de almacenamiento o mediante una cámara digital conectada al computador donde se ejecute la solución computacional.</p> <p>Para cada fuente existe una forma de adquisición diferente, en el caso que la imagen resida en una unidad de almacenamiento se debe especificar la ubicación de dicha imagen, mientras que si la fuente es una cámara digital la imagen será capturada en ese momento.</p>		
Precondición:	Requiere que el Operador haya iniciado previamente el caso de uso Registrar Usuario o el caso de uso Verificar Identidad.	
	Acción del Actor	Respuesta de la Solución Computacional
Secuencia Normal:	1. Este caso de uso inicia cuando el Operador selecciona la forma de obtener una imagen.	

(EVENTOS)	2.1 Solicita obtener una imagen existente.	3.1 Solicita seleccionar la imagen.
	4.1 Selecciona la imagen.	5.1 Obtiene la imagen seleccionada.
	2.2 Solicita capturar una imagen.	3.2 <i>Se realiza el caso de uso Detectar Cámara.</i>
		4.2 Pregunta al Operador si desea capturar la imagen a través del dispositivo detectado.
	5.2 Confirma que desea capturar la imagen.	6.2 Adquiere una imagen a través del dispositivo de captura detectado.
		7. El caso de uso termina
Poscondiciones:	Se ha obtenido la imagen con la que se va a trabajar desde la fuente especificada por el usuario.	
Extensiones:	Paso	Acción
	4.1	a. Si el Operador solicita cancelar esta operación. 1. S: El caso de uso termina.
	4.1	b. Si el Operador intenta obtener una imagen no existente. 1. S: Solicita al Operador que verifique qué imagen desea seleccionar. 2. S: El caso de uso continúa (paso 4.1).
	4.2	a. Si no se ha detectado un dispositivo. 1. S: Informa de esta situación al Operador y le permite solicitar una nueva detección del dispositivo de captura. Si el Operador no desea una nueva detección del dispositivo el caso de uso termina.
	5.2	a. Si el Operador no desea capturar la imagen. 1. S: El caso de uso termina.
6.2	a. Si no se logró capturar la imagen desde el dispositivo. 1. S: Informa de esta situación al Operador. 2. S: El caso de uso termina.	

Tabla 2.4. Descripción del caso de uso Obtener Imagen

CASO DE USO :		DETECTAR CÁMARA (extensión)
Actor:		
Propósito:	Detectar si existe alguna cámara compatible conectada.	
Descripción: Detecta la existencia de una cámara que utilice la interfaz genérica de Windows para dispositivos de video y que esté conectada al computador donde se ejecute la solución computacional.		
Precondición:	Requiere que el Operador haya solicitado capturar una imagen como forma de obtener la imagen.	
Secuencia Normal: (eventos)	Acción del Actor	Respuesta de la Solución Computacional
	1. Este caso de uso inicia cuando el Operador solicita capturar una imagen.	2. Detecta dispositivos que usen la interfase genérica de Windows para dispositivos de video y que estén conectados al computador.
		3. Selecciona el primer dispositivo de este tipo encontrado.
		4. El caso de uso termina
Poscondiciones:	Se ha detectado una cámara compatible conectada.	
Extensiones:	Paso	Acción
	3	a. Si no se ha detectado ningún dispositivo compatible conectado. 1. S: El caso de uso termina.

Tabla 2.5. Descripción del caso de uso Detectar Cámara

CASO DE USO :		LOCALIZAR ROSTRO (inclusión)
Actor: Operador (O)		
Propósito:	Localizar un rostro en una imagen y mostrar su ubicación.	
Descripción: Realiza el preproceso de la imagen y determina si existe o no un rostro en ésta. Muestra la ubicación del rostro dentro de la imagen y lo extrae como una nueva imagen.		
Precondición:	Requiere que el Operador haya iniciado previamente el caso de uso Registrar Usuario o el caso de uso Verificar Identidad y que haya obtenido la imagen que se va a utilizar.	

Secuencia Normal:	Acción del Actor	Respuesta de la Solución Computacional
(eventos)	1. Este caso de uso inicia cuando el Operador solicita localizar un rostro en la imagen obtenida.	2. Realiza el preproceso de la imagen para facilitar la localización del rostro contenido en ésta
		3. Localiza el rostro dentro de la imagen.
		4. Extrae el área del rostro como una nueva imagen (imagen del rostro).
		5. Muestra la ubicación del rostro en la imagen inicial.
		6. El caso de uso termina
Poscondiciones:	Se ha localizado un rostro dentro de la imagen suministrada por el usuario, se ha extraído dicho rostro como una nueva imagen y se muestra su ubicación en la imagen inicial.	
Extensiones:	Paso	Acción
	4	a. Si no se ha localizado un rostro en la imagen. 1. S: El caso de uso termina.

Tabla 2.6. Descripción del caso de uso Localizar Rostro

CASO DE USO : VERIFICAR IDENTIDAD		
Actor:	Operador (O)	
Propósito:	Verificar la identidad de un usuario.	
Descripción:	Este caso de uso permite llevar a cabo la verificación de identidad de un usuario utilizando una imagen del rostro del usuario y una clave para autenticarlo.	
Precondición:	Deben existir usuarios registrados y una red neuronal entrenada para la tarea de verificación de estos usuarios.	
Secuencia Normal:	Acción del Actor	Respuesta de la Solución Computacional
	1. Este caso de uso inicia cuando el Operador solicita comenzar el proceso de verificación de identidad.	2. Presenta las tareas requeridas para la verificación de identidad de un usuario.

(eventos)	3. Selecciona la forma de obtener la imagen: cargar una ya existente o capturarla con una cámara digital.	4. <i>Se realiza el caso de uso Obtener Imagen.</i>
		5. Muestra la imagen obtenida y permite al Operador solicitar la localización del rostro en dicha imagen.
	6. Solicita localizar un rostro en la imagen obtenida.	7. <i>Se realiza el caso de uso Localizar Rostro.</i>
		8. Pregunta al Operador si desea utilizar el rostro localizado para la verificación de identidad.
	9. Confirma que desea utilizar el rostro localizado.	10. Extrae las características más relevantes del rostro, creando así un patrón para ese rostro.
		11. Permite al Operador ingresar la clave de usuario y solicitar la verificación de identidad.
	12. Ingresa la clave de usuario.	
	13. Solicita que se verifique la identidad del usuario utilizando el rostro localizado y la clave ingresada.	14. Valida la clave y verifica que exista un usuario ya registrado con esa clave.
		15. Envía las características extraídas del rostro hacia la red neuronal.
		16. Obtiene la respuesta de la red neuronal, la interpreta y la compara con la identidad proporcionada.
		17. Informa al Operador que la imagen del rostro sí corresponde a la persona cuya clave se ingresó.
		18. Muestra la información almacenada de ese usuario (datos e imágenes).
		19. El caso de uso termina.
Poscondiciones:	Se ha verificado la identidad del usuario empleando una imagen de su rostro y su clave. Se muestra la información del usuario.	
Extensiones:	Paso	Acción
	5	a. Si el Operador no obtuvo una imagen. 1. S: Permanece en espera de una nueva solicitud por parte del Operador.

	8	a. Si no se localizó ningún rostro en la imagen. 1. S: Informa de esta situación al Operador indicándole la posible razón por la que no se encontró un rostro en la imagen. 2. S: Permanece en espera de una nueva solicitud por parte del Operador.
	9	a. Si el Operador no desea utilizar el rostro localizado para la verificación. 1. S: Permanece en espera de una nueva solicitud por parte del Operador.
	14	a. Si la clave ingresada no es válida. 1. S: Solicita al Operador que verifique la clave ingresada y le permite modificarla. 2. S: El caso de uso continúa (paso 13).
	14	b. Si la clave no pertenece a un usuario registrado. 1. S: Solicita al Operador que verifique la clave ingresada y le permite modificarla. 2. S: El caso de uso continúa (paso 13).
	17	a. Si la clave ingresada no pertenece al rostro de la imagen. 1. S: Informa de esta situación al Operador. 2. S: El caso de uso termina.
	3, 6, 12, 13	a. Si el Operador solicita cancelar el proceso de verificación de identidad. 1. S: Cancela todas las acciones realizadas previamente. 2. S: El caso de uso termina.

Tabla 2.7. Descripción del caso de uso Verificar Identidad

CASO DE USO : ENTRENAR RED NEURONAL	
Actor:	Operador (O)
Propósito:	Crear y entrenar una red neuronal para realizar la tarea de verificación de usuarios registrados.
Descripción:	Este caso de uso permite llevar a cabo el proceso de creación y entrenamiento de la red neuronal artificial que luego será empleada para la verificación de identidad.
Precondición:	Deben existir al menos tres usuarios previamente registrados.

	Acción del Actor	Respuesta de la Solución Computacional
Secuencia Normal: (eventos)	1. Este caso de uso inicia cuando el Operador solicita comenzar el entrenamiento de la red neuronal.	2. Obtiene las imágenes almacenadas de todos los usuarios registrados.
		3. Extrae las características más relevantes de cada imagen de cada usuario registrado.
		4. Crea la red neuronal y la entrena utilizando las características antes extraídas y empleando los parámetros definidos para la red.
		5. Almacena la red neuronal y cualquier información requerida para el proceso de verificación.
		6. Informa al Operador que el proceso de entrenamiento de la red finalizó correctamente.
		7. El caso de uso termina.
Poscondiciones:	Se ha almacenado la red neuronal ya entrenada y cualquier información adicional que se requiera para una posterior verificación de identidad.	
Extensiones:	Paso	Acción
	1	a. Si el Operador solicita cancelar el proceso de entrenamiento de la red. 1. S: El caso de uso termina.
	3	a. Si ocurre un error al obtener las imágenes de los usuarios registrados. 1. S: Informa de esta situación al Operador. 2. S: El caso de uso termina.
	4	a. Si no se logró extraer las características. 1. S: Informa al Operador que el proceso de entrenamiento de la red no se realizó. 2. S: El caso de uso termina.
	5	a. Si no se creó o entrenó a la red neuronal. 1. S: Informa al Operador que el proceso de entrenamiento de la red no se realizó. 2. S: El caso de uso termina.
	6	a. Si no se logró almacenar la red neuronal o cualquier información requerida para el proceso de verificación. 1. S: Informa al Operador que el proceso de entrenamiento de la red no se realizó. 2. S: El caso de uso termina.

Tabla 2.8. Descripción del caso de uso Entrenar Red Neuronal

CASO DE USO : ACTUALIZAR PARÁMETROS DE LA RED NEURONAL		
Actor:	Operador (O)	
Propósito:	Permitir la actualización de los valores de dos parámetros utilizados para la creación y entrenamiento de la red neuronal.	
Descripción:	Permite al usuario modificar dos parámetros relacionados a la creación y entrenamiento de la red neuronal. Estos parámetros son: número de neuronas añadidas por desplazamiento y el error mínimo cuadrático estándar esperado para el entrenamiento de la red.	
Precondición:	Ninguna	
Secuencia Normal: (eventos)	Acción del Actor	Respuesta de la Solución Computacional
	1. Este caso de uso inicia cuando el Operador solicita modificar los parámetros de la red neuronal.	2. Muestra los valores actuales de los parámetros que se pueden modificar.
	3. Ingresar los nuevos valores de los parámetros.	
	4. Solicita almacenar la nueva información.	5. Valida los valores ingresados.
		6. Modifica los valores actuales de los parámetros.
		7. El caso de uso termina.
Poscondiciones:	Se han actualizado los valores de los parámetros de la red neuronal.	
Extensiones:	Paso	Acción
	3, 4	a. Si el Operador solicita cancelar el proceso. 1. S: El caso de uso termina.
	5	a. Si los datos ingresados no son válidos. 1. S: Solicita al Operador que verifique los datos ingresados y le permite modificarlos. 2. S: El caso de uso continúa (paso 4).
	6	a. Si no se logró modificar los valores de los parámetros. 1. S: Informa de esta situación al Operador. 2. S: El caso de uso termina.

Tabla 2.9. Descripción del caso de uso Actualizar Parámetros de la Red Neuronal

2.1.5.6.3 Diagrama del modelo de Casos de uso

Las Figuras 2.1 y 2.2 muestran el diagrama del modelo de casos de usos para la solución computacional:

Diagrama de Casos de Uso para FENIXVR - Verificación de Rostros

Parte I : Funcionalidad Básica

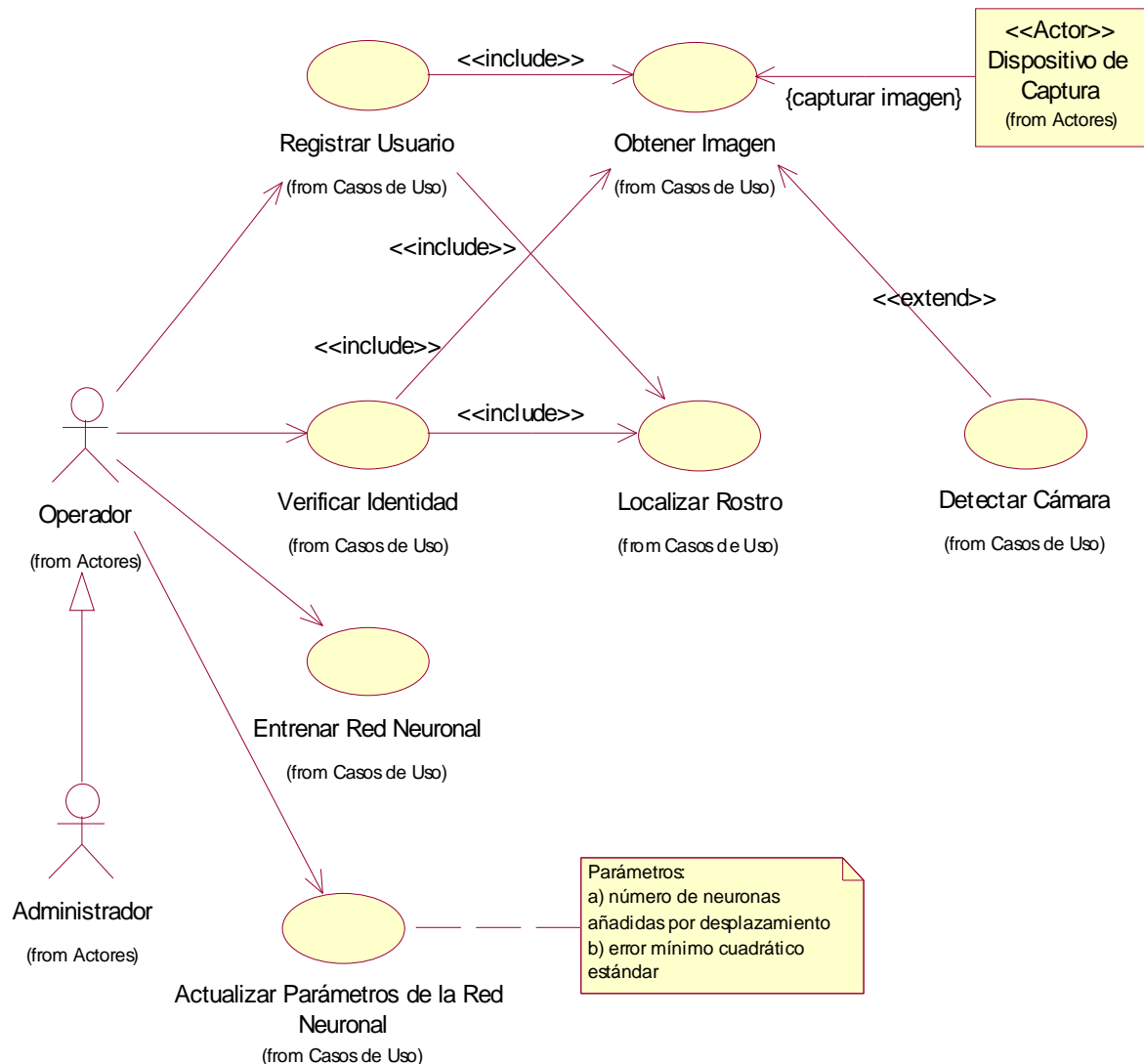


Figura 2.1. Diagrama de Casos de uso para FENIXVR – Verificación de Rostros (Parte I: Funcionalidad Básica)

Diagrama de Casos de Uso para FENIXVR - Verificación de Rostros

Parte II : Funcionalidad Adicional

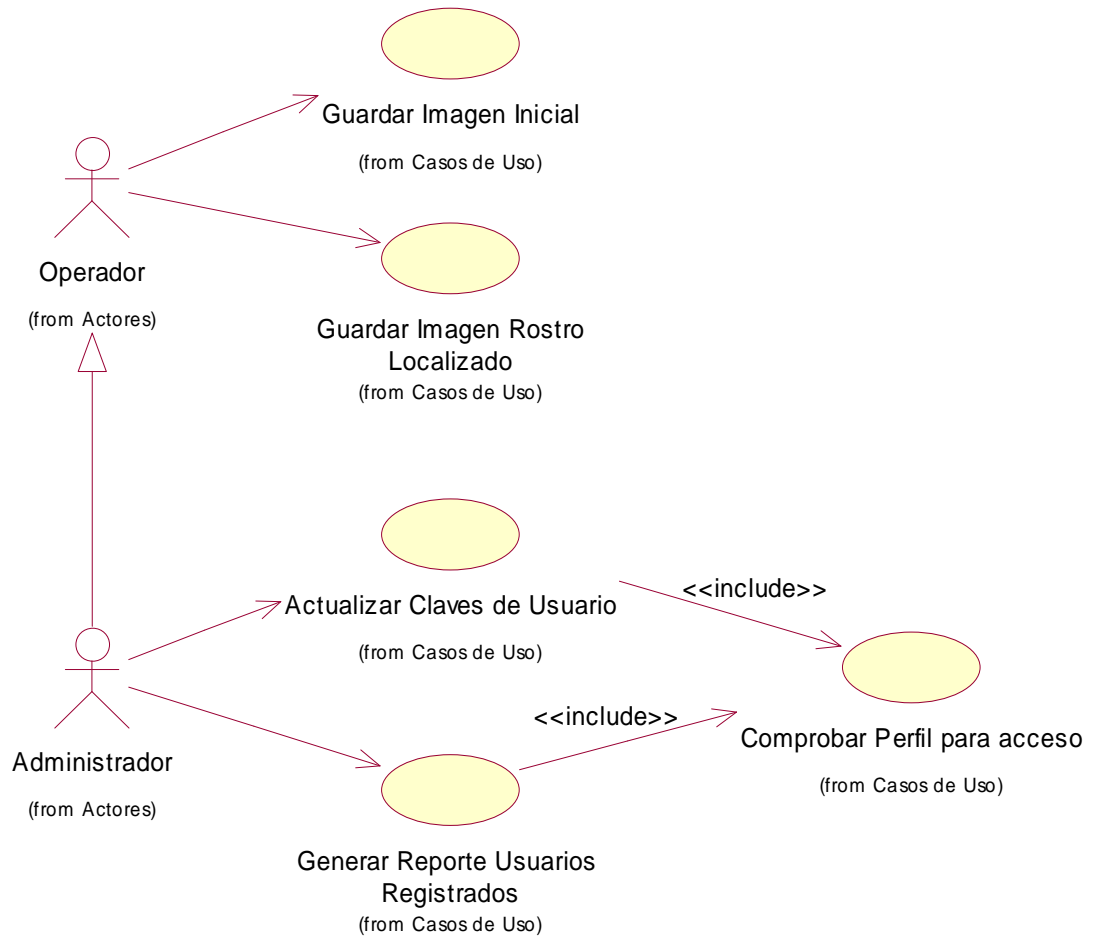


Figura 2.2. Diagrama de Casos de Uso para FENIXVR – Verificación de Rostros (Parte II: Funcionalidad Adicional)

2.1.5.7 Especificación de Requerimientos

Esta sección describe las características de la solución computacional que se va a desarrollar. Se definen desde diferentes puntos de vista: requerimientos funcionales, requerimientos de desempeño y operacionales, requerimientos de usuario y requerimientos de entorno.

2.1.5.7.1 *Requerimientos Funcionales*

La solución computacional que se va a desarrollar no constituye un elemento o componente de una solución mayor. Sin embargo debe ser soportada por la herramienta de desarrollo utilizada (Matlab 7.0).

Funciones

Las principales funciones que llevará a cabo la solución computacional se detallan a continuación:

- a. Preprocesar la imagen.
 - Eliminar ruido y compensar la iluminación en la imagen.
- b. Localizar y extraer el área del rostro a partir de la imagen original.
 - Localizar el rostro.
 - Normalizar la imagen en cuanto orientación del rostro.
 - Extraer el área del rostro como una nueva imagen y normalizar su tamaño.
 - Mejorar la nueva imagen.
- c. Registrar un nuevo usuario.
- d. Extraer las características más relevantes de los rostros de los usuarios registrados.
- e. Crear y entrenar una red neuronal proporcionándole vectores de características obtenidos para el conjunto de usuarios registrados.
- f. Permitir modificar parámetros relacionados a la creación y entrenamiento de la red neuronal.

- g. Extraer las características más relevantes del rostro cuya identidad se desea verificar.
- h. Verificar la identidad de un usuario con ayuda de la red neuronal previamente entrenada.
- i. Permitir obtener la imagen estática con la que se va a trabajar, desde cualquier unidad de almacenamiento o capturarla a través de una cámara digital.
- j. Permitir guardar una copia de la imagen original o una copia de la imagen con el rostro localizado.
- k. Permitir la actualización de claves de usuario.
- l. Generar un reporte de usuarios registrados.

Fuera del Alcance

Existen ciertas funciones que se encuentran fuera del alcance de la solución computacional:

- a. Gestión de usuarios: La solución computacional no podrá realizar la actualización de los datos personales o de las imágenes de los usuarios así como tampoco podrá eliminar usuarios registrados.
- b. Detección y Verificación múltiple: La solución computacional no podrá detectar múltiples rostros en una imagen ni verificar varias identidades a la vez.
- c. Utilización de video: La solución computacional no localizará rostros ni verificará identidades en imágenes dinámicas (video).
- d. Rotación, calibración y configuración de la cámara: La solución computacional no realizará la rotación automática de la cámara, ni efectuará operaciones de calibración o configuración de ésta. La cámara se deberá ubicar de forma manual y se utilizará el formato que tenga por defecto.

2.1.5.7.2 *Requerimientos de Desempeño y Operacionales de la Solución Computacional*

Tiempo de Respuesta

El tiempo de respuesta de la solución computacional sólo se considerará para la etapa de verificación de identidad. Al ser una etapa orientada para ejecutarse en el menor tiempo posible (cercano a tiempo real) se establece que el tiempo de respuesta debe ser inferior a 20 segundos. Este tiempo se fija desde que inicia el proceso de localización del rostro en la imagen proporcionada hasta que se emite el resultado de la verificación (considerando que la localización del rostro fue exitosa).

Confiabilidad

Se espera que la solución computacional tenga un nivel de confiabilidad adecuado. Esto se expresará a través del porcentaje de aciertos obtenidos durante la etapa de pruebas, que se espera esté entre el 80% al 95% (tomando en cuenta las limitaciones de los métodos actuales para reconocimiento de rostros).

También se considera que los resultados deben mostrar porcentajes bajos de error (FAR y FRR), en especial FAR.

Concurrencia

No existirá concurrencia de usuarios. Toda imagen que se utilice deberá contener un sólo rostro, por lo tanto los procesos de registro de usuario y verificación de identidad se llevarán a cabo para una sola persona a la vez.

Escalabilidad

La escalabilidad no se definirá en términos de crecimiento del negocio debido a que el presente proyecto es de carácter exploratorio teórico experimental y por lo tanto no está relacionado a un área de aplicación específica. Se espera que la solución se ejecute sin presentar problemas

o degradación en el tiempo de respuesta para el entorno que se utilice durante las pruebas, sin embargo en términos generales el tamaño de la población que la solución computacional podrá manejar sin degradación estará en relación directa con las capacidades tanto de la red neuronal artificial como de la base de datos utilizada.

Soporte

- o Se deben definir estándares generales para la codificación a fin de obtener una solución computacional bien documentada, de tal manera que se pueda realizar operaciones de mantenimiento de la solución con el menor esfuerzo posible.

2.1.5.7.3 *Requerimientos de Usuario*

Experiencia de los Usuarios

Se deben distinguir dos tipos de usuarios de la solución computacional:

- o Operador
- o Administrador

Se denomina *Operador* a toda persona que interactúa con la solución computacional pero que no tiene acceso a las funciones de administración (actualizar claves de usuario y generar el reporte de usuarios registrados). No se requiere que este tipo de usuario tenga conocimientos o experiencia en un área específica, sin embargo es recomendable que conozca las características que deben tener las imágenes para un correcto desempeño de la solución computacional y también cómo lograr obtener imágenes bajo esos lineamientos.

Adicionalmente es aconsejable que tenga al menos una mínima experiencia en el manejo de aplicaciones bajo entorno Windows y en el manejo de cámaras digitales si va a capturar las imágenes y no las va a cargar desde disco.

El *Administrador* es un *Operador* que tiene acceso a las funciones de administración de la solución computacional, por lo tanto debe tener los conocimientos o experiencia descrita para el *Operador*. Adicionalmente el *Administrador* debe tener conocimientos básicos sobre el manejo de Microsoft Access y contar con perfil de administración del equipo o dominio sobre el que se ejecuta la solución computacional, de acuerdo a la autenticación Windows.

Dependiendo del entorno en el que se utilice la solución computacional se puede requerir los dos tipos de usuarios o sólo requerirse el rol de *Administrador*. En el presente proyecto solo se requerirá del rol de *Administrador*.

Usabilidad

Los usuarios utilizarán la solución computacional ha desarrollarse de forma intuitiva sin requerir un proceso de capacitación.

- Se utilizarán interfaces gráficas (GUI) reduciendo cualquier tipo de complejidad que pueda existir para el usuario.
- Se presentará y se solicitará la información claramente, minimizando el uso de abreviaturas o términos que puedan provocar confusión al usuario.
- Se mantendrá informado al usuario sobre el avance de un proceso cuando el tiempo requerido para la ejecución de éste pueda ser percibido por el usuario. Para esto se cambiará la forma del cursor o puntero y se presentará una barra de progreso.
- Se emplearán barras de herramientas que permitan al usuario realizar acciones rápidas mediante la utilización de íconos gráficos.
- Existirá consistencia en la solución. Se utilizará una misma disposición de títulos, mensajes, botones y otros objetos en las diferentes interfaces gráficas.

Documentación de usuario

- Se deberá contar con un manual de usuario en el que se especifique claramente la forma de operación de la solución computacional, las características y restricciones que se deben tomar en cuenta para la adquisición de imágenes, y en general cualquier información que se considere relevante para el manejo correcto de la solución.
- La solución computacional deberá contar también con documentación de usuario disponible en línea, esto significa que el usuario dispondrá de información de ayuda mientras interactúe con la solución. En el caso que el usuario tenga alguna duda sobre la forma de funcionamiento de la solución o sobre las características de las imágenes o simplemente desee conocer algo más respecto a la solución computacional puede acudir a esta información de ayuda.

2.1.5.7.4 *Requerimientos del Entorno*

Dependencia

La solución computacional será implementada sobre el entorno de la herramienta de desarrollo (Matlab versión 7.0), por lo tanto tendrá dependencia con esta herramienta.

Seguridad

La solución computacional deberá restringir el acceso de los usuarios a las funciones de administración. En cuanto al resto de la funcionalidad cualquier usuario puede utilizarla.

Se definirán como funciones de administración a la actualización de claves de usuarios y a la generación del reporte de usuarios registrados. Los usuarios que pueden acceder a esta funcionalidad están definidos en los *Requerimientos de Usuario*.

2.2 ANÁLISIS

Una vez definido el concepto de la solución computacional se debe entender y analizar los distintos elementos que la componen para así alcanzar una visión clara de la arquitectura de la solución sin entrar en detalles de implementación.

Como primer paso se establecen las alternativas de diseño o configuración de la solución computacional partiendo del concepto de ésta (meta, objetivos, características, dependencias, etc.). Así se pretende encontrar una estrategia a seguir, desde el punto de vista conceptual, que sea adecuada para el contexto, requerimientos y restricciones existentes pero que a la vez permita alcanzar las metas y objetivos establecidos.

Cuando se ha identificado la mejor alternativa de diseño se pueden definir los elementos que formarán parte de la solución computacional, identificándolos, estableciendo relaciones entre ellos y asignándoles responsabilidades.

Esta primera visión de la arquitectura servirá como base para el posterior diseño e implementación de la solución computacional.

2.2.1 ALTERNATIVAS DE DISEÑO CONCEPTUAL DE LA SOLUCIÓN

Definir alternativas de diseño conceptual para la solución computacional basándose en trabajos previos descritos en la literatura permite revisar y entender métodos antes probados en aplicaciones orientadas al mismo campo, en este caso el reconocimiento automático de rostros.

Analizar métodos o técnicas propuestas en investigaciones pasadas tiene como ventaja enfocarse en aspectos que se pueden mejorar o cambiar en éstos y no enfocarse en los aspectos básicos que han sido extensamente probados.

Para el presente proyecto se realizó una *etapa experimental de reconocimiento*, la cual permitió analizar los métodos más adecuados para llevar a cabo el reconocimiento automático de rostros considerando las características previamente definidas para la solución computacional que se busca implementar, así como también los objetivos, metas, suposiciones, dependencias y otros elementos que componen el concepto de dicha solución. Finalmente se estableció la estrategia de diseño conceptual más acorde al contexto del presente proyecto y de la solución computacional en sí.

Los detalles de esta etapa experimental se encuentran descritos en el Anexo 11.

2.2.2 MODELO DE ANÁLISIS

Este modelo permite definir a nivel conceptual los elementos que formarán parte de la solución computacional y su interacción. Para esto los casos de uso se describen mediante clases de análisis, también se describe la interacción entre objetos de dichas clases a través de las realizaciones de los casos de uso.

2.2.2.1 Identificación de Clases de Análisis

Las clases de análisis son abstracciones que permiten refinar y modelar los escenarios descritos en la especificación de casos de uso, en términos de responsabilidades.

Una clase de análisis puede encajar dentro de uno de los estereotipos básicos:

- De interfaz
- De control
- De entidad

Las clases de interfaz se emplean para modelar interacciones entre los actores y la solución computacional, las clases de control modelan el control y coordinación de los casos de uso y las clases de entidad modelan información que generalmente es persistente.

En el Anexo 13 se puntualizan las clases de análisis identificadas por cada uno de los casos de uso que componen la solución computacional, aquellas clases que forman parte de los casos de uso de inclusión o de extensión (casos de uso abstractos) no se describen en otros casos de uso con el objetivo de evitar redundancia.

2.2.2.2 Realización de Casos de Uso - Análisis

Su objetivo es definir desde el punto de vista de los casos de uso cómo interactúan entre sí los objetos que pertenecen a las clases de análisis identificadas.

En el Anexo 14 se emplean diagramas de colaboración para describir los escenarios más representativos de los casos de uso concretos, en términos de clases de análisis y sus objetos. Los mensajes intercambiados entre los objetos no se asocian directamente con operaciones sino con responsabilidades.

2.2.3 IDENTIFICACIÓN DE PAQUETES DE ANÁLISIS

Los paquetes de análisis permiten organizar las clases de análisis en piezas más pequeñas y manejables.

A continuación, la Figura 2.3 muestra el diagrama de paquetes que organiza las clases de análisis identificadas durante esta etapa.

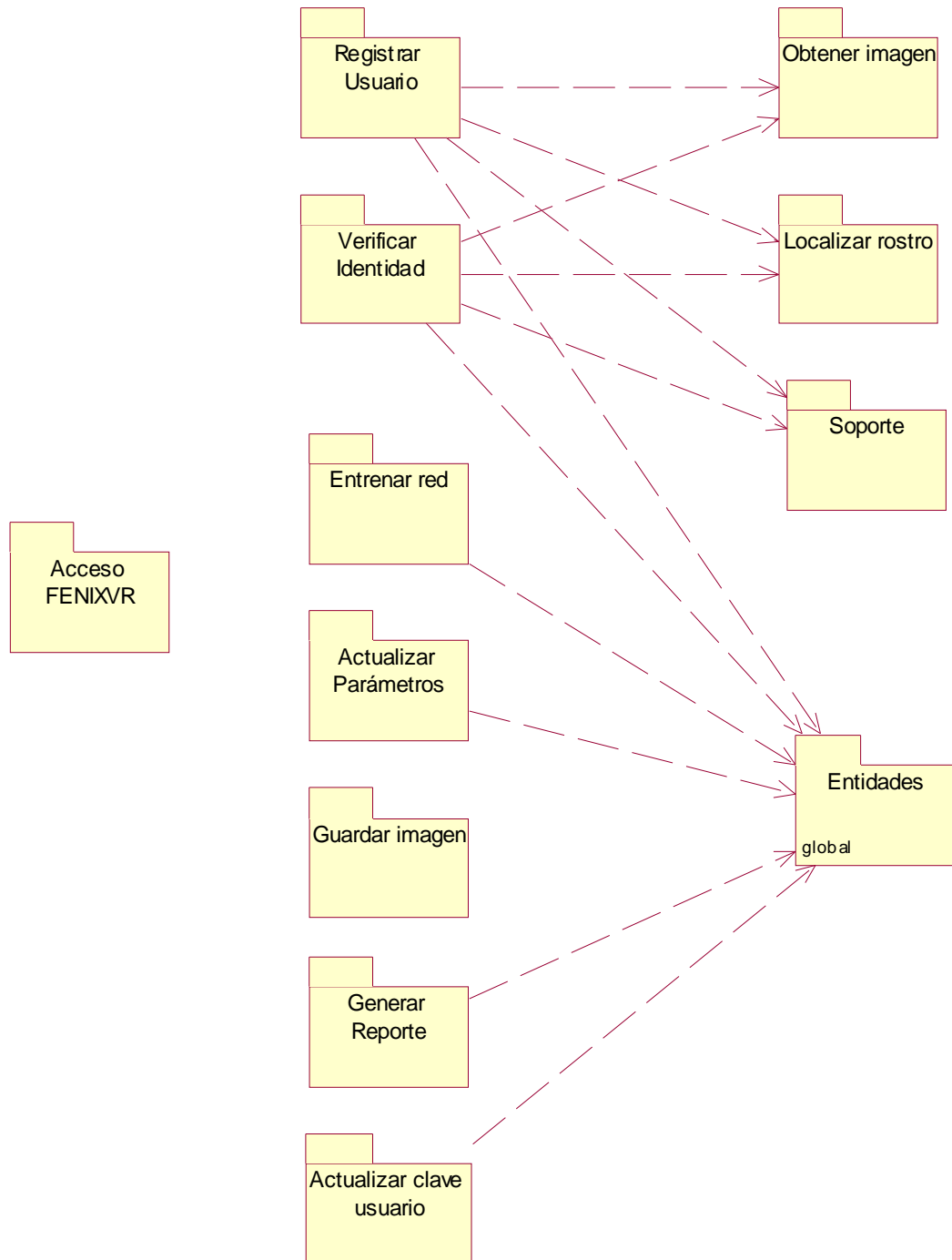


Figura 2.3. Diagrama de Paquetes de Análisis

2.3 DISEÑO DEL SISTEMA

Esta nueva visión de la arquitectura refina la visión proporcionada por el análisis y sirve como base para la implementación de la solución computacional.

2.3.1 MODELO DE DISEÑO

En esta etapa el modelo de clases que proviene del análisis se transforma en un modelo de clases de diseño, donde se detallan las distintas clases que formarán parte de la solución computacional en función de sus atributos y métodos, que en conjunto cubren las responsabilidades identificadas durante el análisis, también se detallan las relaciones establecidas entre éstas. A diferencia del modelo de análisis el modelo de diseño sí considera el entorno tecnológico y de desarrollo elegido para la implementación, por tal razón se considera que es más físico que el modelo de análisis.

2.3.1.1 Identificación de Clases de Diseño

En general cada clase obtenida durante la etapa de análisis es transformada a una o más clases de diseño, y para cada clase de diseño se definen de manera precisa sus atributos y métodos. Sin embargo debido a ciertos aspectos como el diseño de asociaciones y agregaciones entre clases, el entorno de implementación seleccionado o por cuestiones de optimización, algunas clases de análisis pueden desaparecer o combinarse e igualmente clases adicionales pueden ser requeridas.

A continuación se describen algunas de las clases de diseño que formarán parte de la funcionalidad básica de la solución computacional, en la descripción de cada clase no se incluyen métodos de creación ni métodos mutadores (denominados *gets* y *sets*) con el objetivo de simplificar el diseño, sintetizando así el contenido relevante de las

distintas clases y dejando aparte los elementos estándar. El Anexo 15 describe las clases de diseño restantes.

2.3.1.1.1 Identificación de Clases de Diseño – Interfaz

A continuación, en las Tablas 2.10 hasta 2.14, se detalla los atributos y operaciones de las clases de diseño de tipo interfaz.

CLASE : FENIXVR		
Refinamiento clase de análisis : IUOpciones		
ATRIBUTOS		
Nombre	Tipo	Descripción
---	---	---
MÉTODOS		
Nombre	Descripción	
+ btn_reg_usu_Callback() : void	Maneja el evento relacionado a la solicitud de registrar un nuevo usuario, se crea una instancia de la clase registrar_usuario_GUI.	
+ btn_ent_red_Callback() : void	Maneja el evento relacionado a la solicitud de entrenar una red neuronal, se crea una instancia de la clase ManejadorEntrenamientoRed.	
+ btn_verif_ident_Callback() : void	Maneja el evento relacionado a la solicitud de verificar la identidad de un usuario, se crea una instancia de la clase verif_ident_GUI.	
+btn_preferencias_Callback() : void	Maneja el evento relacionado a la solicitud de actualizar parámetros de la red neuronal, se crea una instancia de la clase preferencias_GUI.	
+ btn_salir_Callback() : void	Destruye el objeto.	
- busc_arch_fnc() : Integer	Verifica que existan los archivos referentes al Eigen-Espacio, la red neuronal y la estructura de claves de los usuarios; archivos requeridos para la verificación de identidad.	
- comp_usu_fnc (obj_DB : AccesoBD) : Integer	Verifica que existan usuarios registrados, esto se requiere como paso previo para el entrenamiento de la red neuronal.	

Tabla 2.10. Descripción de la clase de diseño *FENIXVR*

CLASE : registrar_usuario_GUI		
Refinamiento clase de análisis : IURegistrarUsuario		
ATRIBUTOS		
Nombre	Tipo	Descripción
- txt_nomb_img	String	Cuadro de texto para el nombre de la imagen obtenida.
- txt_tam_img	String	Cuadro de texto para las dimensiones de la imagen.
- txt_tiemp_detect	String	Cuadro de texto para mostrar el tiempo empleado para la localización del rostro.
- ax_img_orig	Matriz	Eje para la imagen obtenida (imagen existente o capturada).
- ax_rostro_detect	Matriz	Eje para la imagen del rostro detectado.
- ax_rostro_1 .. ax_rostro_5	Matriz	Ejes para las imágenes en escala de grises del área del rostro del nuevo usuario.
- txt_nomb	String	Cuadro de texto para el nombre del nuevo usuario.
- txt_apel	String	Cuadro de texto para el apellido del nuevo usuario.
- txt_clav	String	Cuadro de texto para la clave del nuevo usuario.
- txt_genero	String	Combo para seleccionar el género.
MÉTODOS		
Nombre	Descripción	
+ btn_analizar_Callback() : void	Maneja el evento relacionado a la solicitud de localizar un rostro en la imagen obtenida y pide los servicios del subsistema <i>Localizar Rostro</i> a través de la interfaz <i>ManejadorLocalizacionRostro</i> .	
+ btn_reg_usu_Callback() : void	Maneja el evento relacionado a la solicitud de almacenar la información del nuevo usuario.	
+ carpeta_Callback() : void	Maneja el evento relacionado a la solicitud de obtener una imagen desde cualquier unidad de almacenamiento.	
+ video_Callback() : void	Maneja el evento relacionado a la solicitud de obtener una imagen mediante una cámara digital.	
+ guard_img_orig_Callback() : void	Maneja el evento relacionado a la solicitud de guardar una copia de la imagen obtenida.	
+ guard_rost_detect_Callback() : void	Maneja el evento relacionado a la solicitud de guardar una copia de la imagen con el rostro localizado.	

+ btn_verif_Callback() : void	Maneja el evento relacionado a la solicitud de verificar una identidad empleando la imagen de un rostro y una clave de usuario.
+ carpeta_Callback() : void	Maneja el evento relacionado a la solicitud de obtener una imagen desde cualquier unidad de almacenamiento.
+ video_Callback() : void	Maneja el evento relacionado a la solicitud de obtener una imagen mediante una cámara digital.
+ guard_img_orig_Callback() : void	Maneja el evento relacionado a la solicitud de guardar una copia de la imagen obtenida.
+ guard_rost_detect_Callback() : void	Maneja el evento relacionado a la solicitud de guardar una copia de la imagen con el rostro localizado.
+ btn_salir_Callback() : void	Destruye el objeto.
- most_img_obt_fnc (imgX_img : mx) : void	Presenta la imagen obtenida (capturada o seleccionada por el usuario), la misma que recibe como parámetro de entrada.
- most_img_rec_fnc (img_recons_img : mx) : void	Presenta la imagen reconstruida en base a componentes principales, dicha imagen la recibe como parámetro de entrada.

Tabla 2.12. Descripción de la clase de diseño *verif_ident_GUI*

CLASE : most_img_usu_GUI		
Refinamiento clase de análisis : IUInfUsuario		
ATRIBUTOS		
Nombre	Tipo	Descripción
- txt_nomb	String	Cuadro de texto para el nombre del usuario autenticado.
- txt_apel	String	Cuadro de texto para el apellido del usuario autenticado.
- txt_clav	String	Cuadro de texto para la clave del usuario autenticado.
- txt_genero	String	Combo para género.
- ax_rostro_1 .. ax_rostro_5	Matriz	Ejes para las imágenes en escala de grises del rostro del usuario autenticado.
MÉTODOS		
Nombre	Descripción	
+ btn_salir_Callback() : void	Destruye el objeto.	

+ disp_id_i	Integer	Identificador del dispositivo de captura detectado.
- nomb_interf_disp_txt	String	Nombre de la interfaz soportada. Valor: 'winvideo'
- disp_inf_st	struct	Información del dispositivo de captura detectado.
MÉTODOS		
Nombre	Descripción	
+ detect_disp_fnc (obj : Object) : Object	Detecta si existe un dispositivo de captura que se encuentre accesible a través de una interfaz genérica de Windows para dispositivos de video (dispositivos que utilicen controladores WDM o VFW).	

Tabla 2.16. Descripción de la clase de diseño *ManejadorDetDisp*

CLASE : LocalizacionRostro		
Refinamiento clase de análisis : ManejadorLocalizacionRostro		
ATRIBUTOS		
Nombre	Tipo	Descripción
- FIL_i	Integer	Número de filas de la imagen original.
- COL_i	Integer	Número de columnas de la imagen original.
- tiemp_db	Double	Tiempo empleado para la localización del rostro (tiempo en segundos).
- obj_extrac_norm_obj	Objeto	Instancia de la clase ExtraccionNormalizacion.
- band_err_i	Integer	Valor que indica si ocurrió un error en la localización del rostro.
MÉTODOS		
Nombre	Descripción	
+ detect_rostro_fnc (obj : Object, img_orig_img : mx, manejador_hdl : hdl) : Object	Gestiona la localización de un rostro en la imagen que recibe como parámetro de entrada.	
+ mst_rostro_fnc (prop_ojos_st : st, prop_boca_st : st, cen1_db : Double, cen2_db : Double, cen3_db : Double, X_min_i : Integer, COL_i : Integer, manejador_hdl : hdl) : void	Muestra el área del rostro, incluyendo la posición aproximada de los ojos y la boca en la imagen original.	

Tabla 2.17. Descripción de la clase de diseño *LocalizacionRostro*

CLASE : PatronImagen		
Refinamiento clase de análisis : PatronImagen		
ATRIBUTOS		
Nombre	Tipo	Descripción
- peso_vc	Vector	Vector de pesos que resulta de proyectar una imagen en el espacio de rostros.
- rostro_recons_img	Matriz	Imagen reconstruida en base a componentes principales (PCA).
MÉTODOS		
Nombre	Descripción	
+ calc_img_recons_fnc (obj : Object, rostro_img : mx, obj_ImgRec : ImagenReconstruida) : Object	Calcula la proyección de una imagen sobre el espacio de rostros y reconstruye dicha imagen a partir de los pesos calculados. Utiliza la imagen que recibe como parámetro de entrada (imagen de un rostro). El vector de pesos calculado se utiliza posteriormente para la simulación de la red neuronal.	

Tabla 2.18. Descripción de la clase de diseño *PatronImagen*

CLASE : ManejadorVerificacion		
Refinamiento clase de análisis : ManejadorVerificacion		
ATRIBUTOS		
Nombre	Tipo	Descripción
- punt_db	Double	Salida de la red neuronal con el mayor valor.
- clave_txt	String	Clave de usuario a la que referencia la salida con mayor valor.
- val_comparacion_db	Double	Valor empleado como referencia para decidir si la salida de la red neuronal con mayor valor puede ser aceptada para autenticar o no a un usuario. La salida con mayor puntuación debe ser mayor ≥ 0.5
- band_verif_i	Boolean	Valor empleado para la verificación de identidad. 1 (verdadero) = Usuario autenticado 0 (falso) = Usuario no autenticado.

MÉTODOS	
Nombre	Descripción
+ verif_red_fnc (obj : Object, peso_vc : vc) : Object	Carga las estructuras que contienen las claves de los usuarios registrados y la red neuronal entrenada. Realiza la simulación de la red empleando el vector de pesos que recibe como parámetro de entrada.

Tabla 2.19. Descripción de la clase de diseño *ManejadorVerificacion*

CLASE : ManejadorPresentacionInfUsu		
Refinamiento clase de análisis : ManejadorPresentacionInfUsu		
ATRIBUTOS		
Nombre	Tipo	Descripción
- num_img_i	Integer	Número de imágenes almacenadas del usuario previamente autenticado.
- dat_usu_curs	cursor	Cursor que contiene los datos del usuario autenticado (nombre, apellido, clave y género).
- img_usu_curs	cursor	Cursor que contiene las imágenes almacenadas del usuario autenticado
MÉTODOS		
Nombre	Descripción	
+ obt_inf_usu_fnc (obj : Object, clave_txt : String, obj_BD : AccesoBD) : Object	Obtiene la información almacenada (datos e imágenes) que pertenece al usuario previamente autenticado.	

Tabla 2.20. Descripción de la clase de diseño *ManejadorPresentacionInfUsu*

CLASE : ManejadorEntrenamientoRed		
Refinamiento clase de análisis : ManejadorEntrenamientoRed		
ATRIBUTOS		
Nombre	Tipo	Descripción
---	---	---

MÉTODOS	
Nombre	Descripción
+ ent_red_fnc (obj_DB : AccesoBD, obj_ImgRec : ImagenReconstruida) : Integer	Gestiona el proceso de entrenamiento de la red neuronal a través de métodos privados.
-pca_fnc (obj_DB : AccesoBD, obj_ImgRec : ImagenReconstruida) : st	Crea el espacio de rostros a partir de las imágenes almacenadas de los usuarios registrados. Calcula el conjunto de pesos asociado a cada usuario proyectando sus imágenes en el espacio de rostros. Retorna una estructura que contiene el espacio de rostros, el rostro promedio, el conjunto de pesos calculados para cada usuario, las claves asociadas a los usuarios y la matriz de salida empleada para el entrenamiento supervisado de la red neuronal.
-rbf_fnc(obj_DB : AccesoBD) : net	Crea y entrena una red neuronal de tipo RBF empleando los pesos calculados.
-guard_red_fnc (eigenespacio_mx : mx, rostroprom_mx : mx, red_rd : net, clav_st : st) : Integer	Almacena en disco la red neuronal previamente creada y entrenada así como también toda la información requerida para la verificación de identidad (espacio de rostros, rostro promedio y claves asociadas a los usuarios registrados).

Tabla 2.21. Descripción de la clase de diseño *ManejadorEntrenamientoRed*

CLASE : ManejadorParametros		
Refinamiento clase de análisis : ManejadorParam y ManejadorActualizacionParam		
ATRIBUTOS		
Nombre	Tipo	Descripción
- num_neur_i	Integer	Número de neuronas añadidas por desplazamiento. Valor por defecto: 2
- val_goal_db	Double	Error mínimo cuadrático estándar. Valor por defecto: 0.8

MÉTODOS	
Nombre	Descripción
+ obt_val_param_fnc (obj : Object, obj_DB : AccesoBD) : Object	Obtiene los valores actuales para los parámetros de la red que se encuentran almacenados.
+ act_val_param_fnc (obj : Object, obj_DB : AccesoBD) : void	Actualiza los valores de los parámetros de la red.

Tabla 2.22. Descripción de la clase de diseño *ManejadorParametros*

2.3.1.1.3 Identificación de Clases de Diseño – Entidad

A continuación, en las Tablas 2.23 hasta 2.29, se detallan los atributos y operaciones de las clases de diseño de tipo entidad.

CLASE : USUARIO		
Refinamiento clase de análisis : Usuario		
ATRIBUTOS		
Nombre	Tipo	Descripción
+ Clave	String	Clave del usuario. Combinación numérica de máximo 10 dígitos.
+ Nombres	String	Nombres del usuario. Cadena de caracteres (longitud máxima 50)
+ Apellidos	String	Apellidos del usuario. Cadena de caracteres (longitud máxima 50)
+ Genero	String	Género.

Tabla 2.23. Descripción de la clase de diseño *Usuario*

CLASE : IMAGEN		
Refinamiento clase de análisis : Imagen		
ATRIBUTOS		
Nombre	Tipo	Descripción
+ Imagen	OLE	Imagen en escala de grises del rostro de un usuario.

Tabla 2.24. Descripción de la clase de diseño *Imagen*

CLASE : SUBESPACIO		
Refinamiento clase de análisis : Subespacio		
ATRIBUTOS		
Nombre	Tipo	Descripción
+ Imagen_sintet	OLE	Imagen sintética en escala de grises.

Tabla 2.25. Descripción de la clase de diseño *Subespacio*

CLASE : PREFERENCIAS		
Refinamiento clase de análisis : Parametro		
ATRIBUTOS		
Nombre	Tipo	Descripción
+ num_neur	Integer	Número de neuronas añadidas por desplazamiento.
+ val_goal	Double	Error mínimo cuadrático estándar.

Tabla 2.26. Descripción de la clase de diseño *Preferencias*

CLASE : RED_RD		
Refinamiento clase de análisis : RedNeuronal		
ATRIBUTOS (*)		
Nombre	Tipo	Descripción
- numInputs		Número de entradas de la red neuronal.
- numLayers		Número de capas de la red neuronal.
- numOutputs		Número de salidas de la red neuronal.
- biasConnect		Conexiones bias.
- inputConnect		Conexiones de entrada.
- layerConnect		Conexiones entre capas.
- outputConnect		Conexiones de salida.
- targetConnect		Conexiones de salidas deseadas.
MÉTODOS (*)		
Nombre	Descripción	
- adaptFcn()	Define los algoritmos a usar para adaptar, inicializar o entrenar a la red neuronal.	

- initFcn()	Define la función utilizada para inicializar los pesos de la red y los vectores bias.
- performFcn()	Define la función utilizada para medir el desempeño de la red neuronal.
- trainFcn()	Define la función utilizada para entrenar a la red neuronal.
(*) La herramienta de desarrollo seleccionada ofrece un conjunto de objetos y operaciones para el manejo de redes neuronales artificiales. Esta herramienta tiene estructuras predefinidas (atributos y métodos) para este tipo de objetos. Los atributos y métodos descritos forman parte de dicha estructura.	

Tabla 2.27. Descripción de la clase de diseño *red_rd*

CLASE : EigenEspacioFx		
Refinamiento clase de análisis : EspacioRostros		
ATRIBUTOS		
Nombre	Tipo	Descripción
+ eigenvectores	Matriz	Conjunto de vectores que forman el espacio de rostros conocidos (eigenespacio).

Tabla 2.28. Descripción de la clase de diseño *EigenEspacioFx*

CLASE : RostroPromFx		
Refinamiento clase de análisis : RostroProm		
ATRIBUTOS		
Nombre	Tipo	Descripción
+ rostro_prom_vc	Vector	Vector que representa al rostro promedio empleado para la proyección de imágenes en el espacio de rostros.

Tabla 2.29. Descripción de la clase de diseño *RostroPromFx*

* Durante la etapa de diseño desaparece la clase de análisis ContadorUsuarios (clase de tipo control).

2.3.1.1.4 Identificación de Clases de Diseño – Nuevas Clases

A continuación, las Tablas 2.30 y 2.31 detallan los atributos y operaciones de las clases identificadas en la etapa de diseño.

CLASE :			ImagenReconstruida
ATRIBUTOS			
Nombre	Tipo	Descripción	
---	---	---	
MÉTODOS			
Nombre		Descripción	
+ proy_img_fnc (img_rostro_img : mx) : vc		Proyecta en el espacio de rostros la imagen que recibe como parámetro de entrada y retorna el vector de pesos que resulta de la proyección.	
+ reconst_img_fnc (imagen_bin : blob) : mx		Reconstruye una imagen a partir del objeto BLOB que recibe como parámetro.	

Tabla 2.30. Descripción de la clase de diseño *ImagenReconstruida*

CLASE :			AccesoBD
ATRIBUTOS			
Nombre	Tipo	Descripción	
- nomb_ODBC_txt	String	Nombre del origen de datos ODBC. Valor predefinido = 'conexODBC'	
- conn	Objeto	Objeto de conexión a la base de datos especificada a través de un ODBC.	
MÉTODOS			
Nombre		Descripción	
+ conexion(obj : Object) : Object		Establece una conexión con la base de datos.	
+ insert_usu_sp (Clave : String, Nombre : String, Apellido : String, Genero : String)		Guarda los datos del usuario.	
+ insert_img_sp (Clave : String, img_vc : vc)		Guarda las imágenes del rostro del usuario.	

+ insert_subesp_sp (Clave : String, imgv1_img : vc)	Guarda las imágenes sintéticas del usuario.
+ elim_usu_sp (clav_txt : String)	Elimina toda la información de un usuario cuya clave es proporcionada como parámetro de entrada.
+ busqueda_clav_sp (clav_txt : String)	Consulta si existe un usuario registrado con la clave proporcionada como parámetro de entrada.
+ select_dat_usu_sp (clave_txt : String)	Consulta los datos de un usuario cuya clave es proporcionada como parámetro de entrada.
+ select_img1_usu_sp (clave_txt : String)	Selecciona una imagen de un usuario cuya clave es proporcionada como parámetro de entrada.
+ select_imgs_usu_sp (clave_txt : String)	Selecciona todas las imágenes de un usuario cuya clave es proporcionada como parámetro de entrada.
+ consultar_num_usus_sp()	Consulta el número de usuarios registrados en la base de datos.
+ select_subesps_sp()	Selecciona todas las imágenes sintéticas almacenadas en la base de datos.
+ select_preferencias_sp()	Consulta los valores de los parámetros que pueden ser modificados para la red neuronal.
+ insert_preferencias_sp (datos_st : st)	Guarda los valores de los parámetros que pueden ser modificados para la red neuronal.
+ actualizar_preferencias_sp (datos_st : st)	Actualiza los valores de los parámetros que pueden ser modificados para la red neuronal.
+ consultar_usuarios_sp()	Consulta el nombre y apellido de todos los usuarios registrados.
+ actualizar_clav_sp (nombre_txt : String, apel_txt : String, clav_txt : String, nuev_clav_txt : String)	Actualiza la clave de un usuario.
+ reporte_sp()	Consulta la información para generar el reporte de usuarios registrados.
+ cerrar_conex(obj : Object)	Destruye el objeto creado por el método conexion().
Esta clase se crea debido a que el SGBD seleccionado (Microsoft Access) no soporta el uso de procedimientos almacenados.	

Tabla 2.31. Descripción de la clase de diseño *AccesoBD*

2.3.1.2 Realización de Casos de Uso – Diseño

La realización de casos de uso – diseño describe cómo los casos de uso se ejecutan en términos de clases de diseño y sus objetos. Cada realización de un caso de uso – diseño tiene una traza directa a la correspondiente realización del caso de uso – análisis.

A continuación se emplean diagramas de secuencia para describir los escenarios más representativos de algunos casos de uso concretos, donde se muestra cómo se vinculan los objetos de las clases de diseño; adicionalmente se complementan los diagramas con una descripción textual. El Anexo 16 muestra los diagramas de secuencia para los casos de uso restantes.

2.3.1.2.1 Diagramas de Secuencia – Diseño: Caso de uso Registrar Usuario

- *Flujo Básico – Emplear una imagen existente*

Véase Figura 2.4.

Descripción del flujo de sucesos-diseño:

La secuencia se inicia cuando el Operador solicita a un objeto de la clase FENIXVR registrar un nuevo usuario al presionar el botón *Registrar Usuario*. Esto genera un evento que es atendido por FENIXVR a través de `btn_reg_usu_Callback()` que activa y despliega la ventana de registro de usuarios (`registrar_usuario_GUI`).

Para iniciar el proceso de registro el Operador selecciona dentro de `registrar_usuario_GUI` la opción correspondiente al escenario que se describe (*Obtener Imagen - Carpeta*), esto genera un evento que es manejado por `carpeta_Callback()` que a su vez se encarga de solicitar un objeto de la clase `ImagenExistente`, el cual ofrece la operación `selec_img_fnc()` que permite al Operador seleccionar la imagen que desea utilizar mediante un cuadro de diálogo estándar para recuperar

archivos. El Operador selecciona una imagen y presiona el botón *Aceptar* del cuadro de diálogo, esto invoca a la operación `obt_img_selec_fnc()` proporcionada por `ImagenExistente` para cargar en memoria la imagen especificada en el cuadro de diálogo.

A continuación `registrar_usuario_GUI` obtiene de `ImagenExistente` la imagen que el Operador seleccionó y emplea la operación `most_img_obt_fnc()` con el fin de desplegar en pantalla dicha imagen, finalmente activa el botón *Analizar* que permite solicitar la localización de un rostro en la imagen. Una vez que se ha presentado dentro de la ventana `registrar_usuario_GUI` la imagen que el Operador seleccionó, éste solicita la localización de un rostro en dicha imagen al presionar el botón *Analizar* que genera un evento que es atendido por `registrar_usuario_GUI` a través de `btn_analizar_Callback()` que se encarga de comprobar que el número de imágenes del nuevo usuario que ya se han agregado para el registro es menor a cinco, en cuyo caso solicita los servicios del subsistema *Localizar Rostro* a través de la interfaz `ManejadorLocalizacionRostro` (`detect_rostro_fnc()`).

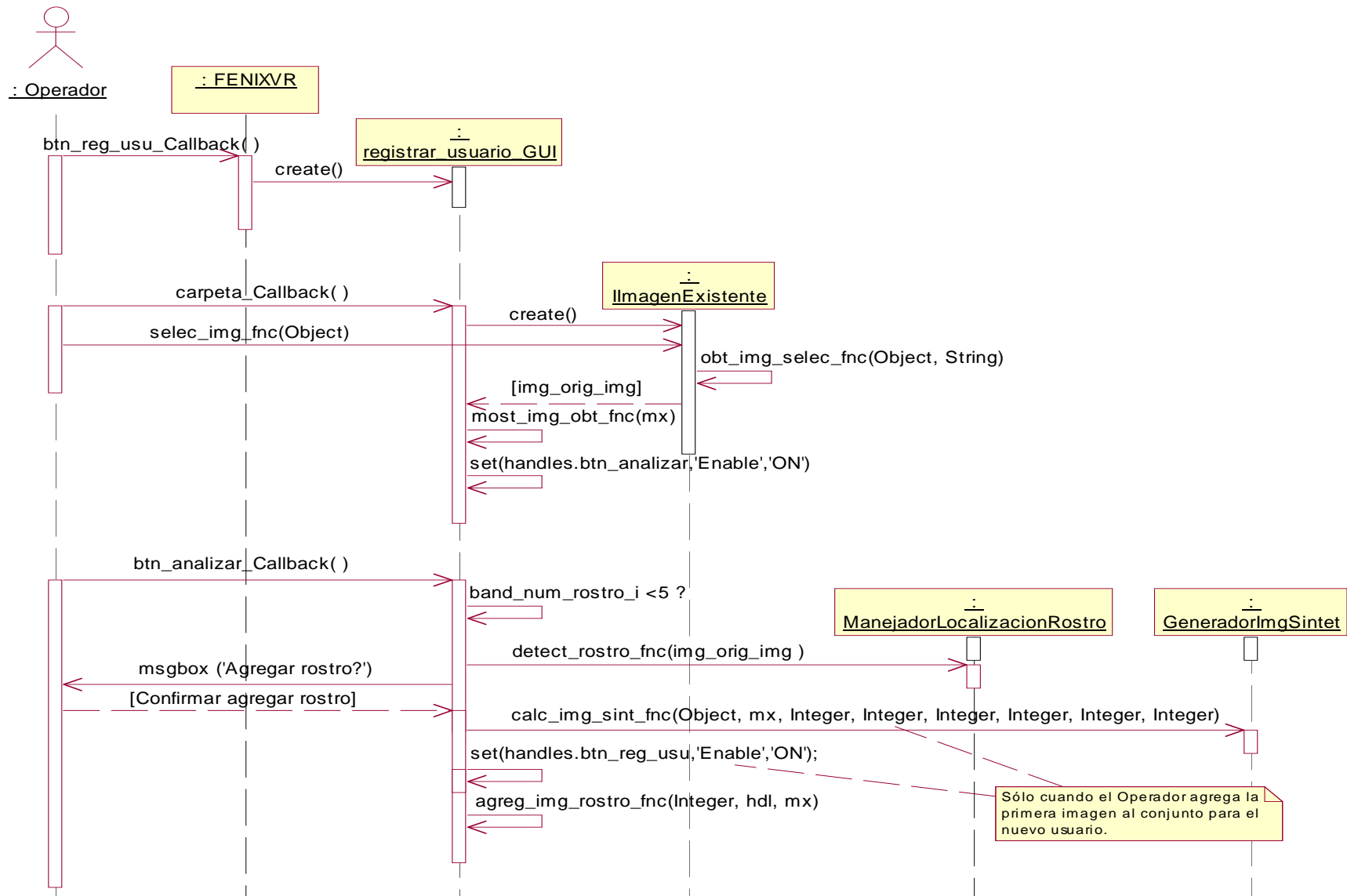
Luego que el subsistema *Localizar Rostro* ha encontrado un rostro en la imagen y ha presentado esta nueva imagen (con el área del rostro señalada) dentro de la ventana `registrar_usuario_GUI`, se solicita al Operador confirmar si desea agregar el rostro localizado al registro del nuevo usuario.

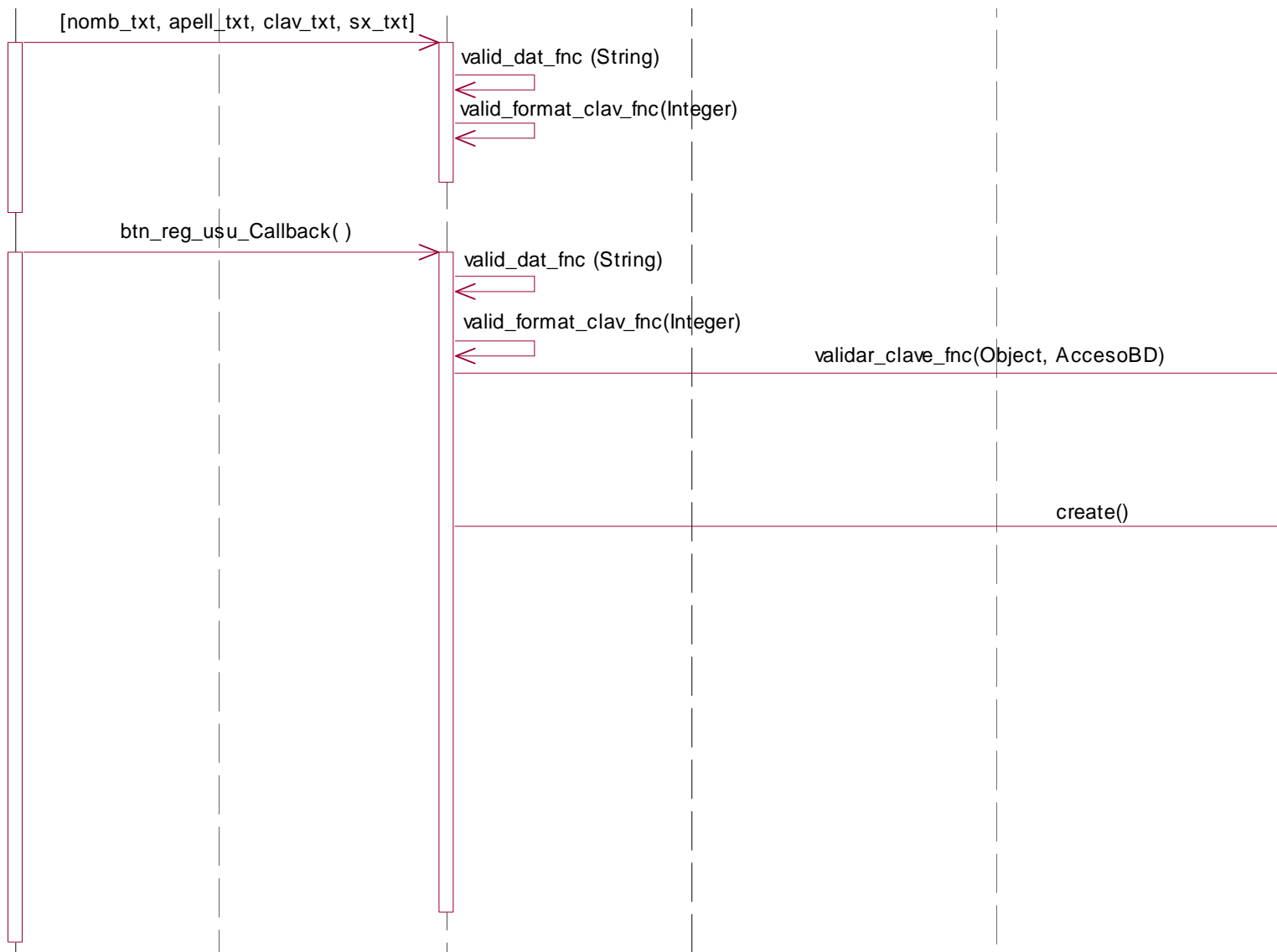
Si el Operador confirma que desea agregar la imagen del rostro localizado al registro, `registrar_usuario_GUI` verifica si ésta es la primera imagen agregada, en cuyo caso invoca la operación `calc_img_sint_fnc()` ofrecida por `GeneradorImgSintet` para crear seis imágenes sintéticas a partir del rostro localizado, que serán utilizadas en la etapa de entrenamiento de la red neuronal; adicionalmente activa el botón *Registrar Usuario* que permite al Operador solicitar guardar la información del nuevo usuario para así completar el proceso.

Toda imagen que sea agregada al registro del nuevo usuario es presentada dentro de la ventana registrar_usuario_GUI por medio de la operación agreg_img_rostro_fnc().

Luego de esto el Operador ingresa el nombre, apellido, clave y género del nuevo usuario, al ingresar cada uno de estos datos registrar_usuario_GUI emplea las operaciones valid_dat_fnc() y valid_format_clav_fnc() para validar el formato de dichos datos. Después el Operador solicita completar el proceso de registro al presionar el botón *Registrar Usuario* que genera un evento que es atendido por registrar_usuario_GUI a través de btn_reg_usu_Callback() que a su vez utiliza las operaciones valid_dat_fnc() y valid_format_clav_fnc() para validar nuevamente el formato de los datos ingresados, y la operación validar_clave_fnc() ofrecida por BuscadorUsuario, para controlar que no se trate de ingresar claves de usuario duplicadas dentro de la base de datos. Esta operación emplea a busqueda_clav_sp() que es proporcionada por AccesoBD para que acceda a la tabla Usuario en la base de datos y consulte si la clave ingresada por el Operador pertenece a un usuario ya registrado.

Finalmente, si la clave ingresada no pertenece a ningún usuario registrado registrar_usuario_GUI solicita un objeto de la clase ManejadorRegistro que se encarga de normalizar todas las imágenes del rostro del nuevo usuario a través de la operación norm_imgs_fnc() y luego almacena toda la información mediante la operación reg_usu_fnc() que a su vez invoca las operaciones insert_usu_sp(), insert_img_sp() e insert_subesp_sp() que son proporcionadas por AccesoBD y que acceden a las tablas Usuario, Imagen y Subespacio respectivamente para insertar la información del nuevo usuario, completando así el proceso de registro.





: Usuario

: Imagen

: Subespacio



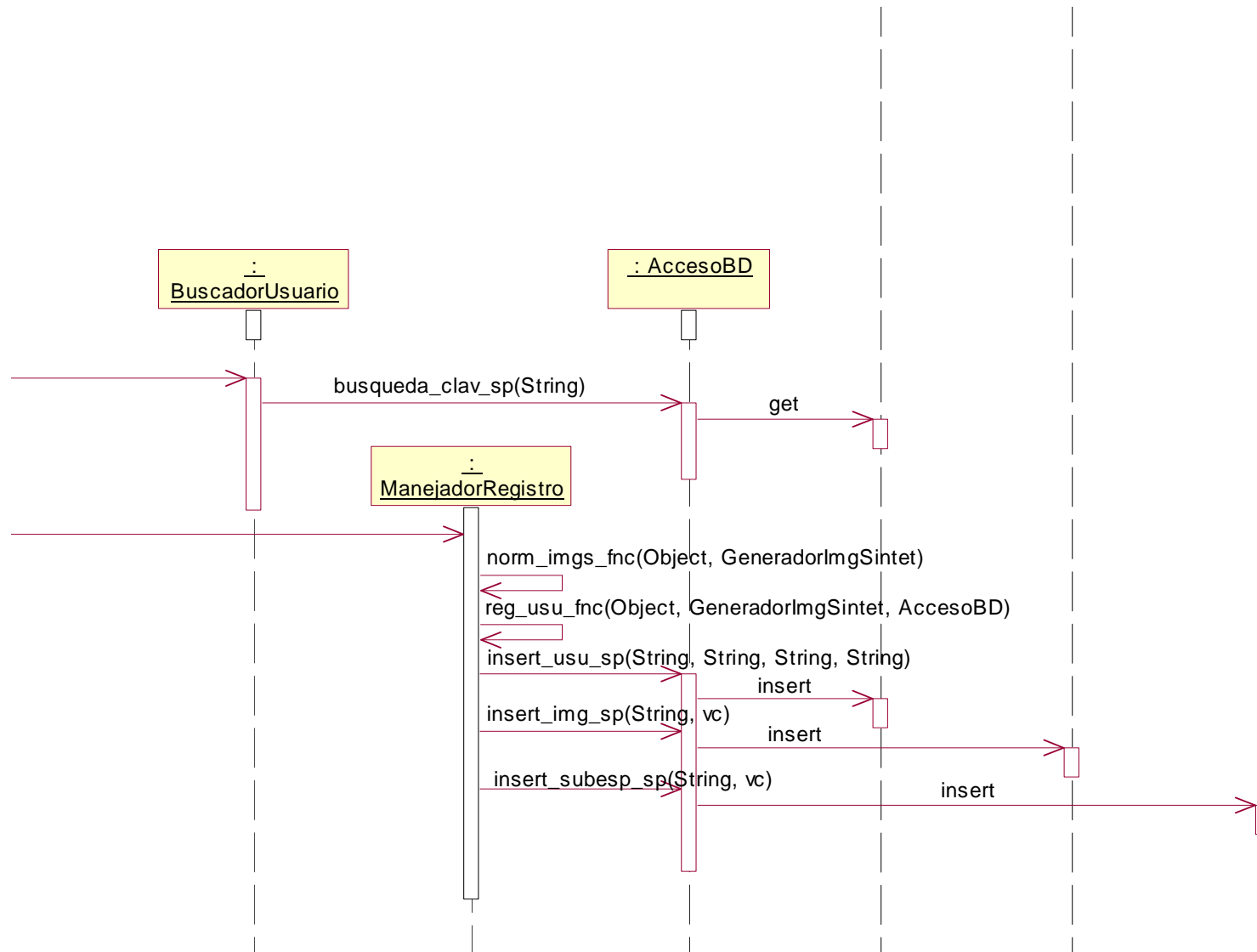


Figura 2.4. Diagrama de Secuencia - Diseño: Caso de uso *Registrar Usuario* (Flujo Básico - Imagen existente)

- *Flujo Básico – Capturar imagen*

Véase Figura 2.5.

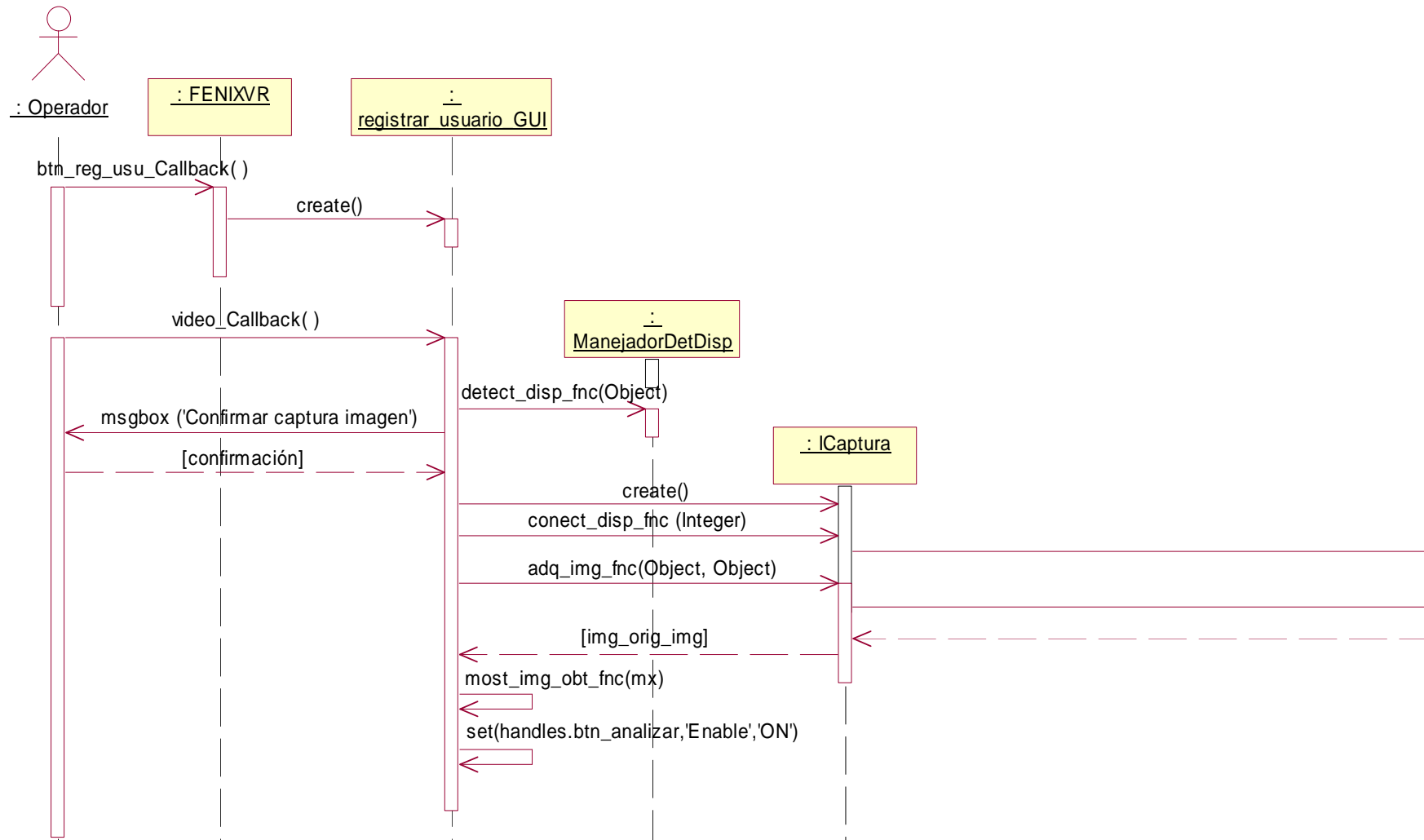
Descripción del flujo de sucesos-diseño:

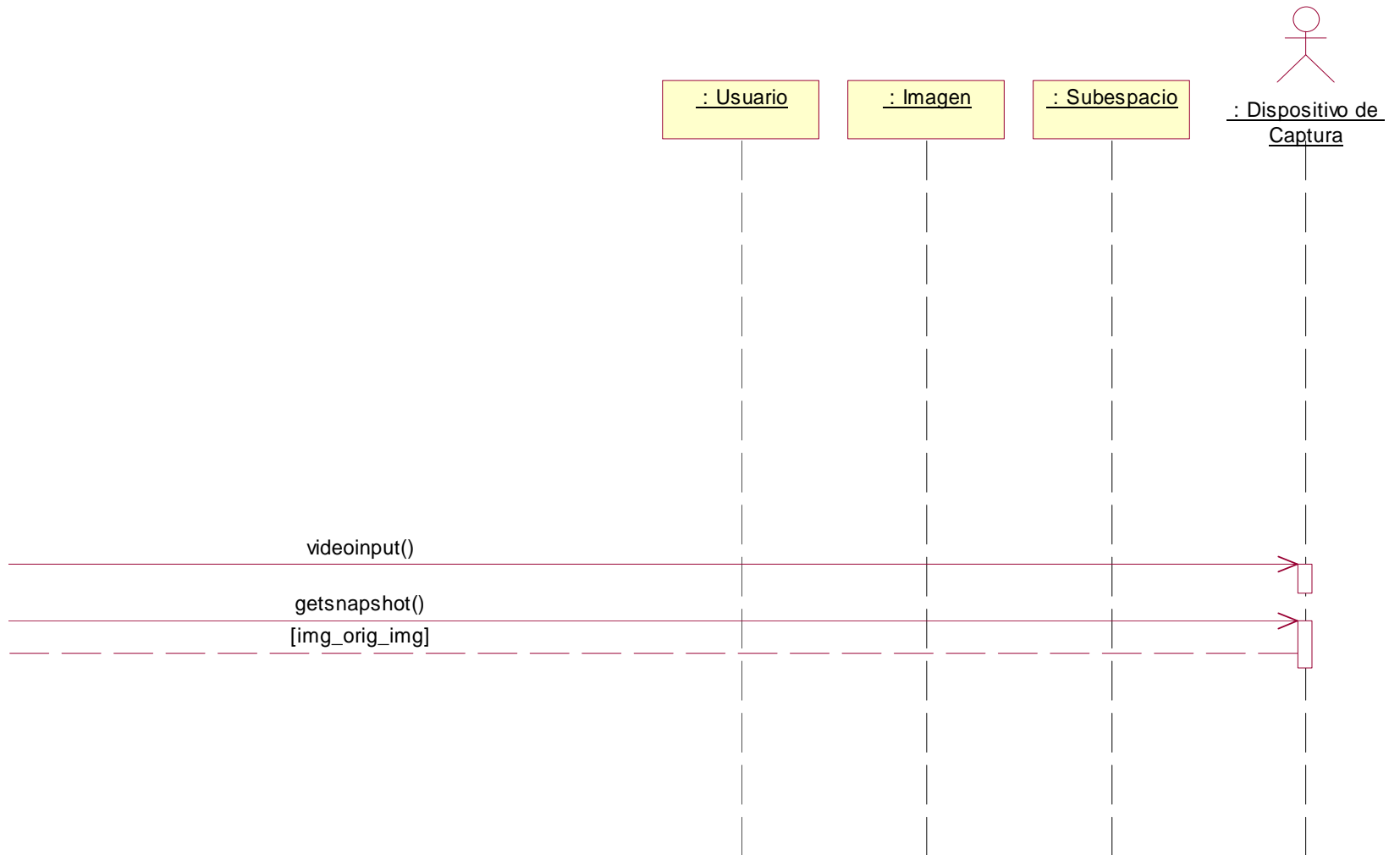
La secuencia se inicia cuando el Operador solicita a un objeto de la clase FENIXVR registrar un nuevo usuario al presionar el botón *Registrar Usuario*. Esto genera un evento que es atendido por FENIXVR a través de `btn_reg_usu_Callback()` que activa y despliega la ventana de registro de usuarios (`registrar_usuario_GUI`).

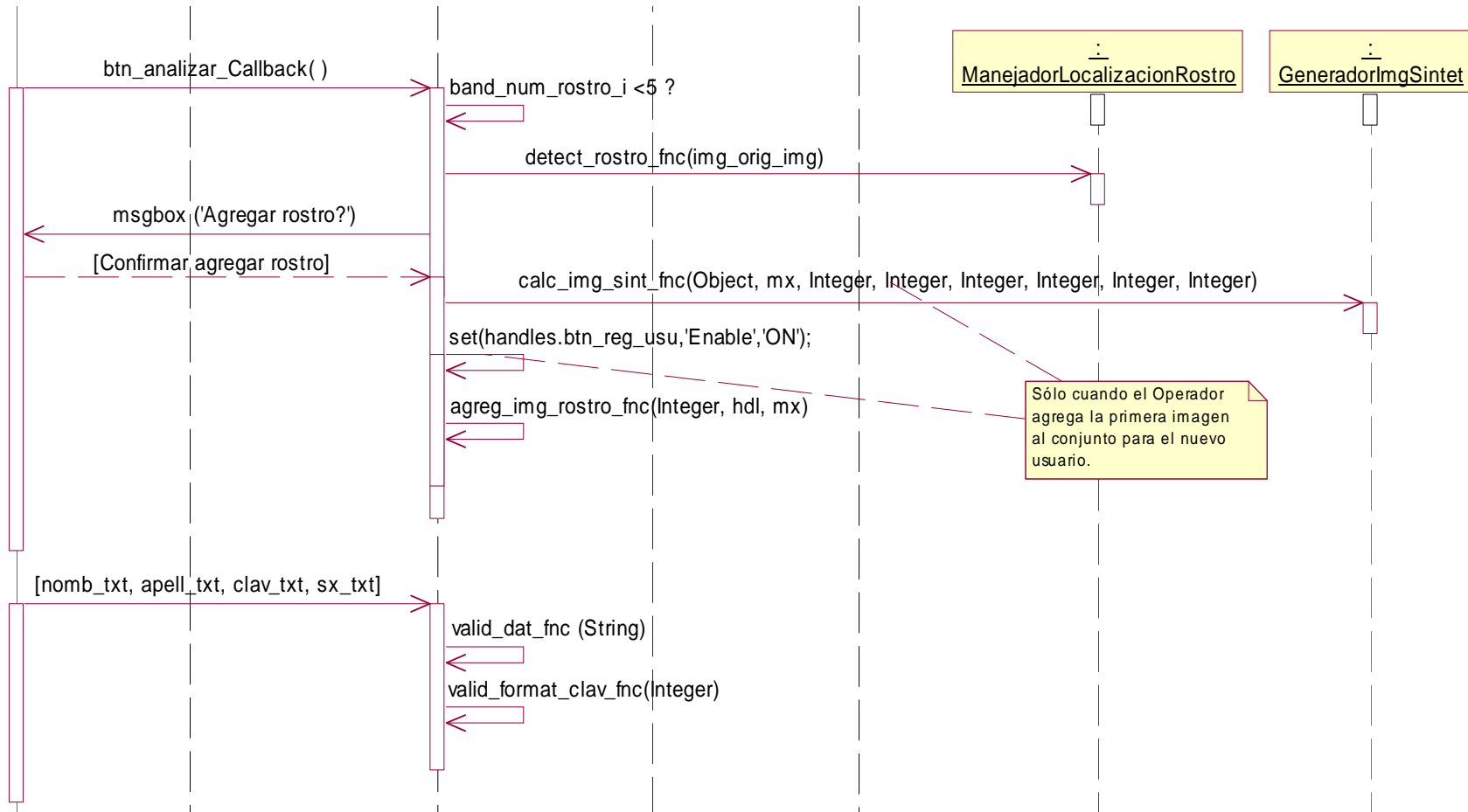
Para iniciar el proceso de registro el Operador selecciona dentro de `registrar_usuario_GUI` la opción correspondiente al escenario que se describe (*Obtener Imagen - Video*), esto genera un evento que es manejado por `video_Callback()` que emplea la operación `detect_disp_fnc()` proporcionada por `ManejadorDetDisp` para detectar si existe un dispositivo de captura que se encuentre accesible a través de una interfaz genérica de Windows para dispositivos de video (dispositivos que utilicen controladores WDM o VFW). Si se ha detectado un dispositivo que cumpla estas condiciones se solicita al Operador confirmar si desea capturar una imagen con ese dispositivo.

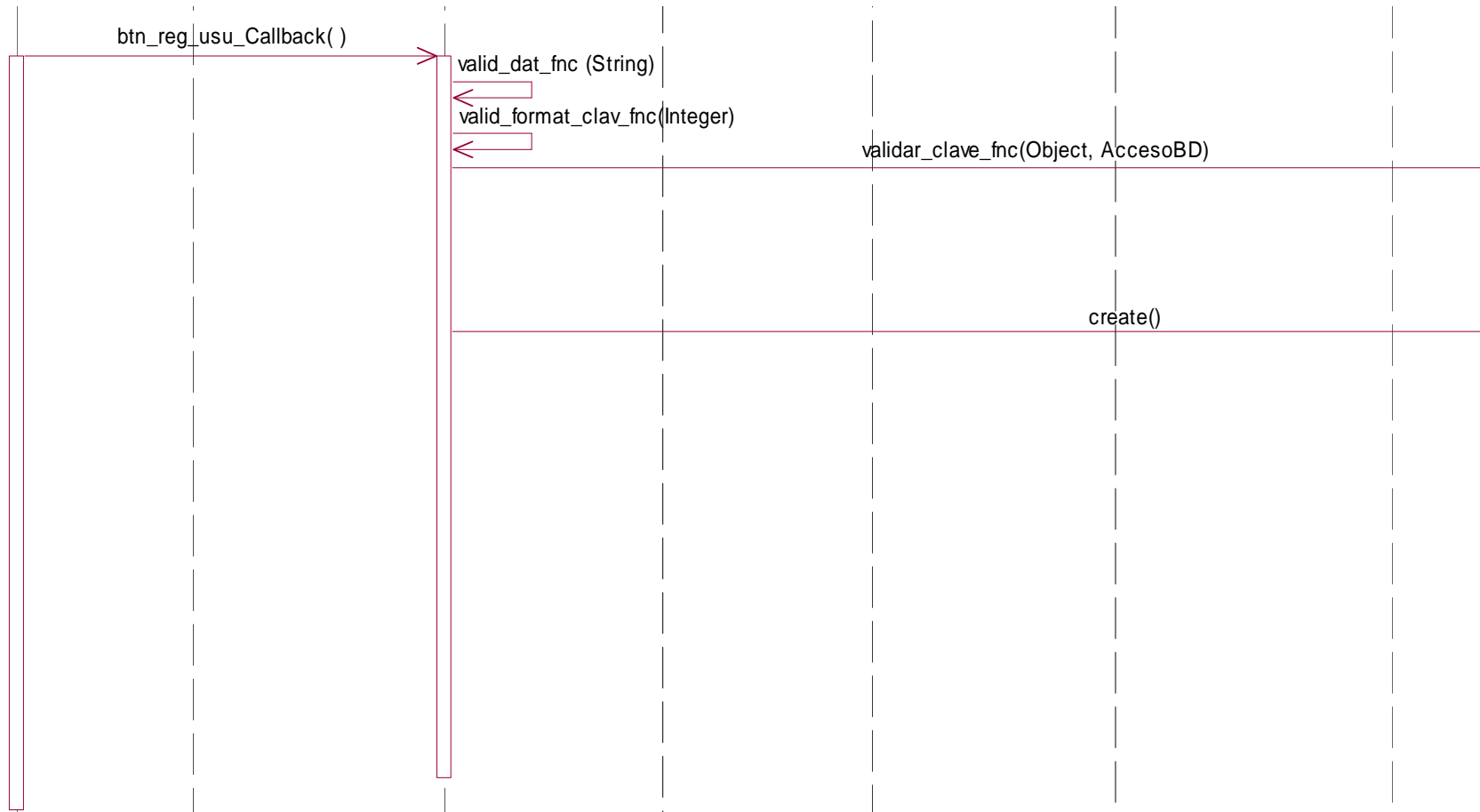
Si el Operador confirma el uso del dispositivo, `registrar_usuario_GUI` solicita un objeto de la clase `ICaptura` e invoca a las operaciones que éste ofrece: `conect_disp_fnc()` para que se encargue de establecer la conexión con el dispositivo y `adq_img_fnc()` para que adquiera una imagen a través de dicho dispositivo y posteriormente finalice la conexión con éste. A continuación `registrar_usuario_GUI` obtiene de `ICaptura` la imagen que el Operador capturó a través del dispositivo y emplea la operación `most_img_obt_fnc()` con el fin de desplegar en pantalla dicha imagen, finalmente activa el botón *Analizar* que permite solicitar la localización de un rostro en la imagen...*

* El flujo de sucesos continúa de la misma forma como está descrito en el *escenario Flujo Básico – Emplear una imagen existente*, de esta sección.









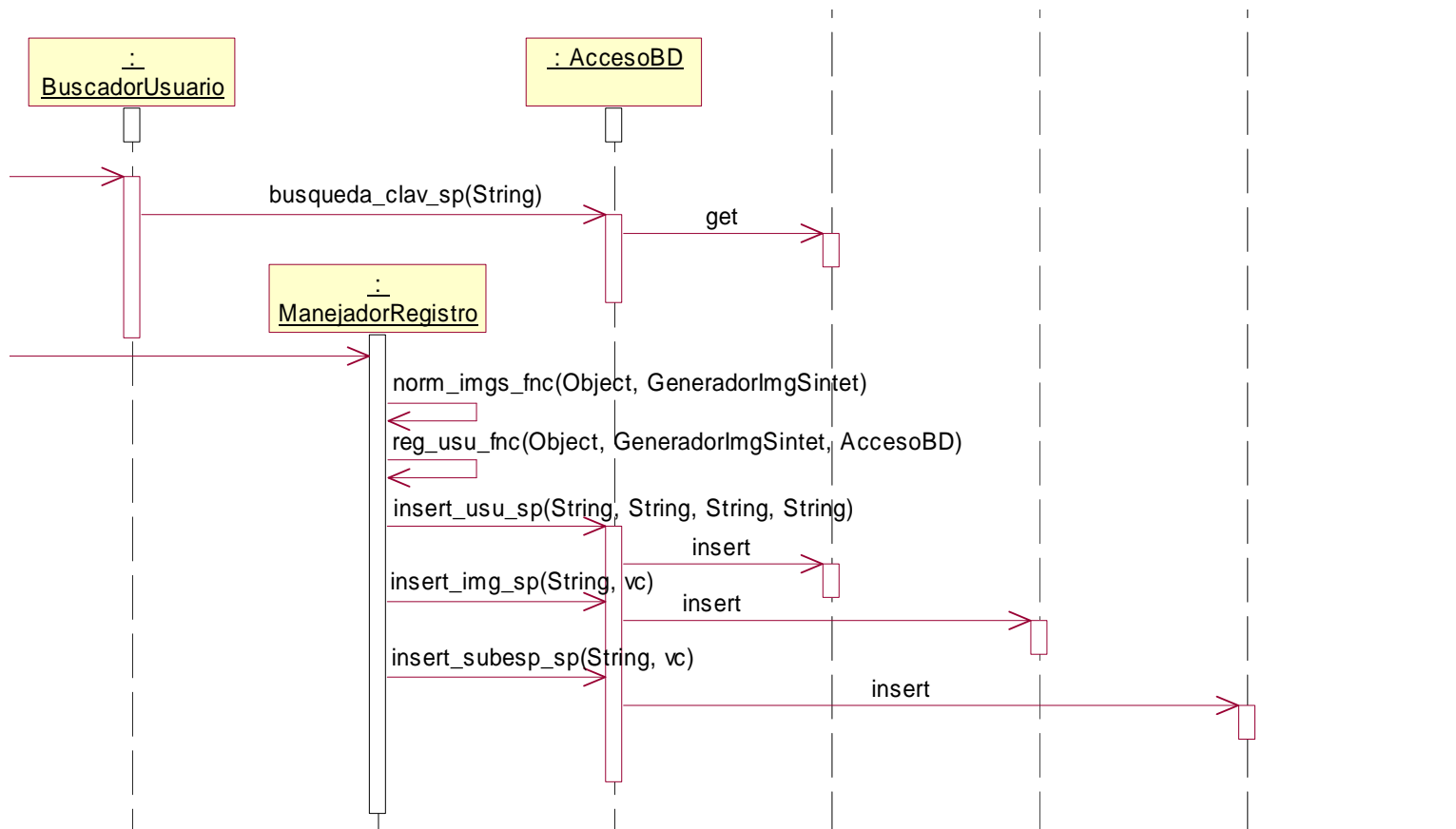


Figura 2.5. Diagrama de Secuencia - Diseño: Caso de uso *Registrar Usuario* (Flujo Básico – Capturar imagen)

- *Extensión – No se localizó ningún rostro en la imagen (emplear una imagen existente)*

Véase Figura 2.6.

Descripción del flujo de sucesos-diseño:

La secuencia se inicia cuando el Operador solicita a un objeto de la clase FENIXVR registrar un nuevo usuario al presionar el botón *Registrar Usuario*. Esto genera un evento que es atendido por FENIXVR a través de `btn_reg_usu_Callback()` que activa y despliega la ventana de registro de usuarios (`registrar_usuario_GUI`).

Para iniciar el proceso de registro el Operador selecciona dentro de `registrar_usuario_GUI` la opción correspondiente al escenario que se describe (*Obtener Imagen - Carpeta*), esto genera un evento que es manejado por `carpeta_Callback()` que a su vez se encarga de solicitar un objeto de la clase `ImagenExistente`, el cual ofrece la operación `selec_img_fnc()` que permite al Operador seleccionar la imagen que desea utilizar mediante un cuadro de diálogo estándar para recuperar archivos. El Operador selecciona una imagen y presiona el botón *Aceptar* del cuadro de diálogo, esto invoca a la operación `obt_img_selec_fnc()` proporcionada por `ImagenExistente` para cargar en memoria la imagen especificada en el cuadro de diálogo.

A continuación `registrar_usuario_GUI` obtiene de `ImagenExistente` la imagen que el Operador seleccionó y emplea la operación `most_img_obt_fnc()` con el fin de desplegar en pantalla dicha imagen, finalmente activa el botón *Analizar* que permite solicitar la localización de un rostro en la imagen.

Una vez que se ha presentado dentro de la ventana `registrar_usuario_GUI` la imagen que el Operador seleccionó, éste solicita la localización de un rostro en dicha imagen al presionar el botón *Analizar* que genera un evento que es atendido por

registrar_usuario_GUI a través de btn_analizar_Callback() que se encarga de comprobar que el número de imágenes del nuevo usuario que ya se han agregado para el registro es menor a cinco, en cuyo caso solicita los servicios del subsistema *Localizar Rostro* a través de la interfaz ManejadorLocalizacionRostro(detect_rostro_fnc()).

Luego que el subsistema *Localizar Rostro* ha finalizado el proceso, registrar_usuario_GUI obtiene del objeto retornado por el subsistema el valor que indica si ocurrió un error en la localización del rostro. Si ocurrió un error, registrar_usuario_GUI invoca la operación gest_msg_fnc() que es ofrecida por ManejadorErrorLocalización y le envía el valor de error retornado por el subsistema, para que a su vez gest_msg_fnc() retorne un objeto que contenga el mensaje de error correspondiente al valor enviado como parámetro de entrada. Una vez retornado este objeto registrar_usuario_GUI obtiene de éste el mensaje de error y lo presenta al Operador en un cuadro de diálogo.

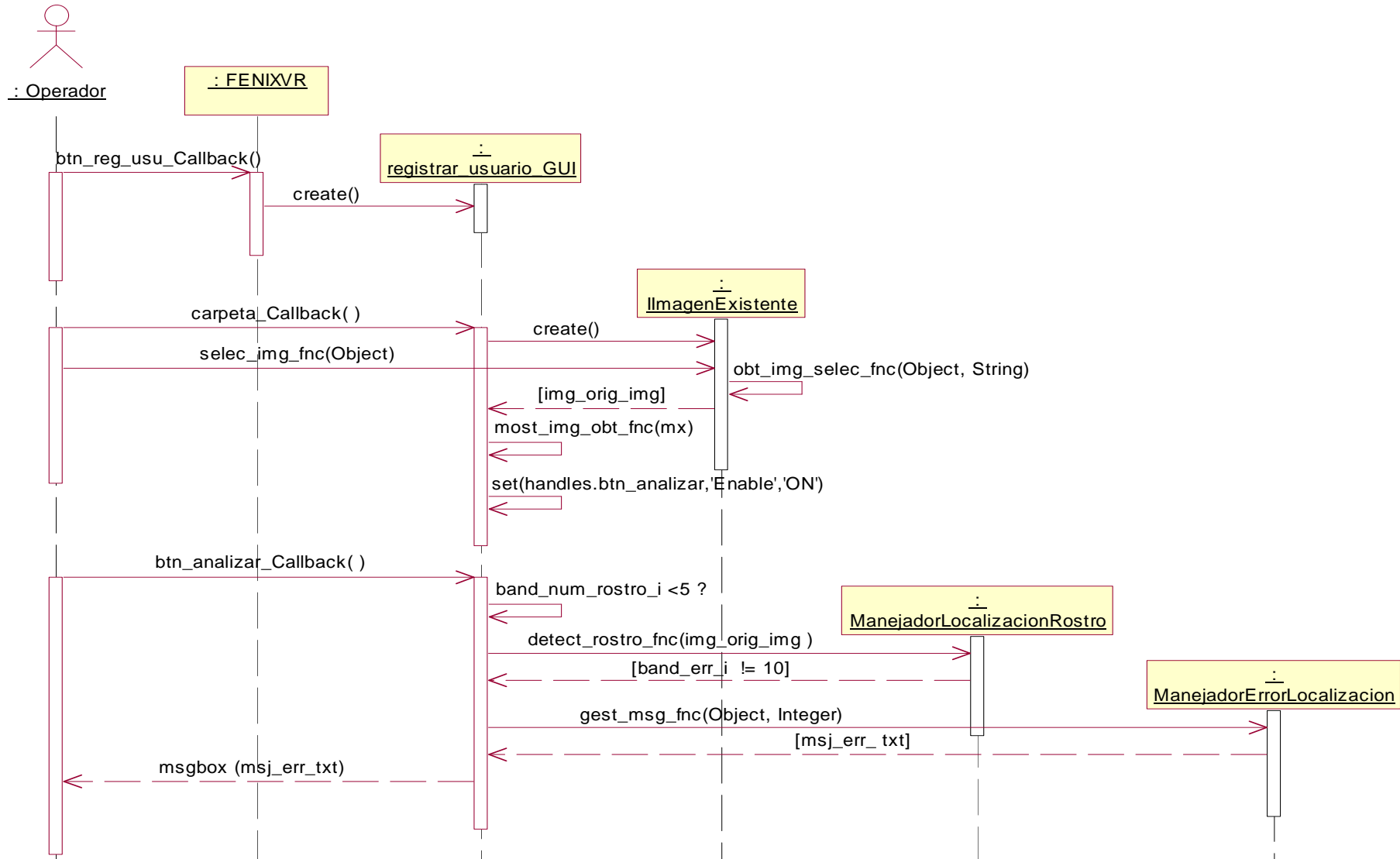


Figura 2.6. Diagrama de Secuencia - Diseño: Caso de uso *Registrar Usuario* (Extensión – No se localizó ningún rostro en la imagen)

2.3.1.2.2 Diagramas de Secuencia – Diseño: Caso de uso Verificar Identidad

- *Flujo Básico – Emplear una imagen existente*

Véase Figura 2.7.

Descripción del flujo de sucesos-diseño:

La secuencia se inicia cuando el Operador solicita a un objeto de la clase FENIXVR verificar la identidad de un usuario al presionar el botón *Verificar Identidad*. Esto genera un evento que es atendido por FENIXVR a través de `btn_verif_ident_Callback()` que activa y despliega la ventana de verificación de identidad (`verif_ident_GUI`).

Para iniciar el proceso de verificación el Operador selecciona dentro de `verif_ident_GUI` la opción correspondiente al escenario que se describe (*Obtener Imagen - Carpeta*), esto genera un evento que es manejado por `carpeta_Callback()` que a su vez se encarga de solicitar un objeto de la clase `ImagenExistente`, el cual ofrece la operación `selec_img_fnc()` que permite al Operador seleccionar la imagen que desea utilizar mediante un cuadro de diálogo estándar para recuperar archivos. El Operador selecciona una imagen y presiona el botón *Aceptar* del cuadro de diálogo, esto invoca a la operación `obt_img_selec_fnc()` proporcionada por `ImagenExistente` para cargar en memoria la imagen especificada en el cuadro de diálogo.

A continuación `verif_ident_GUI` obtiene de `ImagenExistente` la imagen que el Operador seleccionó y emplea la operación `most_img_obt_fnc()` con el fin de desplegar en pantalla dicha imagen, finalmente activa el botón *Analizar* que permite solicitar la localización de un rostro en la imagen. Una vez que se ha presentado dentro de la ventana `verif_ident_GUI` la imagen que el Operador seleccionó, éste solicita la localización de un rostro en dicha imagen al presionar el botón *Analizar* que genera un evento que es atendido por `verif_ident_GUI` a través de `btn_analizar_Callback()` que se encarga de solicitar los servicios del

subsistema *Localizar Rostro* a través de la interfaz `ManejadorLocalizacionRostro` (`detect_rostro_fnc()`).

Luego que el subsistema *Localizar Rostro* ha encontrado un rostro en la imagen y ha presentado esta nueva imagen (con el área del rostro señalada) dentro de la ventana `verif_ident_GUI` se solicita al Operador confirmar si desea emplear el rostro localizado para la verificación de identidad. Si el Operador confirma que desea emplear la imagen del rostro localizado para la verificación de identidad, `verif_ident_GUI` solicita un objeto de la clase `PatronImagen` que provee la operación `calc_img_recons_fnc()` con el fin de calcular el vector de pesos asociado a la imagen del rostro localizado y luego reconstruir esta imagen a partir de dicho vector de pesos. Esta operación por su parte utiliza la operación `proy_img_fnc()` de `ImagenReconstruida` para proyectar en el espacio de rostros la imagen que recibe como parámetro de entrada y obtener así el vector de pesos asociado.

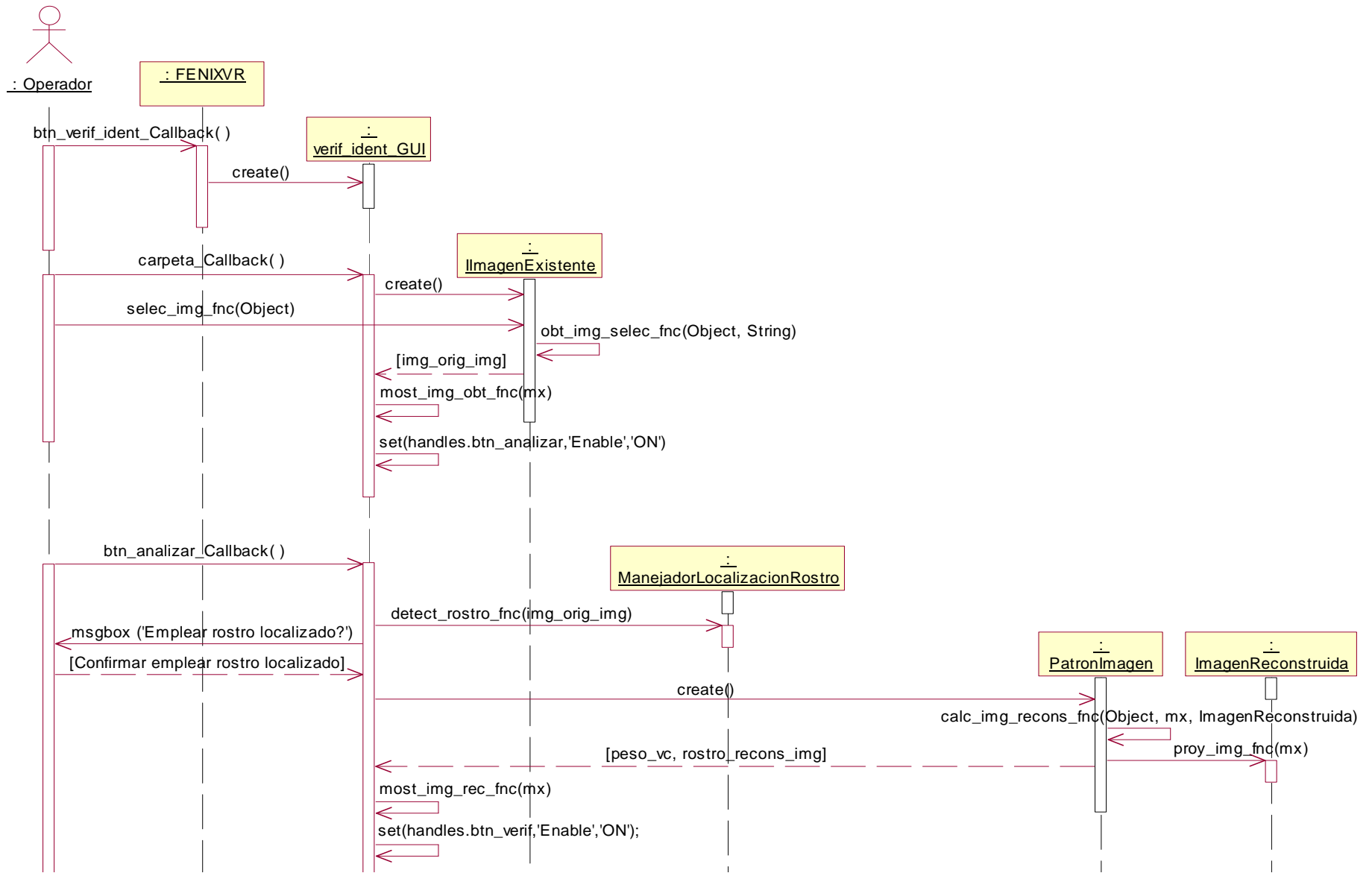
Una vez calculado el vector de pesos `verif_ident_GUI` obtiene de `PatronImagen` dicho vector y la imagen reconstruida a partir de éste y la presenta en pantalla empleando la operación `most_img_rec_fnc()`, adicionalmente activa el botón *Verificar Identidad* que permite al Operador solicitar completar el proceso de verificación de identidad.

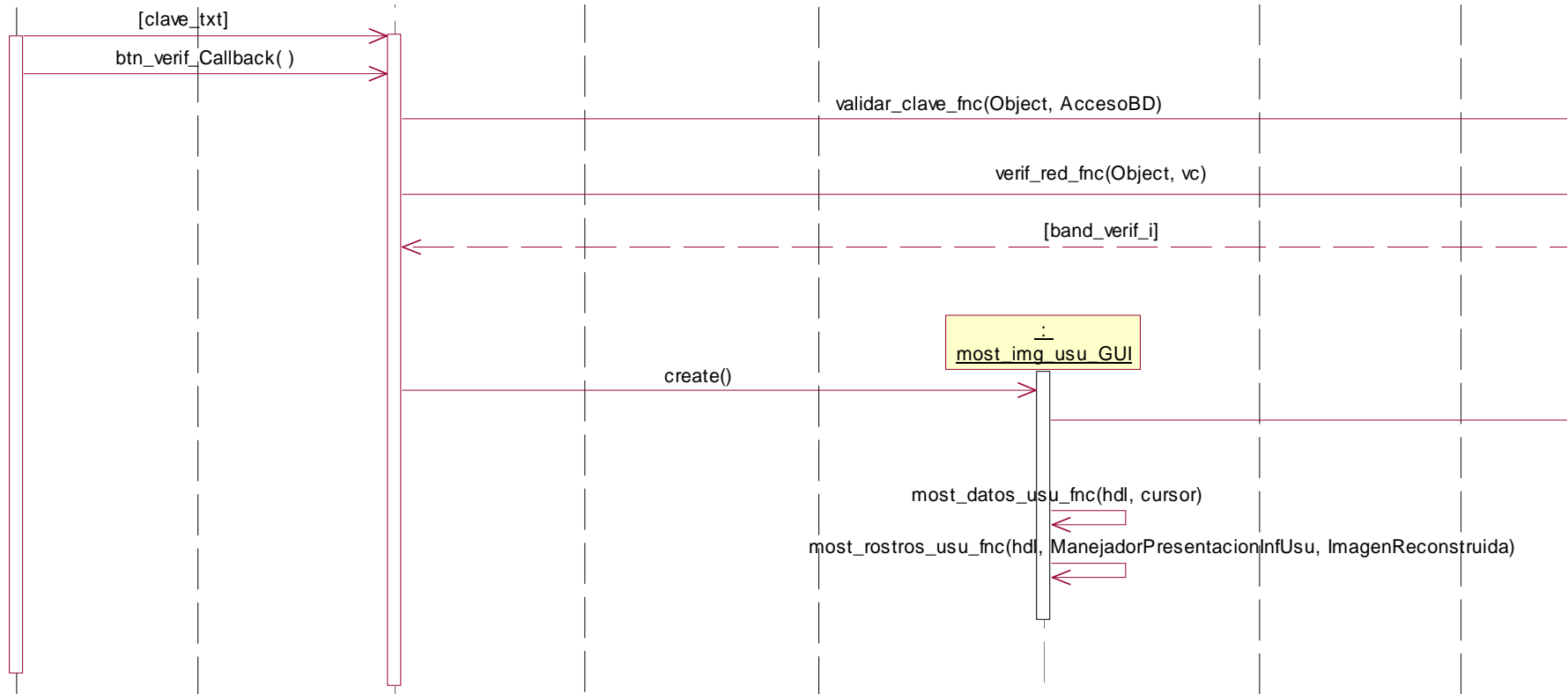
Seguidamente el Operador ingresa la clave que corresponde al usuario cuya imagen ha sido proporcionada y solicita verificar la identidad de dicho usuario, para esto presiona el botón *Verificar Identidad* que genera un evento que es atendido por `verif_ident_GUI` a través de `btn_verif_Callback()` que a su vez emplea la operación `validar_clave_fnc()` ofrecida por `BuscadorUsuario`, para controlar que la clave ingresada corresponda a un usuario previamente registrado. Esta operación emplea a `busqueda_clav_sp()` que es proporcionada por `AccesoBD` para que acceda a la tabla `Usuario` en la base de datos y

consulte si la clave ingresada por el Operador pertenece a un usuario ya registrado.

Si la clave ingresada pertenece a un usuario previamente registrado `verif_ident_GUI` invoca la operación `verif_red_fnc()` ofrecida por `ManejadorVerificacion`. Esta operación a su vez emplea la operación `sim()` que es suministrada por la herramienta de desarrollo para realizar la simulación de la red neuronal que se le envía como parámetro de entrada y retorna una estructura que contiene la respuesta de la red, `verif_red_fnc()` obtiene la salida de la red (respuesta) y busca aquella salida con la mayor puntuación con el fin de compararla con `val_comparacion_db` que es un valor empleado como referencia y que es constante para todo objeto de la clase `ManejadorVerificacion`. Si la salida es mayor o igual al valor de referencia, `verif_red_fnc()` determina la identidad asociada con dicha salida (identidad que de acuerdo a la red neuronal corresponde el vector de pesos proporcionado también como parámetro de entrada para la simulación), finalmente valida si la identidad obtenida después de este proceso es igual a la proporcionada por el Operador (es decir, autentica al usuario).

Finalizada la verificación `verif_ident_GUI` obtiene de `ManejadorVerificación` el valor que determina si las claves coinciden (`band_verif_i`), en cuyo caso activa y despliega la ventana que contiene la información del usuario autenticado (`most_img_usu_GUI`). Antes que `most_img_usu_GUI` se haga visible, invoca la operación `obt_inf_usu_fnc()` ofrecida por `ManejadorPresentacionInfUsu`, que a su vez utiliza las operaciones `select_dat_usu_sp()` y `select_imgs_usu_sp()` proporcionadas por `AccesoBD` y que acceden a las tablas `Usuario` e `Imagen` respectivamente para obtener los datos e imágenes del usuario autenticado empleando la clave de éste. Luego que se ha obtenido la información del usuario autenticado `most_img_usu_GUI` la presenta utilizando para esto las operaciones `most_datos_usu_fnc()` y `most_rostros_usu_fnc()`.





: red_rd

: Usuario

: Imagen

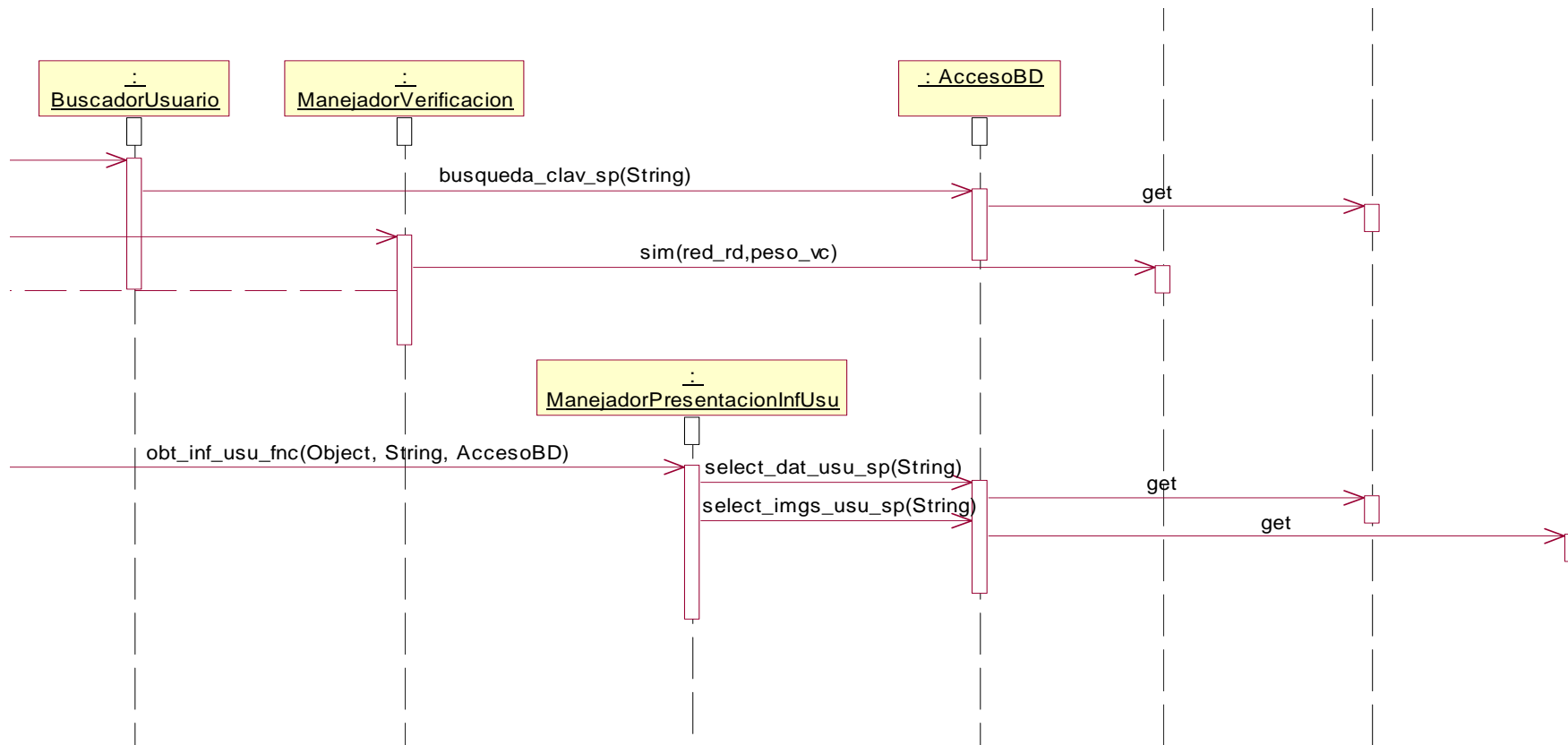


Figura 2.7. Diagrama de Secuencia - Diseño: Caso de uso *Verificar Identidad* (Flujo Básico - Imagen existente)

- *Flujo Básico – Capturar imagen*

Véase Figura 2.8.

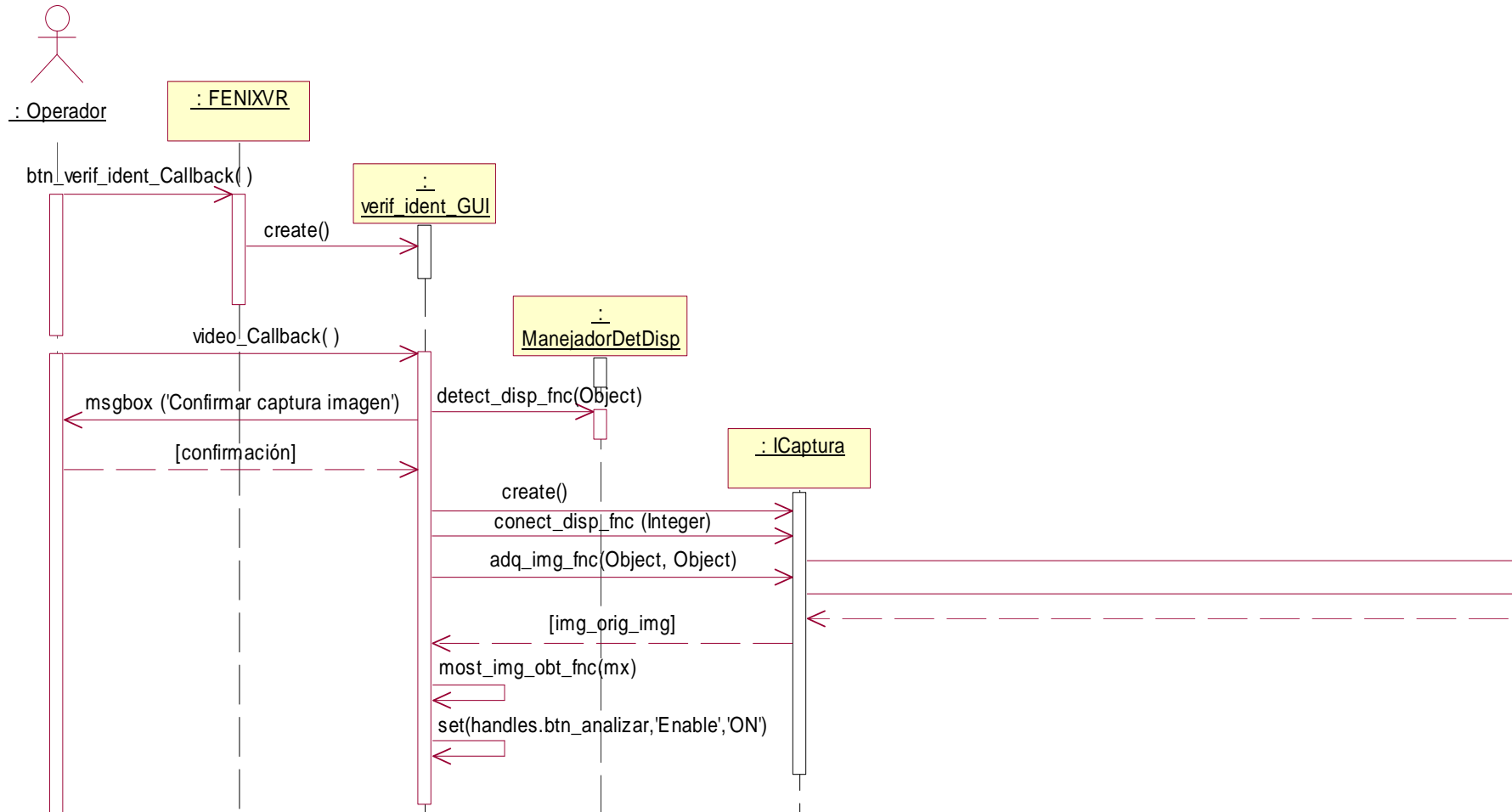
Descripción del flujo de sucesos-diseño:

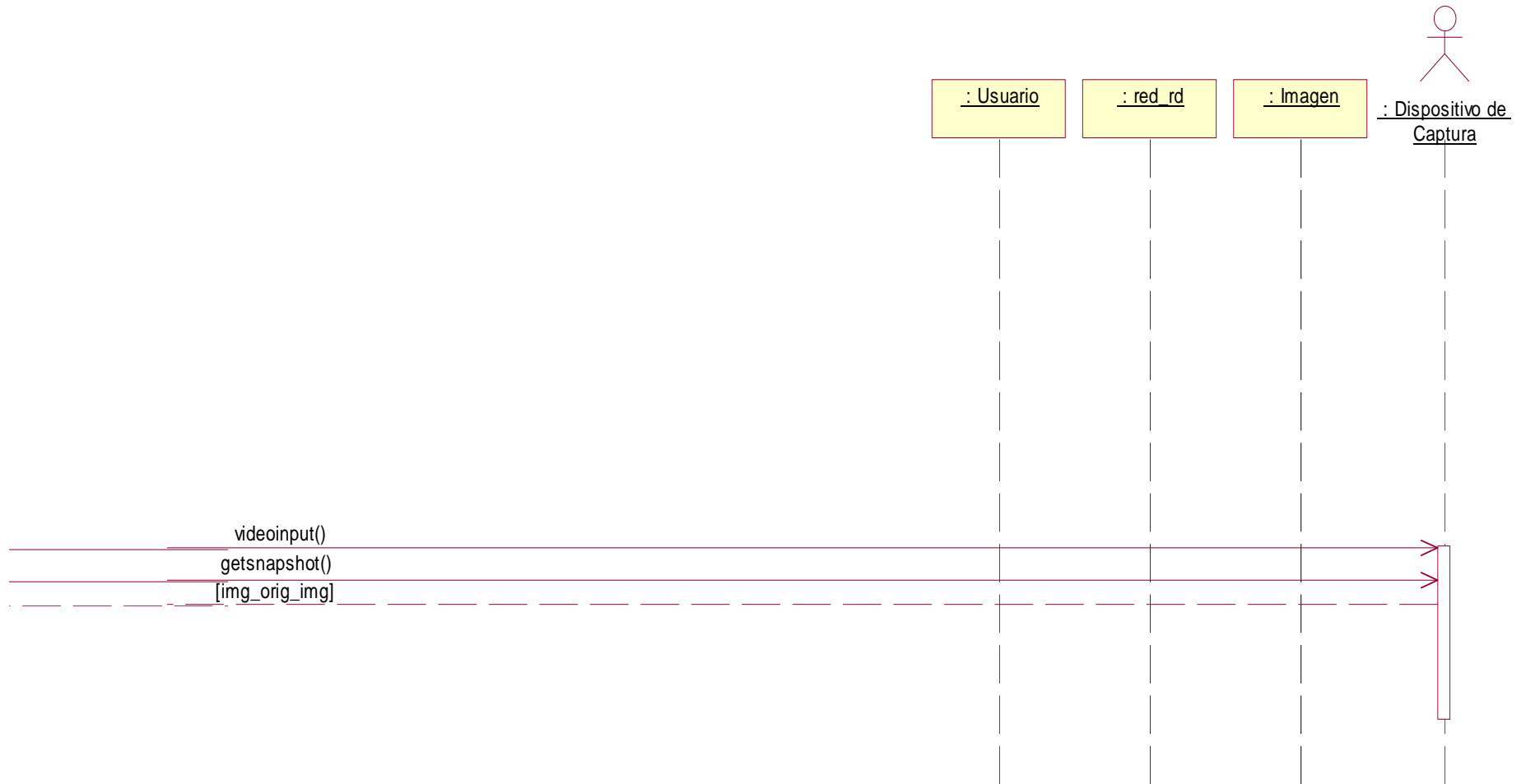
La secuencia se inicia cuando el Operador solicita a un objeto de la clase FENIXVR verificar la identidad de un usuario al presionar el botón *Verificar Identidad*. Esto genera un evento que es atendido por FENIXVR a través de `btn_verif_ident_Callback()` que activa y despliega la ventana de verificación de identidad (`verif_ident_GUI`).

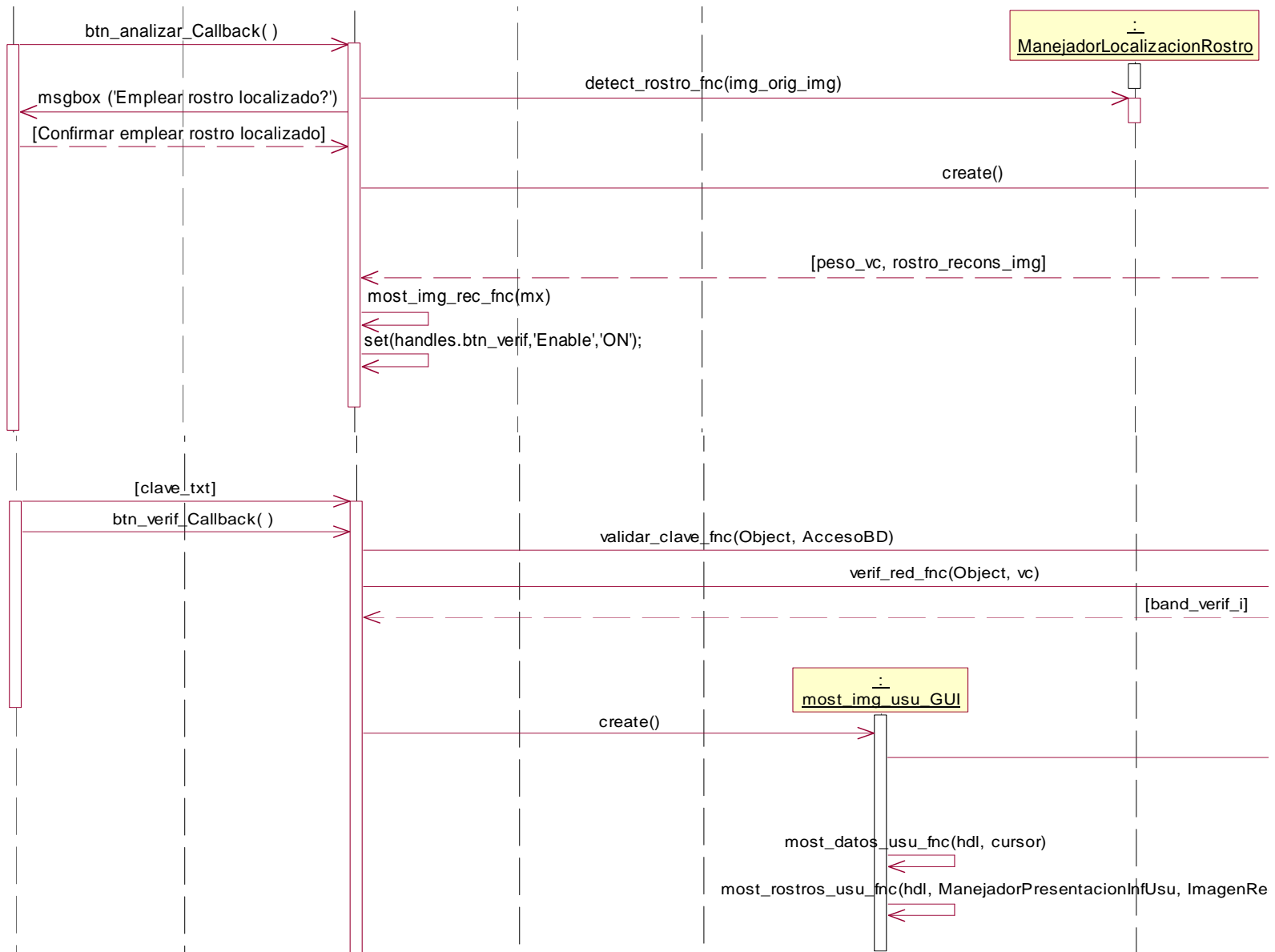
Para iniciar el proceso de verificación el Operador selecciona dentro de `verif_ident_GUI` la opción correspondiente al escenario que se describe (*Obtener Imagen - Video*), esto genera un evento que es manejado por `video_Callback()` que emplea la operación `detect_disp_fnc()` proporcionada por `ManejadorDetDisp` para detectar si existe un dispositivo de captura que se encuentre accesible a través de una interfaz genérica de Windows para dispositivos de video. Si se ha detectado un dispositivo que cumpla estas condiciones se solicita al Operador confirmar si desea capturar una imagen con ese dispositivo.

Si el Operador confirma el uso del dispositivo, `registrar_usuario_GUI` solicita un objeto de la clase `ICaptura` e invoca a las operaciones que éste ofrece: `conect_disp_fnc()` para que se encargue de establecer la conexión con el dispositivo y `adq_img_fnc()` para que adquiera una imagen a través de dicho dispositivo y posteriormente finalice la conexión con éste. A continuación `verif_ident_GUI` obtiene de `ICaptura` la imagen que el Operador capturó a través del dispositivo y emplea la operación `most_img_obt_fnc()` con el fin de desplegar en pantalla dicha imagen, finalmente activa el botón *Analizar* que permite solicitar la localización de un rostro en la imagen ... *

* El flujo de sucesos continúa de la misma forma como está descrito en el *escenario Flujo Básico – Emplear una imagen existente*, de esta sección.







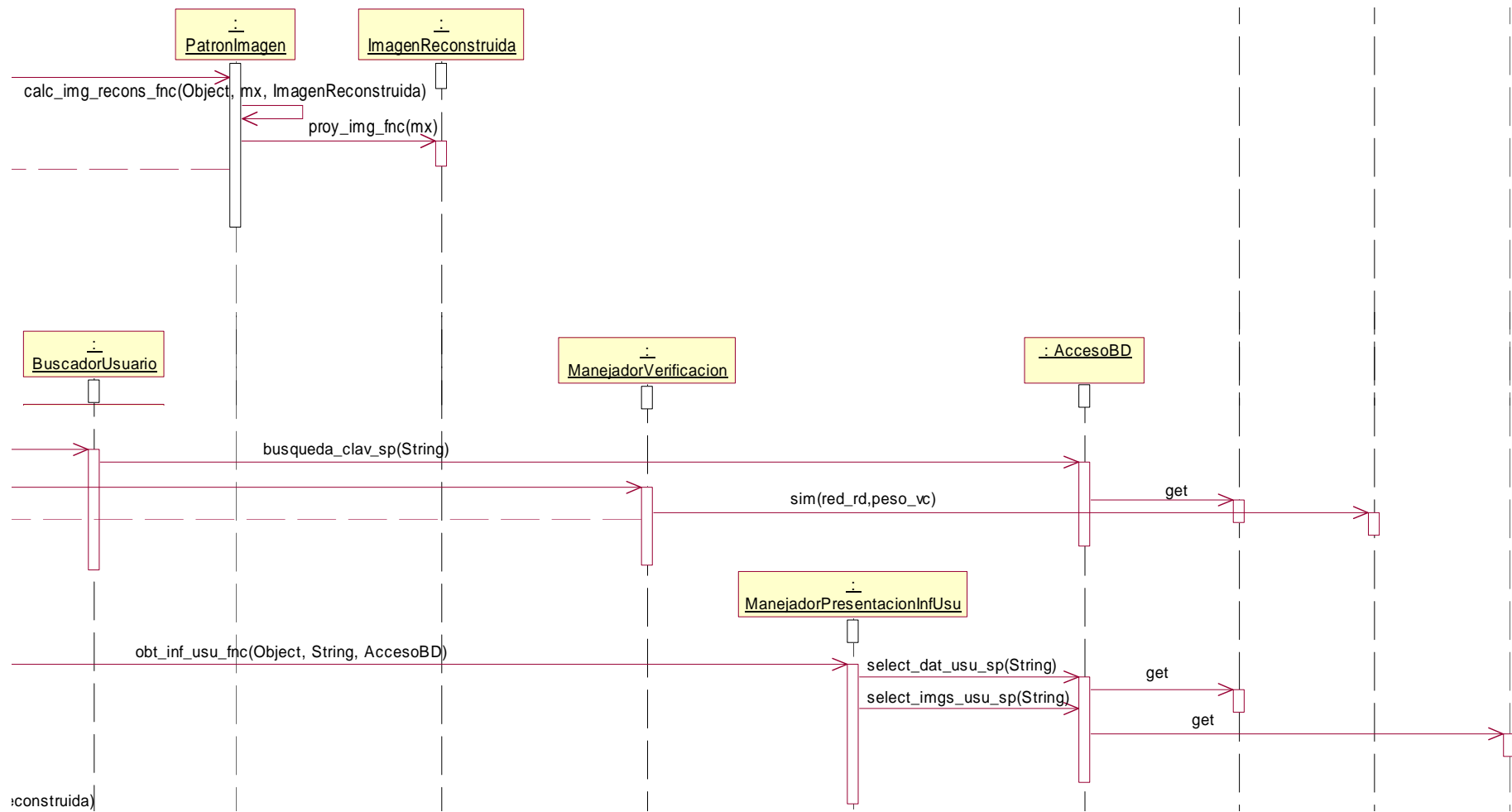


Figura 2.8. Diagrama de Secuencia - Diseño: Caso de uso *Verificar Identidad* (Flujo Básico – Capturar imagen)

- *Extensión – No se localizó ningún rostro en la imagen (emplear una imagen existente)*

Véase Figura 2.9.

Descripción del flujo de sucesos-diseño:

La secuencia se inicia cuando el Operador solicita a un objeto de la clase FENIXVR verificar la identidad de un usuario al presionar el botón *Verificar Identidad*. Esto genera un evento que es atendido por FENIXVR a través de `btn_verif_ident_Callback()` que activa y despliega la ventana de verificación de identidad (`verif_ident_GUI`).

Para iniciar el proceso de verificación el Operador selecciona dentro de `verif_ident_GUI` la opción correspondiente al escenario que se describe (*Obtener Imagen - Carpeta*), esto genera un evento que es manejado por `carpeta_Callback()` que a su vez se encarga de solicitar un objeto de la clase `ImagenExistente`, el cual ofrece la operación `selec_img_fnc()` que permite al Operador seleccionar la imagen que desea utilizar mediante un cuadro de diálogo estándar para recuperar archivos. El Operador selecciona una imagen y presiona el botón *Aceptar* del cuadro de diálogo, esto invoca a la operación `obt_img_selec_fnc()` proporcionada por `ImagenExistente` para cargar en memoria la imagen especificada en el cuadro de diálogo.

A continuación `verif_ident_GUI` obtiene de `ImagenExistente` la imagen que el Operador seleccionó y emplea la operación `most_img_obt_fnc()` con el fin de desplegar en pantalla dicha imagen, finalmente activa el botón *Analizar* que permite solicitar la localización de un rostro en la imagen.

Una vez que se ha presentado dentro de la ventana `verif_ident_GUI` la imagen que el Operador seleccionó, éste solicita la localización de un rostro en dicha imagen al presionar el botón *Analizar* que genera un evento que es atendido por `verif_ident_GUI` a través de

btn_analizar_Callback() que se encarga de solicitar los servicios del subsistema *Localizar Rostro* a través de la interfaz ManejadorLocalizacionRostro (detect_rostro_fnc()).

Luego que el subsistema *Localizar Rostro* ha finalizado el proceso, verif_ident_GUI obtiene del objeto retornado por el subsistema el valor que indica si ocurrió un error en la localización del rostro. Si ocurrió un error, verif_ident_GUI invoca la operación gest_msg_fnc() que es proporcionada por ManejadorErrorLocalización y le envía el valor de error retornado por el subsistema, para que a su vez gest_msg_fnc() retorne un objeto que contenga el mensaje de error correspondiente al valor enviado como parámetro de entrada.

Una vez retornado este objeto registrar_usuario_GUI obtiene de éste el mensaje de error y lo presenta al Operador en un cuadro de diálogo.

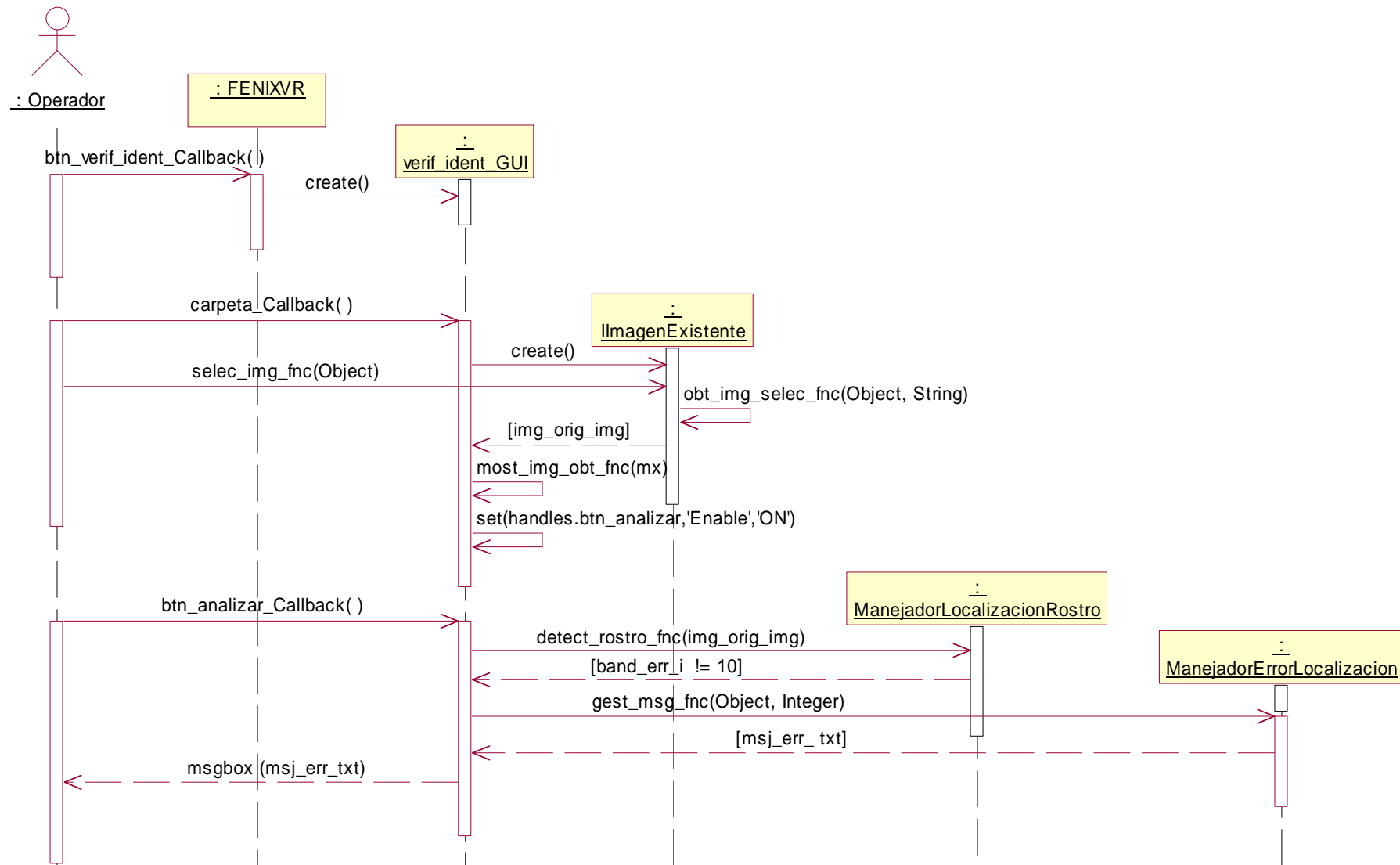
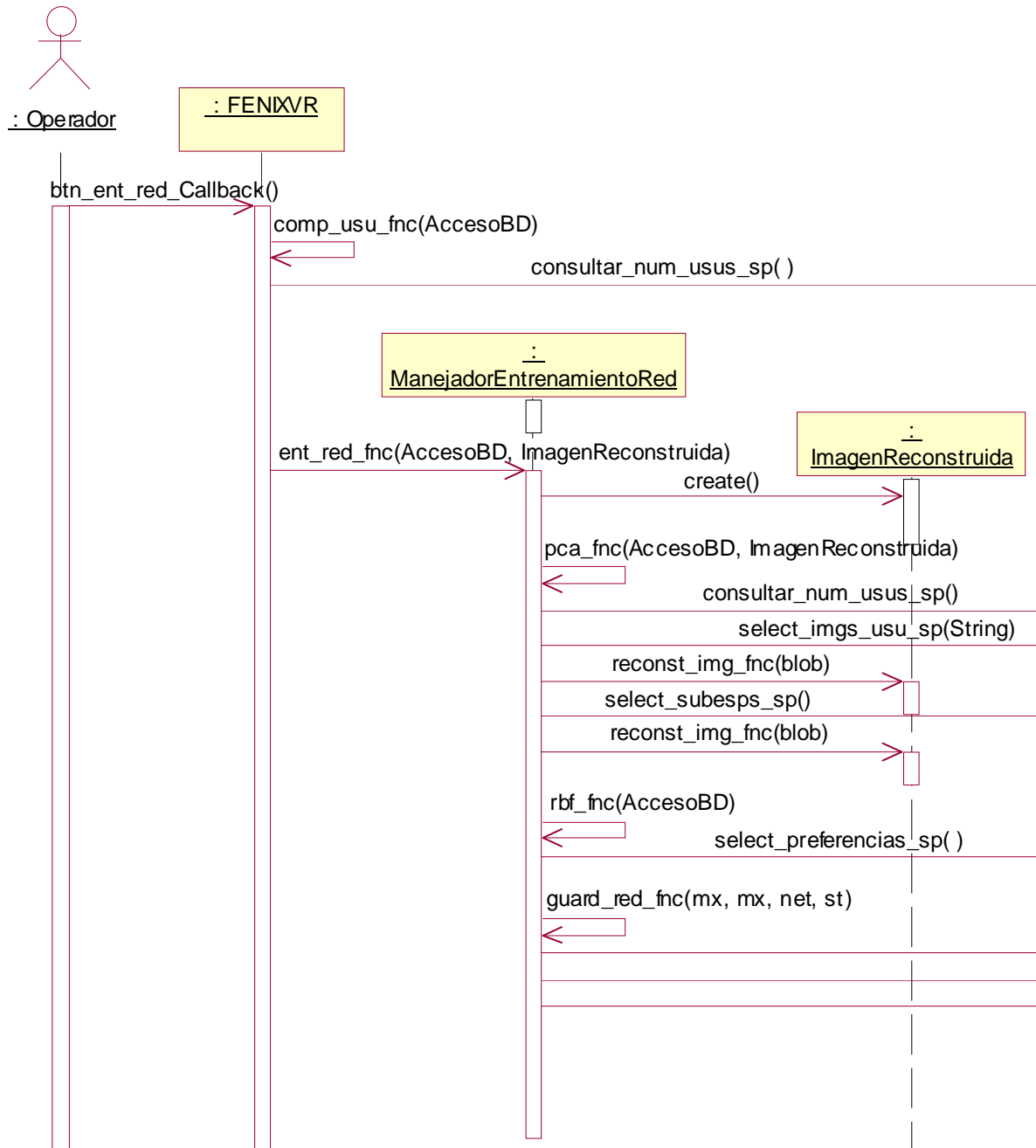


Figura 2.9. Diagrama de Secuencia - Diseño: Caso de uso *Verificar Identidad* (Extensión – No se localizó ningún rostro en la imagen)

2.3.1.2.3 Diagrama de Secuencia – Diseño: Caso de uso Entrenar Red Neuronal

○ Flujo Básico



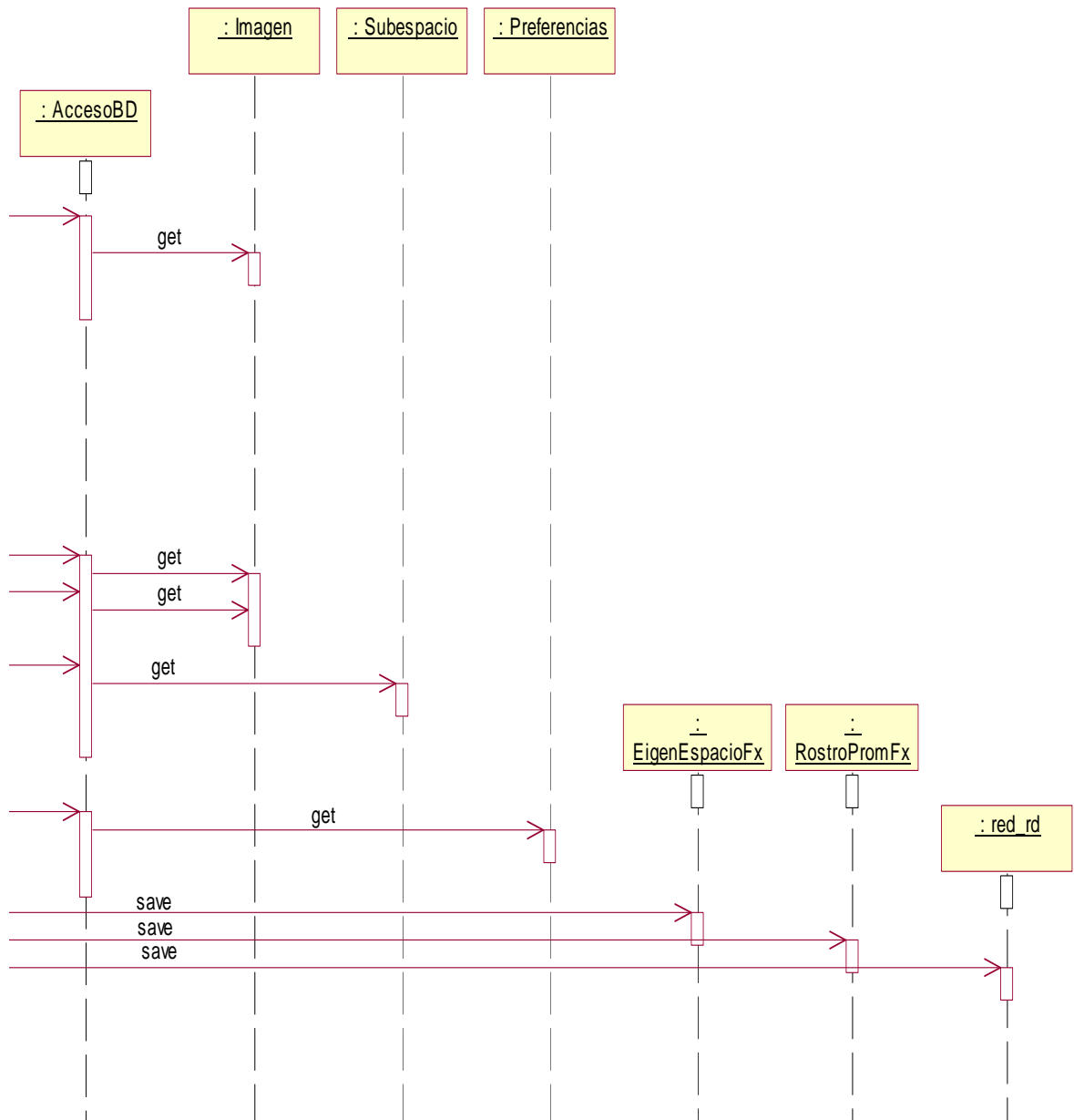


Figura 2.10. Diagrama de Secuencia - Diseño: Caso de uso *Entrenar Red Neuronal* (Flujo Básico)

Descripción del flujo de sucesos-diseño:

La secuencia se inicia cuando el Operador solicita a un objeto de la clase FENIXVR entrenar una red neuronal al presionar el botón *Entrenar Red Neuronal*. Esto genera un evento que es atendido por FENIXVR a través de `btn_ent_red_Callback()` que utiliza la operación `comp_usu_fnc()` para verificar que existan usuarios previamente registrados. Esta operación a su vez emplea la operación `consultar_num_usus_sp()` que es proporcionada por AccesoBD para que acceda a la tabla Imagen en la base de datos y consulte el número de usuarios previamente registrados.

Si existe un número adecuado de usuarios registrados FENIXVR invoca la operación `ent_red_fnc()` ofrecida por `ManejadorEntrenamientoRed`. Esta operación a su vez solicita un objeto de la clase `ImagenReconstruida` e invoca la operación `pca_fnc()` de `ManejadorEntrenamientoRed`, con el fin de crear el espacio de rostros conocidos a partir de las imágenes almacenadas de los usuarios registrados. Para esto primero obtiene las imágenes almacenadas (objetos BLOB) utilizando las operaciones `consultar_num_usus_sp()`, `select_imgs_usu_sp()` y `select_subesps_sp()` de AccesoBD, luego emplea la operación `reconst_img_fnc()` del objeto `ImagenReconstruida` para transformar cada objeto BLOB a una estructura bidimensional (matriz), finalmente crea el espacio de rostros. Una vez creado dicho espacio `pca_fnc()` proyecta sobre éste las imágenes reconstruidas y obtiene los vectores de pesos asociados a cada usuario registrado. Luego de esto se emplea la operación `rbf_fnc()` de `ManejadorEntrenamientoRed` para crear y entrenar una red neuronal de tipo RBF con los vectores de pesos antes calculados. Esta operación a su vez invoca a `select_preferencias_sp()` de AccesoBD para consultar los valores actuales de los parámetros para la red neuronal (número de neuronas añadidas por desplazamiento y error mínimo cuadrático estándar), con el fin de usarlos en la creación y entrenamiento de la red.

Finalizado el entrenamiento se emplea la operación `guard_red_fnc()` de `ManejadorEntrenamientoRed` para almacenar en disco la red neuronal y toda la información requerida para la verificación de identidad.

2.3.1.2.4 *Diagrama de Secuencia – Diseño: Caso de uso Actualizar Parámetros de la Red Neuronal*

o *Flujo Básico*

Véase Figura 2.11.

Descripción del flujo de sucesos-diseño:

La secuencia se inicia cuando el Operador solicita a un objeto de la clase `FENIXVR` modificar los parámetros de creación y entrenamiento de la red neuronal al presionar el botón *Preferencias*. Esto genera un evento que es atendido por `btn_preferencias_Callback()` que activa y despliega la ventana de preferencias (`preferencias_GUI`). Antes que esta ventana sea visible, invoca la operación `obt_val_param_fnc()` de `ManejadorParametros` para obtener los valores actuales de los parámetros que pueden ser modificados, esta operación a su vez utiliza a `select_preferencias_sp()` de `AccesoBD` para consultar dichos valores en la tabla `Preferencias`. Luego `preferencias_GUI` obtiene de `ManejadorParametros` los valores actuales y los presenta en pantalla utilizando la operación `most_val_act_fnc()`.

El Operador modifica los valores y solicita actualizarlos presionando el botón *Salir* que genera un evento que es atendido por `preferencias_GUI` a través de `btn_salir_Callback()`, que a su vez emplea la operación `valid_format_param_fnc()` para comprobar que los valores ingresados estén dentro de los rangos permitidos para los parámetros a los que representan. Validados los datos se utiliza la operación `act_val_param_fnc()` de `ManejadorParametros`, que emplea la operación `actualizar_preferencias_sp()` de `AccesoBD` para que se encargue de actualizar los valores en la tabla `Preferencias`.

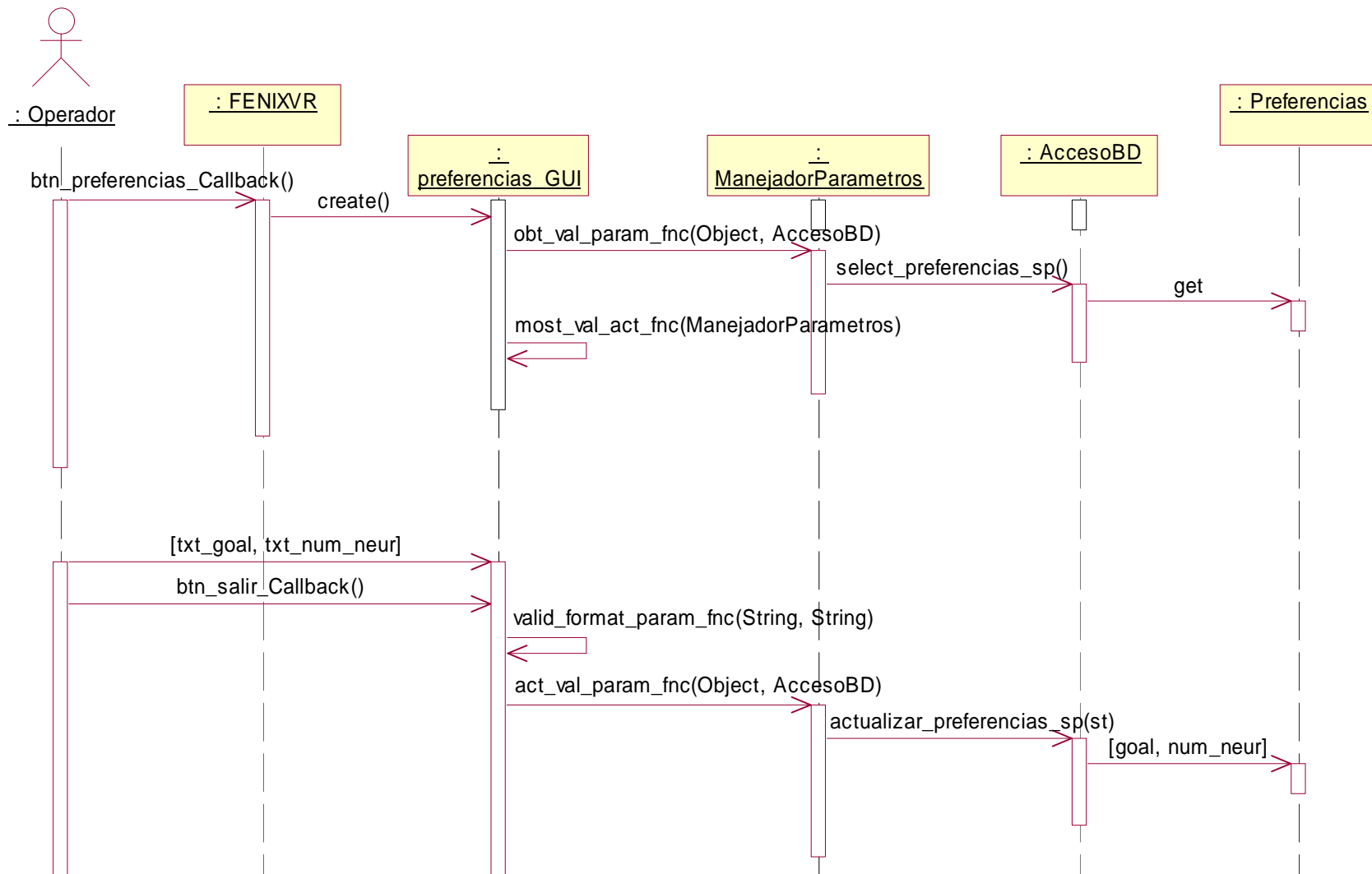


Figura 2.11. Diagrama de Secuencia - Diseño: Caso de uso Actualizar Parámetros de la Red Neuronal (Flujo Básico)

2.3.1.3 Diagrama de Clases

El diagrama de clases ha sido dividido en dos capas: la capa de aplicación que contiene aquellas clases relacionadas con la interfaz de usuario y la lógica inmersa en la solución computacional, y la capa de datos que agrupa las clases relacionadas con información de tipo persistente. Las Figuras 2.12 hasta 2.15 muestran los diagramas de clases resultantes.

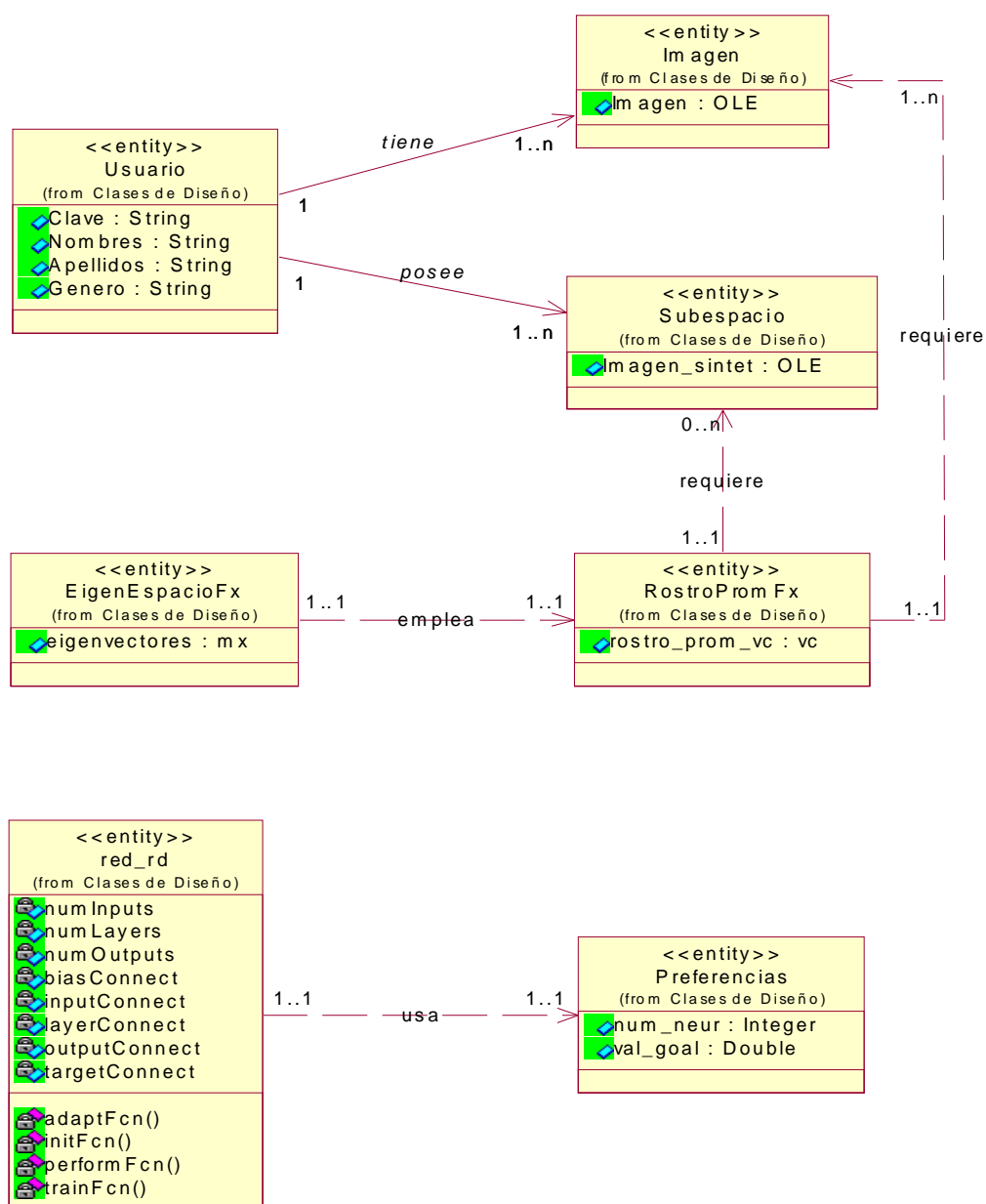


Figura 2.12. Diagrama de Clases – Capa de Datos

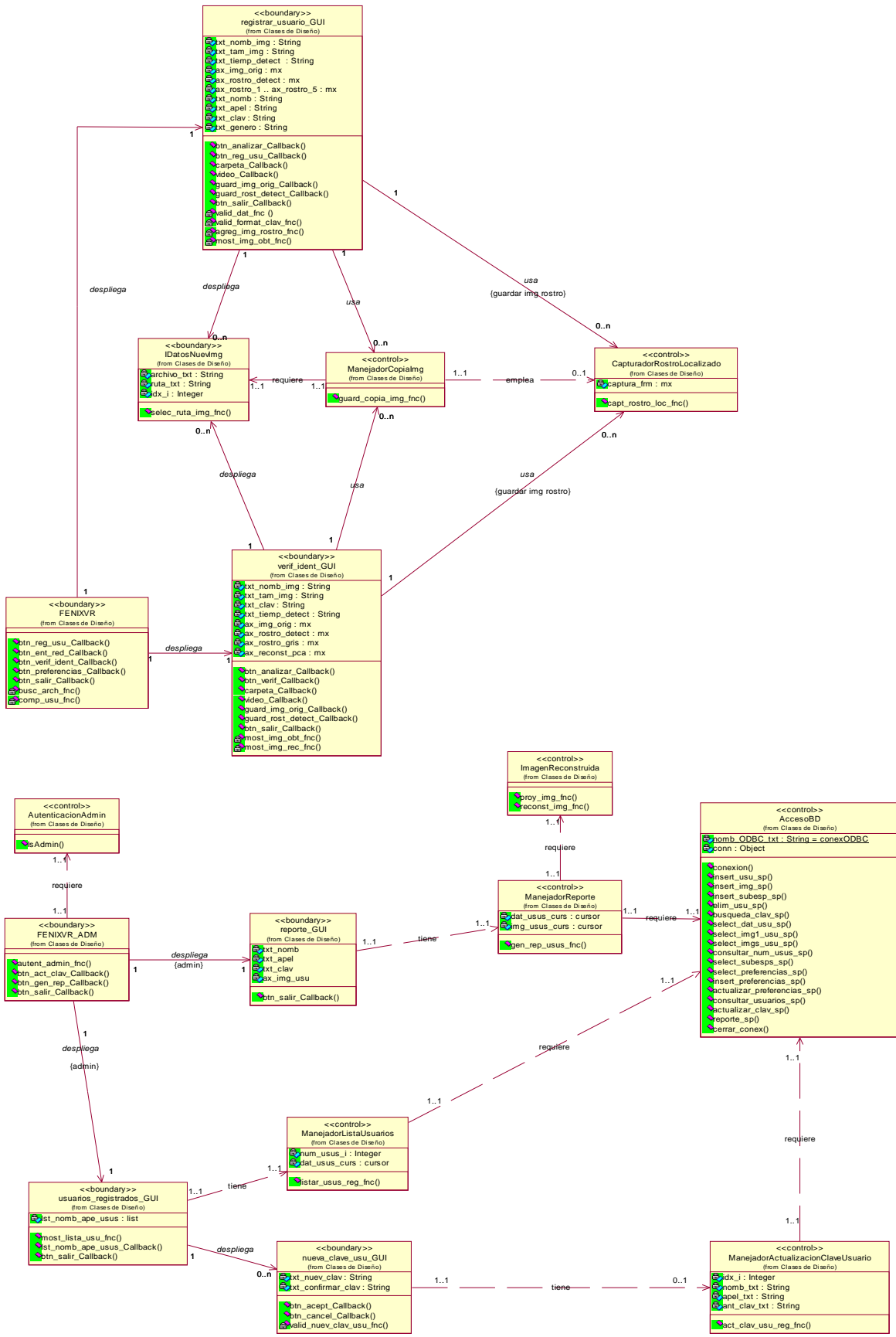


Figura 2.14. Diagrama de Clases – Capa de Aplicación (Funcionalidad Adicional)

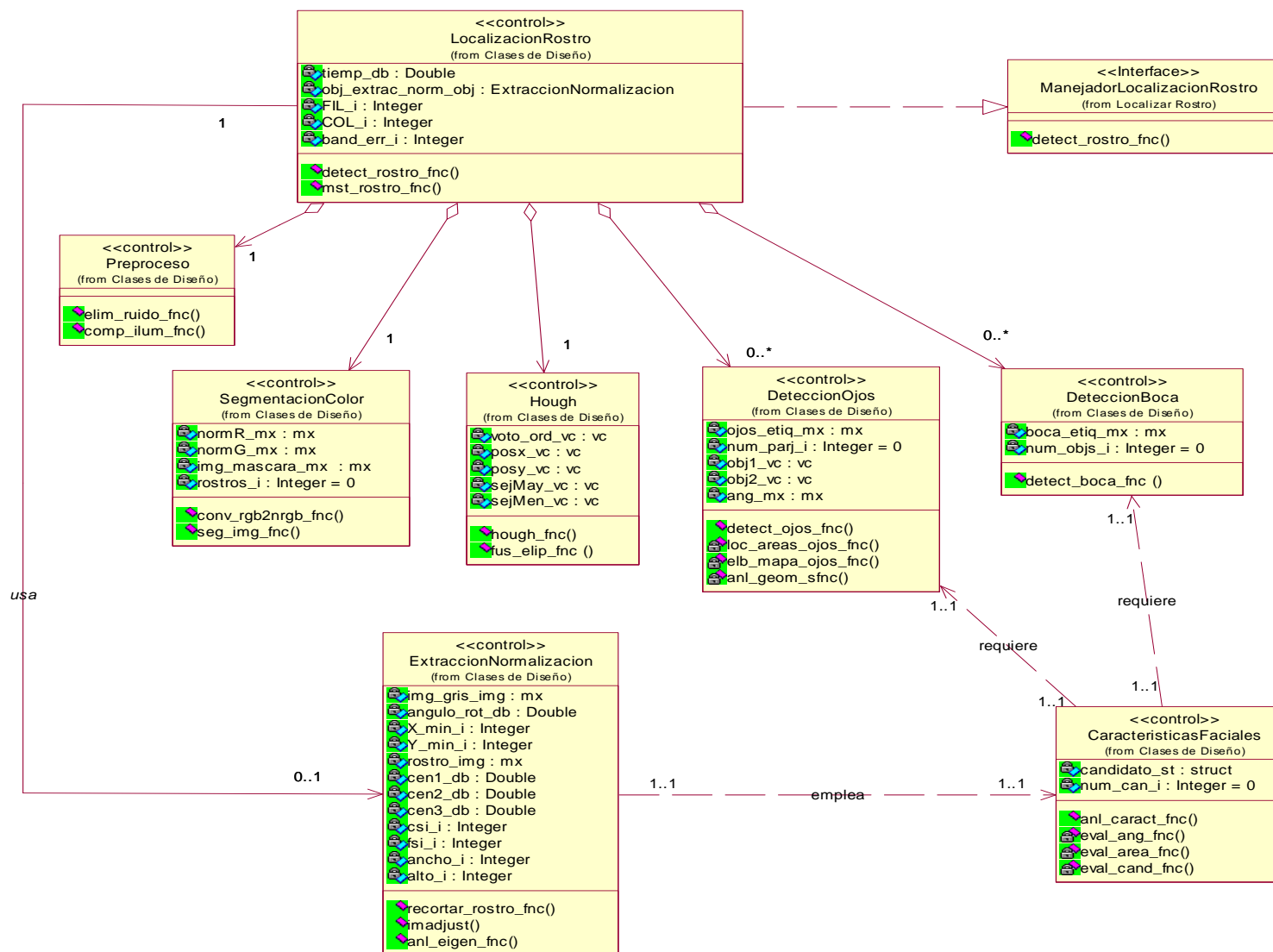


Figura 2.15. Diagrama de Clases – Capa de Aplicación (Subsistema Localizar Rostro)

2.3.2 IDENTIFICACIÓN DE SUBSISTEMAS Y SUS INTERFACES

Los subsistemas permiten organizar el modelo de diseño en piezas más manejables. Un subsistema de servicio ofrece servicios en términos de interfaces y operaciones y suele originar un componente ejecutable en la implementación.

La identificación de subsistemas de servicio durante el diseño toma como base inicial los paquetes de servicio del modelo de análisis, generalmente con una traza uno a uno.

También existen los subsistemas intermedios y de software, entre estos se encuentran los sistemas operativos, SGBD, software de soporte, kits de construcción de GUIs, etc.

A continuación, la Figura 2.16 muestra los subsistemas de servicio identificados como parte de la solución computacional, sus interfaces y las dependencias entre ellos.

Adicionalmente se muestra una dependencia a un paquete global que es accesible por todos los subsistemas, puesto que contiene clases de tipo control requeridas en diferentes realizaciones ya sea para la gestión de datos o para el control y gestión de errores de localización de un rostro.

Cabe aclarar que en dicha figura no se incluyen subsistemas propios del lenguaje utilizado para la implementación u otros subsistemas intermedios.

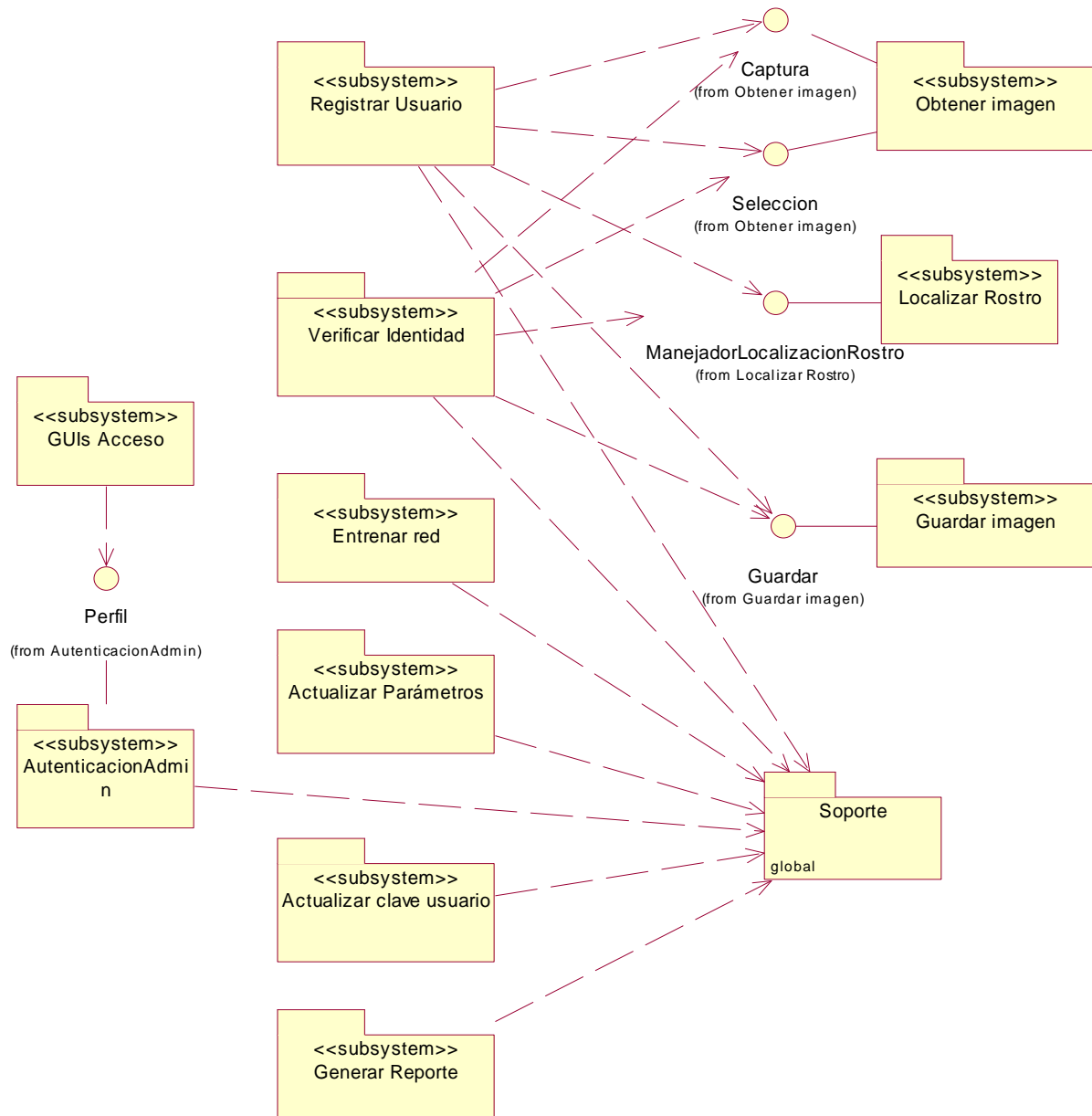


Figura 2.16. Identificación de Subsistemas e interfaces

2.3.2.1 Detalle de los Subsistemas de Diseño

El Anexo 18 muestra el detalle de los subsistemas de diseño.

2.3.3 DISEÑO DE PANTALLAS

2.3.3.1 Modelo de Navegación de Interfaz

A continuación se muestran las distintas interfaces gráficas de usuario (GUIs) y la forma cómo se vinculan entre sí, (véase Figuras 2.17 y 2.18).

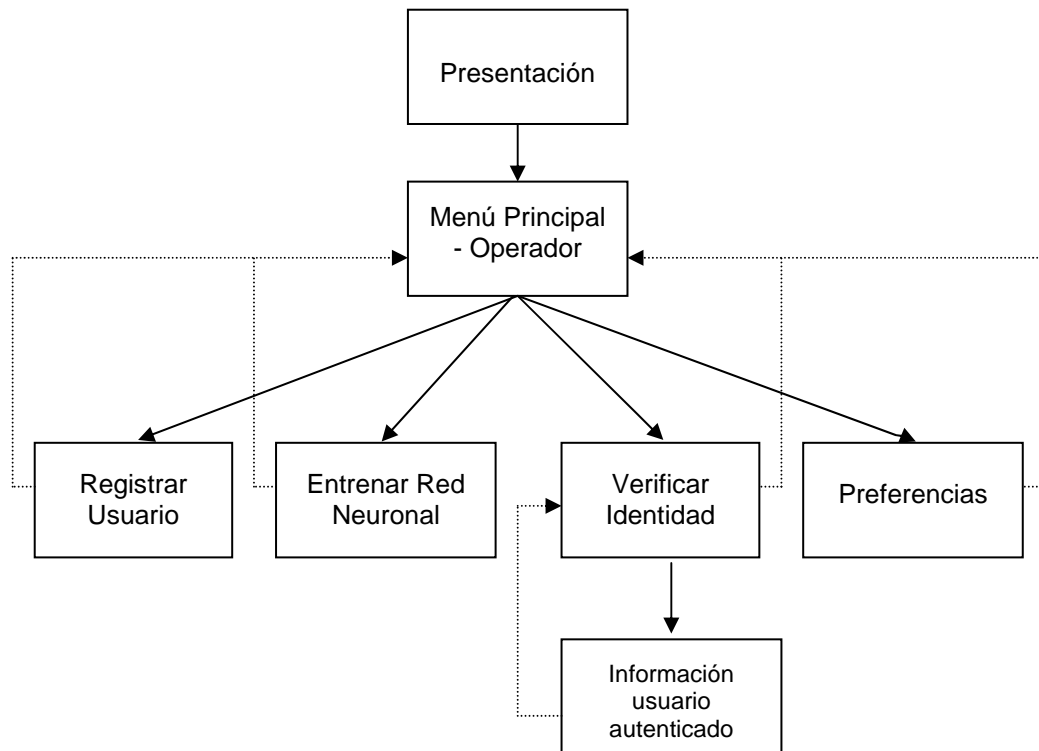


Figura 2.17. Diagrama de jerarquía de pantallas (Opciones disponibles para el Operador)

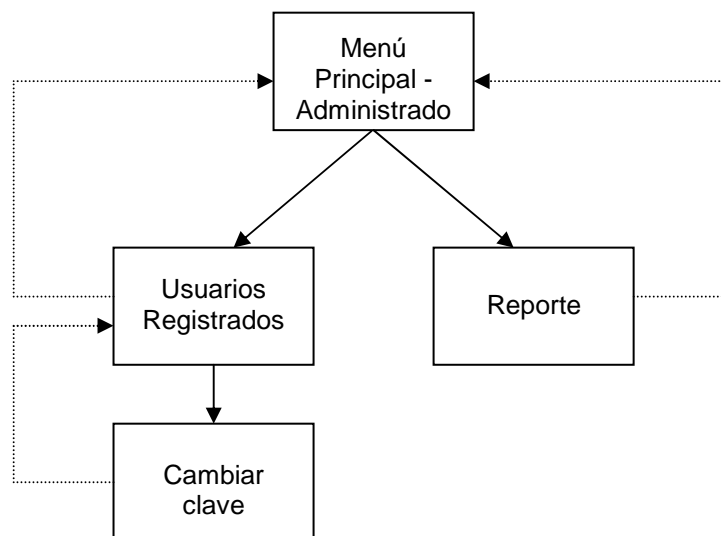


Figura 2.18. Diagrama de jerarquía de pantallas (Opciones disponibles - Administrador)

2.3.3.2 Descripción de las características generales de cada pantalla (GUI)

El Anexo 17 contiene la descripción de las diferentes interfaces gráficas que componen la solución computacional.

2.3.4 DISEÑO FÍSICO DE DATOS

Los métodos seleccionados para almacenar la información generada dentro de la solución computacional son dos y se definieron considerando los requerimientos y restricciones preestablecidas, así como también el entorno de desarrollo escogido. La información referente a los usuarios (datos e imágenes) y los parámetros de creación y entrenamiento de la red neuronal serán almacenados en una base de datos relacional (provista por Microsoft Access).

En tanto que la red neuronal, el espacio de rostros y el rostro promedio, debido a su complejidad de estructura serán almacenados por separado en disco, en archivos de texto planos (tipo de archivos: ASCII), por requerir poco espacio, ser fácilmente portables y por ser soportados por la herramienta de desarrollo (Matlab versión 7.0).

Puesto que uno de los métodos seleccionados es una base de datos relacional, se requiere que el diagrama de clases correspondiente a la capa de datos sea referenciado a un diagrama de relación de entidades (sólo para las clases Usuario, Imagen, Subespacio y Preferencias).

En el Anexo 19 se define la estructura lógica y física de datos a partir del modelo de clases, donde las clases persistentes antes mencionadas y sus atributos hacen referencia directamente a las entidades lógicas y a sus atributos considerando las relaciones existentes. En dicho anexo también se detallan los atributos y tipos de datos físicos.

CAPITULO 3. IMPLEMENTACIÓN, PRUEBAS Y EVALUACIÓN

Este capítulo describe aspectos relacionados a la implementación, pruebas y evaluación de la solución computacional.

La primera sección refiere algunas características relevantes de la herramienta de desarrollo empleada, con el propósito de explicar ciertas ventajas derivadas de la selección de dicha herramienta. La sección 3.2 detalla la forma cómo las clases de diseño se implementan en términos de componentes, considerando el entorno de desarrollo seleccionado. Finalmente, las secciones 3.3 y 3.4 definen y detallan las pruebas efectuadas y la evaluación realizada a la solución computacional.

3.1 LA HERRAMIENTA MATLAB

Matlab es un entorno de computación y desarrollo de aplicaciones completamente integrado, orientado a soluciones que requieran el manejo de diferentes herramientas a la vez, como por ejemplo recurrir a complejos cálculos matemáticos y también a la visualización gráfica de los mismos.

Matlab integra análisis numérico, cálculo matricial, procesamiento de señales e imágenes, etc. Estas y otras funciones forman parte de los toolboxes que extienden las capacidades de este entorno de desarrollo, donde las soluciones son expresadas de la misma manera en que se escribirían racionalmente y no de la forma en la que lo hace la programación tradicional, gracias al lenguaje flexible que provee.

Actualmente los toolboxes con los que cuenta Matlab permiten el desarrollo de soluciones en diversas áreas de la ingeniería y la simulación, entre los de mayor relevancia se encuentran el de procesamiento de imágenes, señales, estadística, análisis financiero, matemáticas simbólicas, redes neuronales, lógica difusa, simulación de sistemas dinámicos, entre otros.

Como se puede notar, Matlab es un entorno de cálculo técnico con grandes capacidades en computación y visualización numérica, ideal para soluciones complejas en las que generalmente se encuentran inmersos aspectos de diferentes áreas de la ingeniería y la simulación. Otra de sus características importantes, es su capacidad de ejecución en múltiples plataformas y su arquitectura abierta que permite crear herramientas personalizadas.

Para el desarrollo de la presente solución computacional se analizó especialmente las capacidades del entorno base y de los toolboxes de procesamiento de imágenes y redes neuronales. En cuanto al toolbox de procesamiento de imágenes, éste ofrece una gran variedad de operaciones que permiten fácilmente manipular, procesar y analizar datos de imágenes, presentar en pantalla imágenes 2-D y 3-D. Debido a la orientación basada en matrices del lenguaje de Matlab, las imágenes son representadas como matrices, lo que permite realizar de forma sencilla e intuitiva operaciones como filtrado de imágenes, transformadas, morfología binaria, manipulación geométrica, conversión entre espacios de color, análisis de componentes conectados, entre otras operaciones orientadas al procesamiento de imágenes. Por su parte el toolbox de redes neuronales proporciona operaciones para el diseño, inicialización, entrenamiento y simulación de redes neuronales de tipo perceptrón, redes lineales, de retropropagación, de base radial, redes de Hopfield, entre otras.

Uno de los aspectos de esta herramienta que debe ser considerado para la implementación, es la capacidad de su compilador, que en su última versión (4.0) la misma que está incluida en el entorno de Matlab 7.0 (7.0.0.19920 - R14) versión empleada para llevar a cabo el desarrollo, no soporta la compilación de ciertas funciones por ser consideradas de *tiempo de diseño*, dentro de las cuales se encuentran las funciones de entrenamiento de redes. Por lo tanto la solución computacional no contará con componentes de despliegue creados en Matlab, puesto que esta solución ofrece como parte de su funcionalidad la creación y entrenamiento de una red neuronal a partir de las imágenes proporcionadas.

3.2 IMPLEMENTACIÓN DEL SISTEMA USANDO MATLAB

Para la implementación de la solución computacional se utilizó como Front-End Matlab versión 7.0, Microsoft Visual Basic 6.0 para el desarrollo de un componente de despliegue requerido para autenticar a los usuarios con perfil de Administrador, y como Back-End Microsoft Access 2000.

3.2.1 IDENTIFICACIÓN DE COMPONENTES

El Anexo 20 presenta la implementación de los componentes a través de las clases del modelo de diseño.

3.2.2 DIAGRAMAS DE COMPONENTES

En el Anexo 21 se muestran todos los diagramas de componentes para los subsistemas de implementación, los mismos que tienen una traza uno a uno (isomófica) con los subsistemas de diseño. A continuación se presentan algunos diagramas de componentes representativos para la solución computacional y un diagrama general.

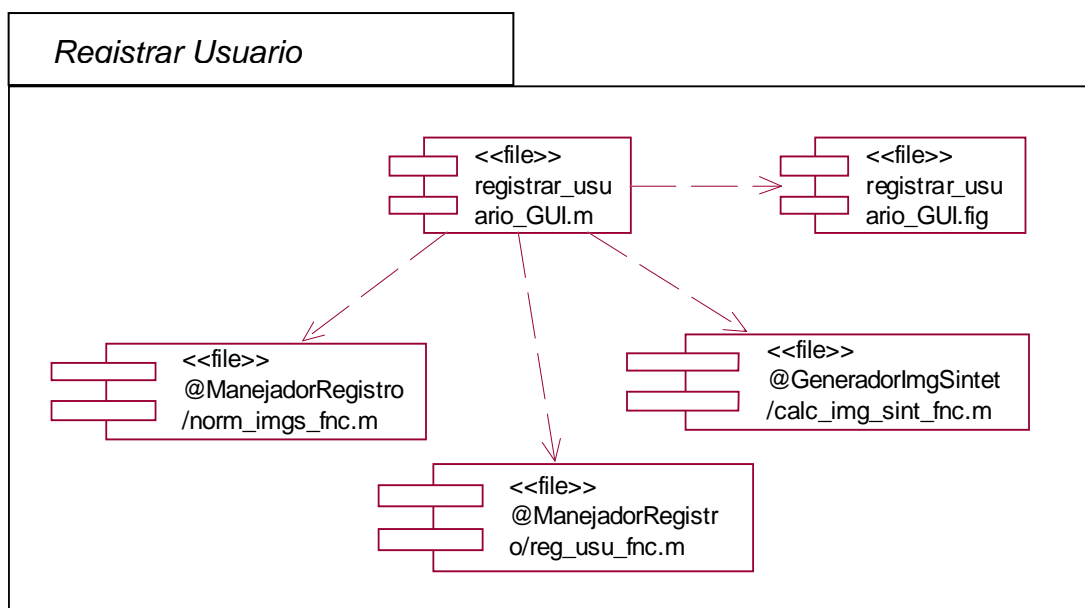


Figura 3.1. Subsistema Registrar Usuario

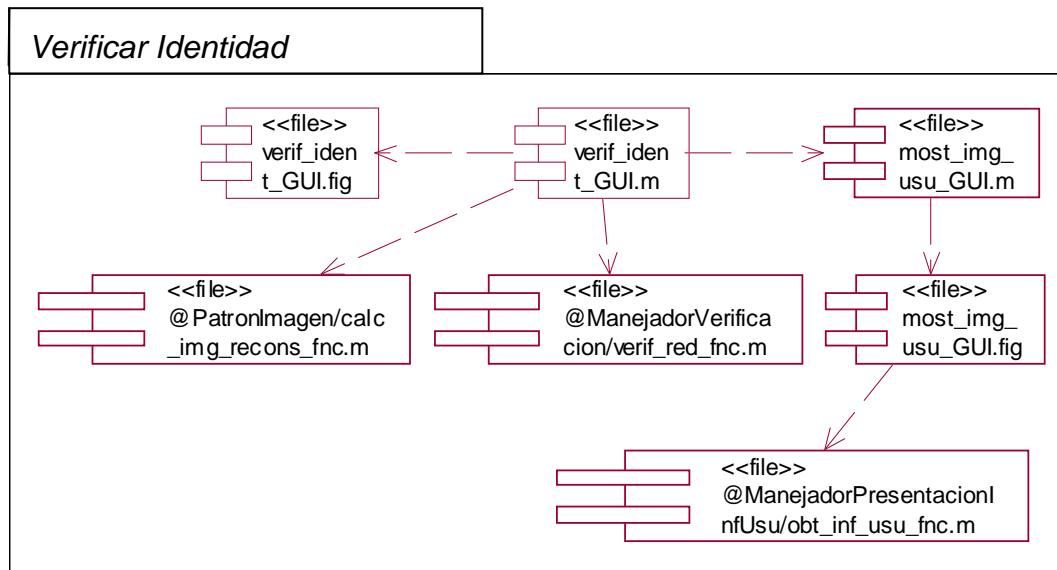


Figura 3.2. Subsistema Verificar Identidad

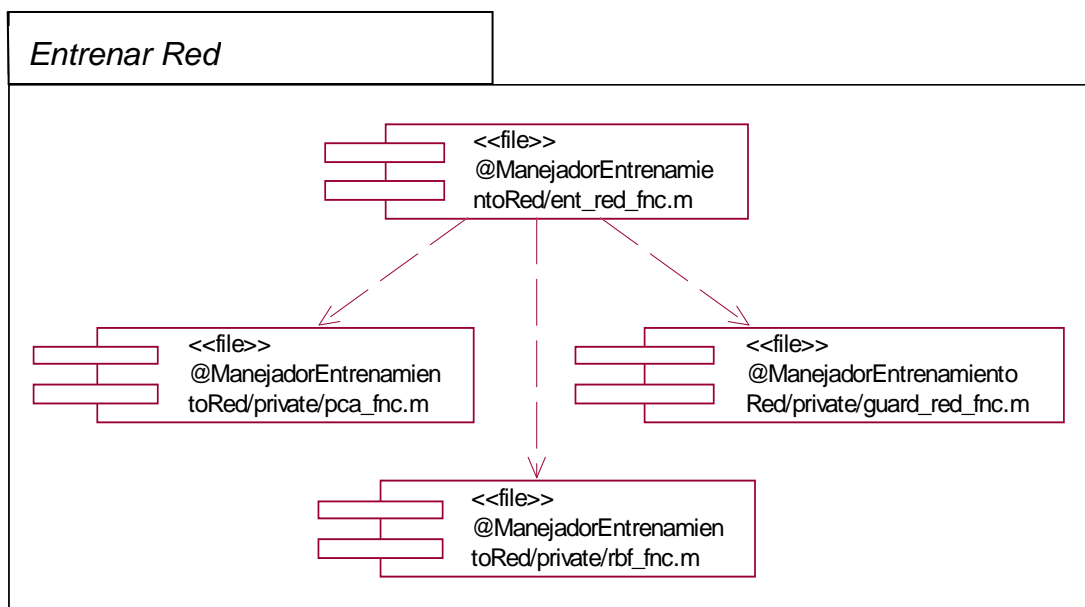


Figura 3.3. Subsistema Entrenar Red

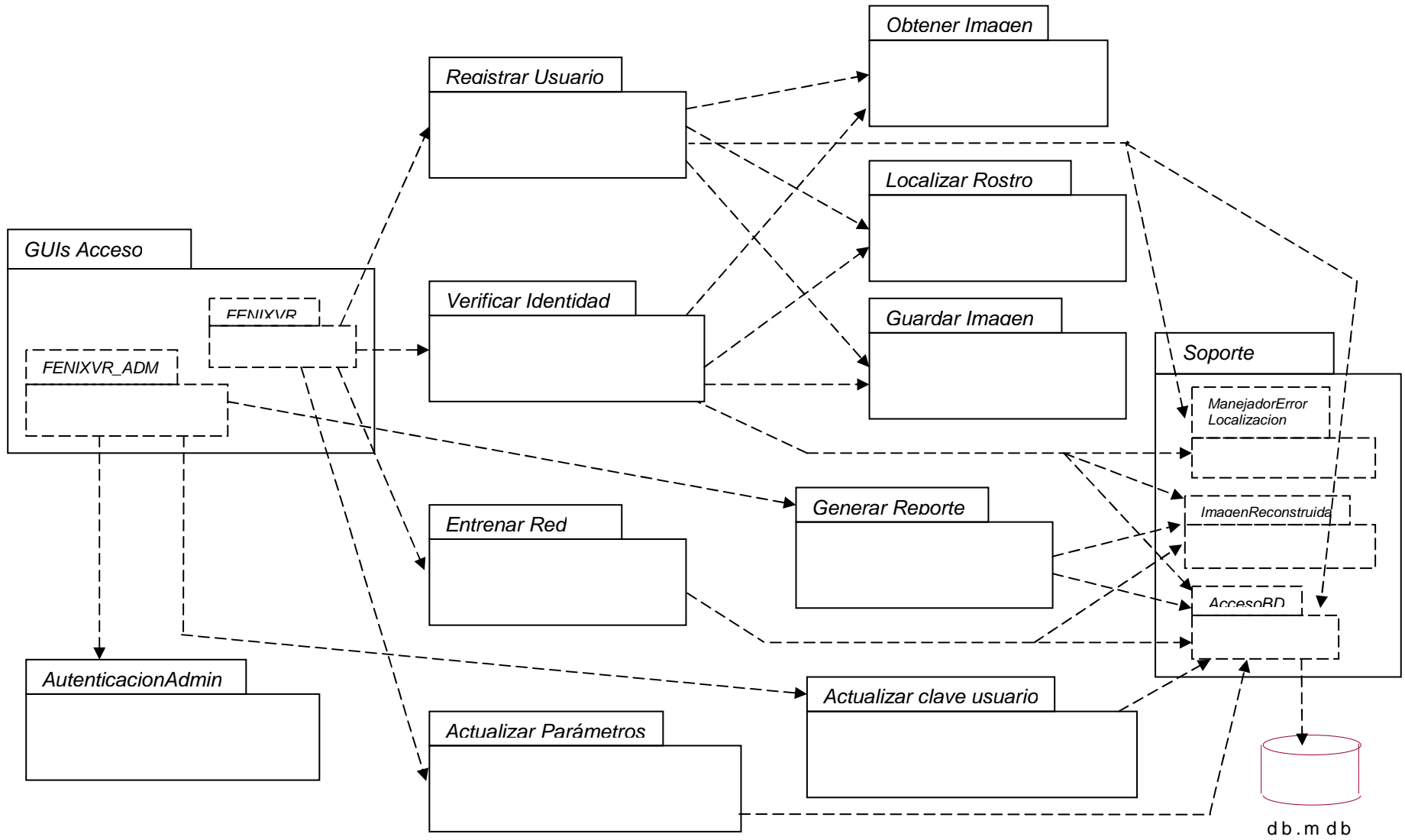


Figura 3.4. Diagrama de componentes – Subsistemas de implementación

3.3 PRUEBAS

En esta fase se procede a comprobar si la solución computacional cumple los requerimientos planteados en etapas previas. Para esto se debe definir el modelo de pruebas, el mismo que está compuesto de casos de pruebas, procedimientos y componentes de prueba.

Un caso de prueba especifica una forma de probar la solución computacional, definiendo las entradas que requiere la prueba, las condiciones bajo las que se ha de ejecutar y los resultados esperados. Los procedimientos especifican cómo realizar los casos de prueba mientras que los componentes de prueba automatizan a los procedimientos.

Los diferentes casos de prueba deben ser planteados en el contexto del tipo de prueba a ser ejecutada, por lo tanto se deben establecer los tipos de pruebas a realizarse.

En el presente proyecto se considera conveniente realizar pruebas de unidad, de integración y pruebas globales de la solución computacional; entre las características que no serán probadas se encuentran las aplicaciones y herramientas no desarrolladas, como por ejemplo el Sistema Operativo, motor base y toolboxes de Matlab 7.0 y el SGBD.

3.3.1 PRUEBAS DE UNIDAD

Consisten en probar cada clase de forma independiente con el objetivo de demostrar que se ajustan a la funcionalidad establecida. No todas las clases deben ser probadas de esta manera, puesto que muchas de ellas necesitarían de colaboraciones de otras clases durante la ejecución de dichas pruebas, por lo tanto se las analiza en conjunto durante las pruebas globales (de sistema). En el Anexo 22 se describe en detalle los casos de prueba a realizarse en esta fase.

3.3.2 PRUEBAS DE INTEGRACIÓN Y DE SISTEMA

Las pruebas de integración tienen como finalidad analizar el funcionamiento de los componentes y subsistemas de la solución computacional cuando éstos trabajan integrados para proveer uno o varios servicios. Por tal razón las pruebas de integración se realizan sobre los casos de uso implementados, para esto se cuenta con una base de datos cuya estructura es la especificada durante el diseño y con un conjunto de imágenes de rostros de diferentes personas (25 personas).

En el Anexo 23 se detallan los casos de prueba asociados a los diferentes casos de uso.

En cuanto a las pruebas de sistema su objetivo es comprobar la integración de todos los componentes y subsistemas que forman parte de la solución computacional, así como la interacción de ésta con elementos externos con los que se relaciona, en este caso el dispositivo de captura (cámara web).

Por lo tanto las pruebas antes definidas para los diferentes casos de uso implementados, permiten en conjunto probar todos los servicios provistos por la solución computacional, a fin de establecer si dicha solución cumple todos los requerimientos planteados.

3.3.3 REALIZACIÓN Y EVALUACIÓN DE PRUEBAS

3.3.3.1 Pruebas de Unidad

Se analizan y evalúan los resultados de las pruebas de unidad para comprobar que los resultados obtenidos son los esperados.

A continuación, en la Tabla 3.1 se presentan los resultados de las pruebas de unidad efectuadas.

Caso de Prueba	Resultado Esperado	Resultado Obtenido
CP-001	Debe desplegarse la ventana de registro.	Se despliega la ventana de registro.
CP-002	Debe presentarse un mensaje que confirme el inicio del entrenamiento de la red neuronal.	Mensaje: Confirmar inicio entrenamiento red neuronal.
CP-003	Debe presentarse un mensaje de error - No existen usuarios registrados.	Mensaje de error: No existen usuarios registrados.
CP-004	Debe desplegarse la ventana de verificación de identidad.	Se despliega la ventana de verificación de identidad.
CP-005	Debe presentarse un mensaje de error.	Mensaje de error.
CP-006	Debe desplegarse la ventana de preferencias.	Se despliega la ventana de preferencias.
CP-007	Deben presentarse mensajes de error que soliciten verificar formato.	Mensajes de error: Verificar formato.
CP-008	Debe iniciarse el proceso de almacenamiento.	Se inicia el proceso de almacenamiento.
CP-009	Debe presentarse un mensaje de error - Clave de usuario no válida.	Mensaje de error: Clave de usuario no válida.
CP-010	Debe obtenerse la imagen desde la ruta especificada y también sus dimensiones.	Se obtiene la imagen desde la ruta especificada y se obtienen sus dimensiones.
CP-011	Debe presentarse un mensaje de error - Ruta no válida.	Mensaje de error: Ruta no válida.
CP-012	Debe obtenerse una imagen a través de la cámara web y también las dimensiones de dicha imagen.	Se obtiene una imagen a través de la cámara web y se obtienen las dimensiones de dicha imagen.
CP-013	Debe presentarse un mensaje de error - No se ha detectado dispositivo.	Mensaje de error: No se ha detectado dispositivo.
CP-014	Debe iniciarse el proceso de verificación de identidad.	Se inicia el proceso de verificación de identidad.
CP-015	Debe presentarse un mensaje de error - No existe un usuario previamente registrado con la clave ingresada.	Mensaje de error: No existe un usuario previamente registrado con la clave ingresada.
CP-016	Debe presentarse un mensaje de error - Verificar formato de la clave.	Mensaje de error: Verificar formato de la clave.
CP-017	Debe desplegarse la ventana de usuarios registrados.	Se despliega la ventana de usuarios registrados.

CP-018	Debe desplegarse el reporte de usuarios registrados.	Se despliega el reporte de usuarios registrados.
CP-019	Debe iniciarse el proceso de actualización de clave.	Se inicia el proceso de actualización de clave.
CP-020	Debe presentarse un mensaje de error - Verificar formato de la nueva clave.	Mensaje de error: Verificar formato de la nueva clave.
CP-021	Debe retornar 1.	Retorna valor = 1
CP-022	Debe retornar 0.	Retorna valor = 1
CP-023	Debe retornar un objeto de la clase ManejadorDetDisp cuyo atributo disp_id_i sea el identificador asociado al dispositivo detectado (cámara web).	Retorna un objeto de la clase ManejadorDetDisp cuyo atributo disp_id_i es el identificador asociado al dispositivo detectado (cámara web).
CP-024	Debe retornar un objeto de la clase ManejadorDetDisp cuyo atributo num_adap_disp_l tenga un valor = 0.	Retorna un objeto de la clase ManejadorDetDisp cuyo atributo num_adap_disp_l tiene valor = 0.
CP-025	Debe desplegar en pantalla la imagen RGB con el área del rostro señalada y retornar un objeto de la clase LocalizacionRostro cuyos atributos representen la información requerida sobre la imagen del rostro.	Despliega en pantalla la imagen RGB con el área del rostro señalada y retorna un objeto de la clase LocalizacionRostro cuyos atributos representan la información requerida sobre la imagen del rostro.
CP-026	Debe retornar un objeto de la clase LocalizacionRostro con el atributo band_err_i <> 10.	Retorna un objeto de la clase LocalizacionRostro en el que el atributo band_err_i es <> 10.
CP-027	Debe retornar un objeto de la clase LocalizacionRostro con el atributo band_err_i <> 10.	Retorna un objeto de la clase LocalizacionRostro en el que el atributo band_err_i es <> 10.
CP-028	Debe haberse almacenado en disco el archivo que contiene a la red neuronal, el archivo del eigenspacio y el archivo referente al rostro promedio.	Se ha almacenado en disco el archivo que contiene a la red neuronal, el archivo del eigenspacio y el archivo referente al rostro promedio.
CP-029	Debe haberse almacenado en disco el archivo que contiene a la red neuronal, el archivo del eigenspacio y el archivo referente al rostro promedio.	Se ha almacenado en disco el archivo que contiene a la red neuronal, el archivo del eigenspacio y el archivo referente al rostro promedio.
CP-030	Debe retornar 1.	Retorna un valor = 1
CP-031	Debe retornar 0.	Retorna un valor = 0

Tabla 3.1. Resultado de la realización de los casos de prueba de unidad

3.3.3.2 Pruebas de Integración y de Sistema

A continuación, en la Tabla 3.2 se presentan los resultados de las pruebas efectuadas a nivel global, con los casos de prueba definidos.

Caso de Prueba	Caso de uso	Resultado Esperado	Resultado Obtenido
CP-032	Registrar Usuario	Usuario registrado (información almacenada)	Usuario registrado (información almacenada)
CP-033	Registrar Usuario	Usuario registrado (información almacenada)	Usuario registrado (información almacenada)
CP-034	Registrar Usuario	Mensaje de error. No se localizó un rostro en la imagen. Botón <i>Registrar Usuario</i> deshabilitado.	Mensaje de error. No se localizó un rostro en la imagen. Botón <i>Registrar Usuario</i> deshabilitado.
CP-035	Verificar Identidad	Mensaje : Acceso Concedido. Se presenta una ventana con toda la información almacenada del usuario autenticado.	Mensaje : Acceso Concedido. Se presenta una ventana con toda la información almacenada del usuario autenticado.
CP-036	Verificar Identidad	Mensaje: Acceso Denegado.	Mensaje: Acceso Denegado.
CP-037	Verificar Identidad	Mensaje: Acceso Concedido. Se presenta una ventana con toda la información almacenada del usuario autenticado.	Mensaje: Acceso Concedido. Se presenta una ventana con toda la información almacenada del usuario autenticado.
CP-038	Verificar Identidad	Mensaje de error: No se localizó un rostro en la imagen. Botón <i>Verificar Identidad</i> deshabilitado.	Mensaje de error: No se localizó un rostro en la imagen. Botón <i>Verificar Identidad</i> deshabilitado.
CP-039	Obtener Imagen	Mensaje de error: No se ha detectado dispositivo.	Mensaje de error: No se ha detectado dispositivo.
CP-040	Comprobar Perfil para acceso	Se presenta la ventana del menú de administración.	Se presenta la ventana del menú de administración.

CP-041		Acceso denegado.	Acceso denegado.
CP-042	Actualizar Parámetros de la Red Neuronal	Se han actualizado los valores.	Se han actualizado los valores.
CP-043	Actualizar Parámetros de la Red Neuronal	Mensaje de error: Verificar valores ingresados.	Mensaje de error: Verificar valores ingresados.
CP-044	Guardar Imagen Inicial	Se ha creado una copia de la imagen inicial en la ubicación especificada.	Se ha creado una copia de la imagen inicial en la ubicación especificada.
CP-045	Guardar Imagen Rostro Localizado	Se ha creado una copia de la imagen con el rostro localizado en la ubicación especificada.	Se ha creado una copia de la imagen con el rostro localizado en la ubicación especificada.
CP-046	Actualizar Claves de Usuario	Se ha actualizado la clave	Se ha actualizado la clave
CP-047	Actualizar Claves de Usuario	Mensaje: Error en la clave nueva o en la confirmación (no coinciden).	Mensaje: Error en la clave nueva o en la confirmación (no coinciden).
CP-048	Actualizar Claves de Usuario	Mensaje de error: Verificar el formato de la nueva clave.	Mensaje de error: Verificar el formato de la nueva clave.
CP-049	Generar Reporte Usuarios Registrados	Reporte de usuarios registrados.	Reporte de usuarios registrados.

Tabla 3.2. Resultado de la realización de los casos de prueba de integración y de sistema

3.3.3.3 Valoración de los resultados de las pruebas

Con base en el análisis de los resultados de los casos de prueba descritos en las Tablas 3.1 y 3.2 se puede concluir que la solución computacional cumple los requerimientos definidos durante etapas previas puesto que los resultados obtenidos son los esperados.

3.4 EVALUACIÓN

La evaluación de la solución computacional se llevó a cabo en dos fases. En la primera fase se evaluó el subsistema para la localización de un rostro en una imagen dada, en tanto que en la segunda fase se evaluó el proceso de verificación de una identidad partiendo de la imagen del rostro proporcionada por el subsistema de localización.

En el Anexo 24 se describe en detalle las condiciones bajo las cuales se llevó a cabo el proceso de evaluación de la solución computacional.

A continuación se muestran los resultados obtenidos en esta etapa.

3.4.1 RESULTADOS EXPERIMENTALES

Como se mencionó en secciones anteriores, la evaluación se realizó en dos fases. A continuación se presentan los resultados experimentales obtenidos en dichas fases:

1. Evaluación del Subsistema de Localización de Rostros

Para la evaluación del subsistema de localización se empleó 154 imágenes pertenecientes a la base de rostros disponible a través de Internet. Este subconjunto de imágenes corresponde a las primeras 52 personas que forman parte de dicha base de rostros.

La utilización de un subconjunto de imágenes en lugar de la base de rostros completa, obedece únicamente a cuestiones de limitación de tiempo para llevar a cabo la evaluación del subsistema.

Cada imagen del subconjunto fue cargada dentro de la solución computacional y analizada con el objetivo de localizar un rostro. El subsistema encontró correctamente 146 rostros (uno por cada imagen

presentada), es decir realizó una correcta localización en el 95% del total de las imágenes. Del porcentaje restante, el 1% se refiere a localizaciones erróneas, lo que significa que si bien el subsistema localizó el área del rostro falló en la localización del área de los ojos o de la boca; el 4% final se refiere a imágenes en las que el subsistema no localizó un rostro. El tiempo promedio de localización de un rostro para este subconjunto de imágenes está alrededor de los 15 segundos.

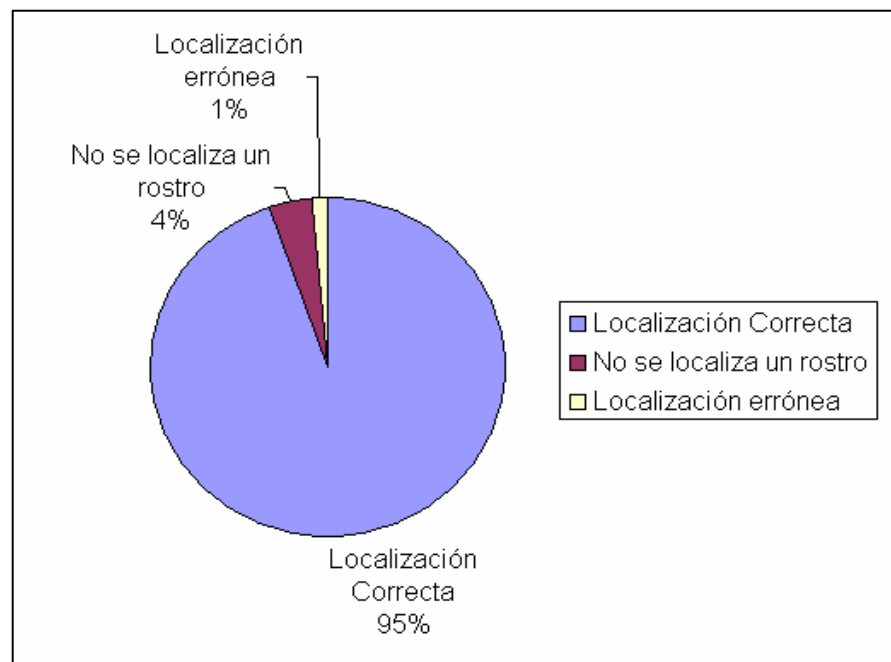


Figura 3.5. Porcentaje de aciertos y fallos en la localización de rostros

Adicionalmente se realizaron pruebas con imágenes obtenidas desde Internet e imágenes adquiridas con la cámara web. En este caso no se presentan resultados relacionados con porcentajes de aciertos y fallos en la localización, puesto que las imágenes no tienen un formato estándar y no representan una muestra de tamaño adecuado, sin embargo se presentan algunos de los resultados alcanzados (ejemplos representativos) con el objetivo de demostrar que la solución computacional es bastante robusta en cuanto a rotación lateral del rostro así como también a distintos tipos de color de piel, siempre y cuando se cuente con niveles de iluminación adecuada.

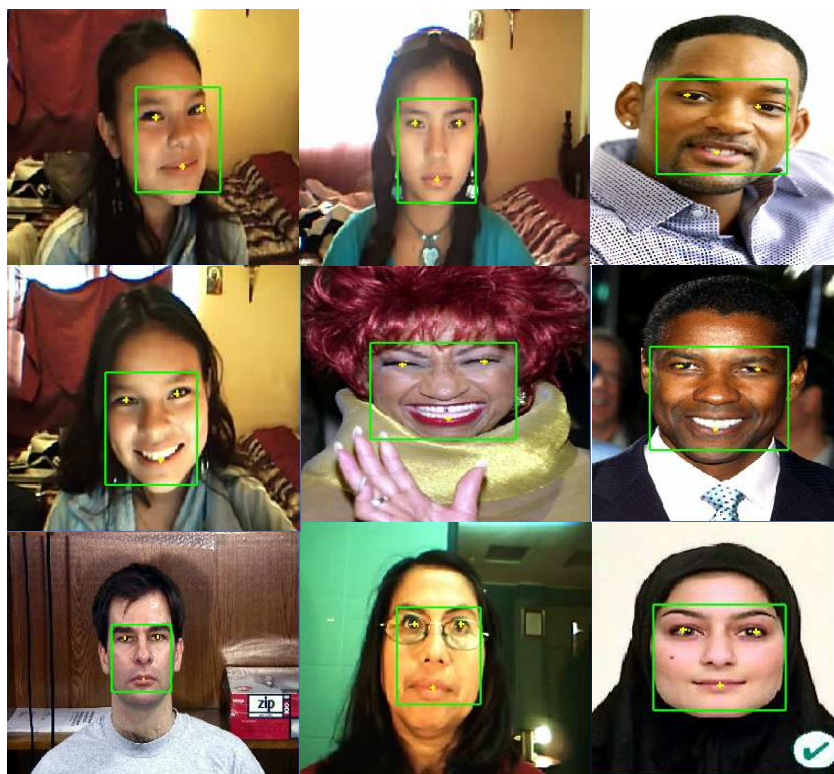


Figura 3.6. Ejemplos de localización correcta del área del rostro



Figura 3.7. Ejemplos de localizaciones erróneas y localizaciones fallidas (no se localizó el rostro presente en la imagen)

Una vez concluida la evaluación del subsistema se procedió a realizar la creación de la base de rostros FENIXVR, a través de la propia interfaz de la solución computacional. Se adquirió un total de 390 imágenes de las cuales en 379 (más del 97%) fue localizado correctamente el rostro presente en las mismas y en 11 imágenes (cerca del 3%) falló la localización de los rostros o existió una localización incorrecta. El tiempo promedio empleado para la localización de rostros en estas imágenes fue de 3.7 segundos

Al comparar los promedios de los tiempos requeridos para la localización de rostros en las imágenes de la base FENIXVR y en las imágenes de la base disponible a través de Internet, se puede notar que el tamaño de las imágenes impacta directamente en la cantidad de tiempo requerido para llevar a cabo la tarea de localización. Mientras en imágenes de tamaño 320x240 el tiempo promedio está alrededor de 3.7 segundos, en imágenes de tamaño superior (el doble del tamaño, en este caso) 640x480, el tiempo promedio está alrededor de los 15 segundos.

2. Evaluación de la Verificación de Identidad

Se realizó una evaluación de escenario, que tiene como fin medir el desempeño de la solución computacional para un escenario prototipo que modela el dominio de esta solución. Dicho escenario consistió en la verificación de identidad para el conjunto de personas que forman parte de la base de rostros FENIXVR.

Para medir el desempeño de la solución con relación a la verificación de identidades, se empleó la técnica de *validación cruzada*. Esta técnica consiste en dividir al conjunto de imágenes disponible (base de rostros FENIXVR) en dos subconjuntos independientes entre sí, de tal manera que un grupo de imágenes forme el conjunto de entrenamiento y las imágenes restantes formen el conjunto de prueba. El objetivo de esta técnica es que la calidad del desempeño de la solución para el conjunto de prueba, refleje en cierto nivel, cuan bueno puede ser el desempeño de la solución cuando se le proporcionen nuevas imágenes de los usuarios registrados. Por esta razón se eligió esta técnica, a pesar de proporcionar una estimación bastante pesimista con relación a otras técnicas de evaluación existentes, muchas de las cuales son empleadas en investigaciones publicadas en la literatura.

Aplicando la técnica antes descrita, se obtuvo dos conjuntos de imágenes (véase Anexo 24). Con el conjunto de entrenamiento se procedió a

registrar, a través de la solución computacional, los datos e imágenes (5 imágenes) de cada persona de la base FENIXVR, a excepción de las personas asignadas con los números 5, 9 y 12 (véase Anexo 24), puesto que estas personas forman parte del conjunto de pruebas como usuarios desconocidos. Una vez finalizado el registro de los usuarios se realizó el entrenamiento de la red neuronal. Esto a su vez requiere la extracción de características de los rostros de los diferentes usuarios registrados.

Luego de varias pruebas para determinar la manera más adecuada de seleccionar el número de eigenvectores a utilizar en la creación del espacio de rostros, se concluyó que el método de *dimensión de la energía* es el que mejor resultados provee. Este método usa el mínimo número de eigenvectores para garantizar que la energía es mayor a un valor umbral, en este caso mayor al 95% del total de energía.

Finalmente con el conjunto de vectores de pesos, asociados a los diferentes usuarios, se procedió a crear y entrenar a una red neuronal artificial de tipo RBF. De acuerdo a la herramienta de desarrollo empleada (Matlab 7.0) se deben definir algunos parámetros como por ejemplo el error mínimo cuadrático estándar, el número de neuronas añadidas por desplazamiento y la desviación estándar o área de influencia de cada neurona.

Todos estos parámetros fueron definidos después de una serie de pruebas, ensayando con diferentes valores de los parámetros e interpretando y comparando los resultados obtenidos. De esta manera se definieron los siguientes valores:

- error mínimo cuadrático estándar = 0.8
- número de neuronas añadidas por desplazamiento = 2
- la desviación estándar es calculada en tiempo de ejecución como el promedio de las desviaciones estándar de cada conjunto de imágenes que pertenece a un usuario.

Cuando se contó con una red entrenada y almacenada en disco se inició el proceso de verificación empleando el conjunto de imágenes de entrenamiento previamente creado. Finalizada la tarea de verificación para todas las imágenes del conjunto de entrenamiento se analizó y evaluó los resultados obtenidos.

Utilizando todos los elementos antes descritos, la solución computacional desarrollada alcanzó un 81% de verificaciones correctas, de las 269 imágenes que componen el conjunto de entrenamiento. Del porcentaje restante, el 16% representa la tasa de rechazos falsos, es decir un rostro conocido no fue identificado, mientras que el 3% representa la tasa de reconocimientos falsos, es decir un rostro fue identificado como conocido sin serlo.

La Figura 3.8 muestra los porcentajes obtenidos durante esta fase de evaluación.

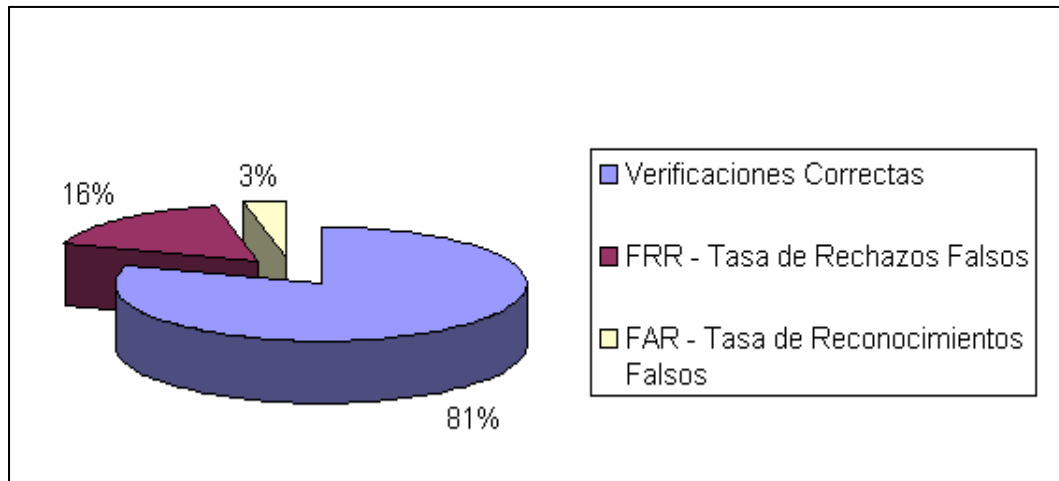


Figura 3.8. Porcentajes obtenidos para la verificación de rostros

Es importante destacar que la solución computacional presenta un bajo porcentaje de FRR, es decir que el número de usuarios identificados como conocidos sin serlo es mínimo. Esta característica cubre uno de los requerimientos establecidos en etapas previas. Adicionalmente se debe mencionar que dentro del conjunto de prueba, existen algunas imágenes

que no presentan condiciones de iluminación constante ni similar a la mayoría de imágenes que forman parte de este conjunto.

A continuación se muestran algunos ejemplos de imágenes que forman parte de las tasas FRR o FAR.

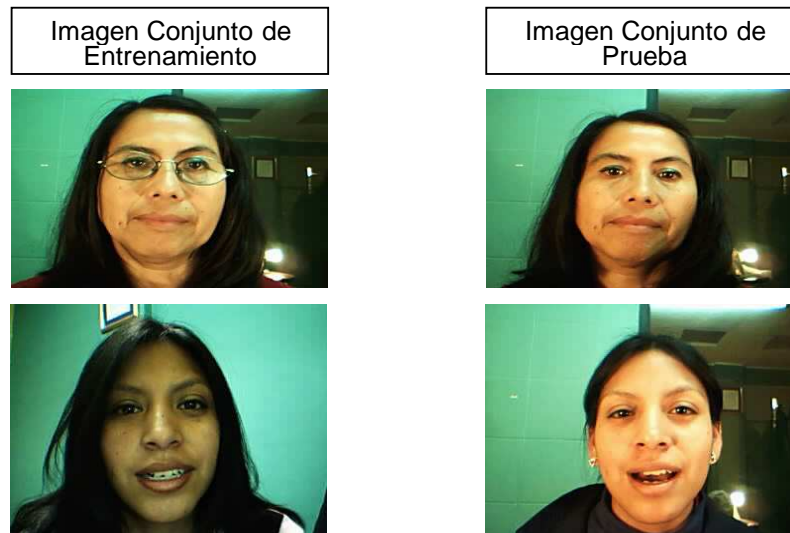


Figura 3.9. Imágenes que forman parte de las tasas FRR y FAR

La Figura 3.10 muestra ejemplos de verificaciones correctas:

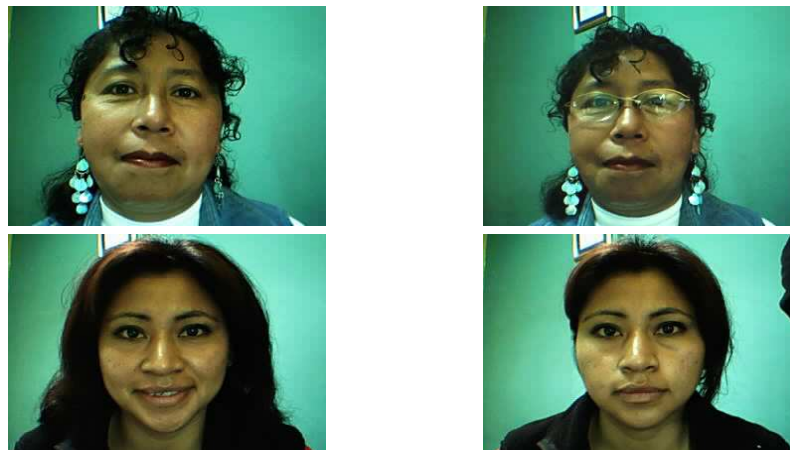


Figura 3.10. Imágenes que forman parte de las verificaciones correctas

El tiempo promedio empleado para la verificación de identidad de un usuario fue de 0.38 segundos. Este tiempo sólo se refiere a la tarea de

verificación y no incluye el tiempo empleado para la localización del rostro en la imagen.

De los resultados obtenidos en la evaluación de verificación de identidad se pueden establecer las siguientes conclusiones:

- Los cambios en iluminación pueden afectar la verificación de identidad
- Para la arquitectura de red empleada se notó que en el caso de aquellas personas que utilizan lentes, el proporcionar imágenes de entrenamiento de estas personas sin la presencia de lentes permite obtener mejores resultados durante la verificación de identidad cuando se presentan imágenes de prueba en las que las personas están utilizando lentes. Mientras que proporcionar imágenes de entrenamiento de las personas sólo con la presencia de lentes dificulta la verificación cuando se presentan imágenes de prueba de dichas personas sin la presencia de lentes.
- Los cambios en el estilo de peinado no influyen en la verificación de identidad, esto se debe a que el área del rostro empleada no contiene las zonas del cabello ni de cualquier fondo presente en la imagen inicial.

Una vez finalizadas las fases de evaluación, se puede concluir que la solución computacional desarrollada cumple con los requerimientos establecidos, logrando un 81% de verificaciones correctas bajo la técnica de validación cruzada, con un tiempo promedio total de 4.08 segundos para llevar a cabo la verificación de identidad de un usuario (tiempo empleado desde que la imagen es obtenida y se solicita la localización de un rostro en la misma). Sin embargo se debe destacar que este tiempo total está en función del tamaño de las imágenes empleadas, que en este caso es de 320x240. Si se emplean imágenes de mayor tamaño el tiempo total requerido será mayor como se vio durante la fase de evaluación del subsistema de localización de rostros (15.38 segundos para imágenes de tamaño 640x480), por lo que se recomienda emplear imágenes del tamaño de las utilizadas durante esta evaluación, a fin de obtener resultados en tiempo cercano al tiempo real.

CAPITULO 4. CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- El marco de trabajo empleado para el desarrollo de la solución computacional permitió establecer un esquema flexible de iteraciones cortas a fin de planificar y comprobar el progreso parcial y total del desarrollo de la solución, contando siempre con la retroalimentación proveniente de iteraciones anteriores.
- El empleo de UML como lenguaje de modelado permitió mantener la trazabilidad de los diferentes elementos creados y refinados en las diversas etapas de desarrollo, teniendo siempre como base los casos de uso, que se refinaron a lo largo del proceso.
- Cualquier solución está orientada a las necesidades del usuario final, sin embargo existen procesos intermedios, para alcanzar satisfacer dichas necesidades, que no tienen relevancia para los usuarios. En el caso de la presente solución computacional las tareas relacionadas con la localización de rostros son internas y no se encuentran visibles para el usuario final, por lo que la forma cómo se realizan dichas tareas depende de cuestiones de diseño, de optimización y de otros factores que deben ser considerados por quien desarrolla una solución de este tipo. Por lo tanto, antes de plantear un esquema de solución se debe tener claro el conjunto de conceptos inmerso en el dominio del problema.
- En el área de procesamiento de imágenes generalmente no existen técnicas con mejor desempeño que otras, ya que esto depende del entorno y del objetivo para el que se emplee una técnica. Por ejemplo, la ecualización del histograma de una imagen puede resultar adecuado para imágenes satelitales, pero puede provocar pérdida de información en imágenes radiológicas.

- Actualmente no existe una arquitectura de red neuronal que sea adecuada para una tarea específica como es el caso de la verificación de rostros, considerando diversas condiciones de iluminación, rotación, apariencia, expresiones faciales, etc. Por tal razón, la selección de una u otra arquitectura no deja de ser un problema de prueba y error, donde se invierte gran cantidad de tiempo probando diferentes arquitecturas y formas de entrenamiento.
- De la evaluación que se realizó a la solución computacional se puede decir que el desempeño de la misma depende de las características de la base de rostros empleada, tanto del conjunto de entrenamiento como del conjunto de prueba.
- El mayor problema encontrado en el desarrollo de esta solución es el contar con un área para la adquisición de imágenes, en la que las condiciones de iluminación sean adecuadas de forma constante para así obtener una base de rostros completamente homogénea. Ya que como se analizó durante la fase de evaluación, la variación en iluminación sí es un factor decisivo en los resultados de la verificación de identidades.
- El número de usuarios registrados y el número de imágenes por cada usuario también constituye un factor que influye en los resultados de la verificación de identidad. A un mayor número de usuarios registrados, existe una mayor probabilidad que la red neuronal cometa más errores en la verificación, en tanto que a un mayor número de imágenes por usuarios (imágenes lo más diferente posible entre sí) se mejoran las condiciones para que la red realice una mejor tarea de verificación.
- Se debe establecer un equilibrio en el desempeño de la solución en lo referente a las tasas FFR y FAR, ya que si se establecen valores de referencia muy altos, la solución será muy rígida y si bien reducirá el número de reconocimientos falsos también puede aumentar el número de

rechazos falsos. De la misma manera, valores muy bajos provocarán una solución demasiado flexible.

- Contar con la ayuda en línea permite a los usuarios de la solución computacional consultar en cualquier momento las características que deben tener las imágenes con las que se va a trabajar. Esto puede reducir problemas de localización de rostros derivados de presentar imágenes no adecuadas.

4.2 RECOMENDACIONES

- Se recomienda incentivar la investigación y desarrollo de aplicaciones orientadas al campo de visión por computador. Puesto que esta es un área que actualmente no ha sido explotada en su totalidad, tiene muchas aplicaciones comerciales, y las tecnologías asociadas a este campo así como al de la biometría muestran un gran crecimiento para los próximos años.
- Se puede emplear la solución computacional desarrollada como base para una serie de experimentos con vistas a incrementar el desempeño de los algoritmos empleados tanto en el procesamiento de las imágenes como en la verificación de identidades con ayuda de la red neuronal. Así mismo incorporando otras soluciones biométricas, por ejemplo verificación con huellas dactilares o reconocimiento de voz, se puede obtener una solución mucho más robusta, que en un futuro puede ser implantada donde se requiera.
- Contar con material como libros, revistas o documentos en línea actualizados, sobre los diferentes avances que se realizan diariamente en diversos campos de las tecnologías de la información, permite tener un espectro más amplio no sólo de lo que actualmente se está desarrollando sino que también sirve como base para nuevas soluciones.

BIBLIOGRAFÍA

LIBROS

- [1] GONZALEZ, R. C.; WOODS, R. E. Tratamiento Digital de Imágenes. Editorial Addison-Wesley. Washington - EUA. 1996.
- [2] PARKER, J. R. Algorithms for image processing and Computer Vision. Editorial John Wiley & Sons. EUA. FALTA AÑO
- [3] PRATT, W.K. Digital Image Processing. Editorial John Wiley & Sons. 1991.

PROYECTOS DE TITULACIÓN

- [4] BASANTES, Fabián W. Procesamiento digital de imágenes utilizando la transformada de Hough. Proyecto de titulación (eléctronica). 2004.
- [5] CERÓN, Gabriel; OVIEDO, Mónica. Herramienta de Generación de Rostros para ser usada en la Web que permite una manipulación realista de características faciales para aplicaciones policiales. Proyecto de titulación (sistemas). 2002.
- [6] CHANATASIG, Marco; FIGUEROA, Alex. Sistema para la identificación Biométrica de Huellas Dactilares. Proyecto de titulación (sistemas). 2000.
- [7] CHICO, Catalina. Diseño de un mapa de Hábitat Faunística en Oyacachi Papallacta utilizando Técnicas de Procesamiento de Imágenes. Proyecto de titulación (sistemas). 2003.
- [8] MARTÍNEZ, Gonzalo. Estudio de diferentes Métodos de Extracción de Características de Imágenes Digitales y su aplicación en el Reconocimiento de Patrones. Proyecto de titulación (eléctrica). 1999.
- [9] VEGA, Jefferson. Simulación de un Sistema de Seguridad Antirrobo basado en el Procesamiento Digital de Imágenes y Reconocimiento de Patrones. Proyecto de titulación (eléctrica). 1997.

ARTICULOS

- [10] ANIL, K.; DUIN, R.; MAO, J. Statistical Pattern Recognition: A Review. EUA. 1999.
- [11] ANIL, K. Jain; MAO, J.; MOHIUDDIN, K. Artificial Neural Networks: A Tutorial. IEEE Computer. Vol. 29. N° 3. pp.31-44. 1996.
- [12] AKSOY, Selim. Structural and Syntactic Pattern Recognition. Bilkent University, Department of Computer Engineering. 2006.

- [13] ARCA, Stefano; CAMPADELLI, Paola; LANZAROTTI, Rafaella. A face recognition system based on automatically determined facial fiducial points. Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano Via Comelico. Italia. 2005.
- [14] ASHBROOK, A.; THACKER, N.A. Tutorial: Algorithms For 2-Dimensional Object Recognition. Imaging Science and Biomedical Engineering Division, Medical School, University of Manchester. Manchester. 1998.
- [15] ATALAY, Ilker. Face Recognition Using Eigenfaces, M. Sc. Thesis. Computer Engineer B.Sc.Istanbul. 1996.
- [16] BLANZIERI, Enrico. Theoretical Interpretations and Applications of Radial Basis Function Networks. ELSEVIER. Technical Report # DIT-03-023. University of Trento, Department of Information and Communication Technology. Trento -Italia. 2003.
- [17] BRYLIUK, Dmitry; STAROVOITOV, Valery. Access Control By Face Recognition Using Neural Networks. Institute of Engineering Cybernetics, Laboratory of Image Processing and Recognition. 2002.
- [18] CAMPADELLI, Paola; LANZAROTTI, Rafaella; SAVAZZI, Chiara. A feature-based face recognition system. Dipartimento di Scienze dell'Informazione Università degli Studi di Milano Via Comelico. Milano - Italia. 2003.
- [19] CENDRILLON, Raphael. Real Time Face Recognition using Eigenfaces. Department of computer science and Electrical Engineering, University of Queensland. 1999.
- [20] CHEN, Li-Fen; MARK, Hong-Yuan; LIN, Ja-Chen; HAN, Chin-Chuan. Why Recognition in a Statistics-based Face Recognition System Should be based on the pure Face portion: a Probabilistic Decision-based Proof. Department of computer and Information Science, National Chiao Tung University. Taiwan. 1999.
- [21] CHEN, Phil; GRECOS, Christos. A Fast Skin Region Detector. Department of IEEE, Loughborough University. Division Research. 2005.
- [22] CHI MAI, Luong. Chap 4 - Segmentation And Edge Detection; Chap 5 - Morphological and other area operations. MapScan group. IOIT. 2005.
- [23] DELAKIS, M.; GARCIA, C. Robust Face Detection Based on Convolutional Neural Networks. Department of Computer Science, University of Crete. 2nd. Hellenic Conf. On AI. SETN-2002. Greece. Companion Volume, pp. 367-378. 2002.
- [24] ECABERT, Olivier; THIRAN, Jean-Philippe. Adaptive Hough transform for the detection of natural shapes under weak affine transformations. Elsevier. Pattern Recognition Letters. 2004.

- [25] FANG, Yuchun; WANG, Yunhong; TAN, Tieniu. Personalized Feature Combination for Face Recognition. National Laboratory of Pattern Recognition, Institute of Automation. Chinese Academy of Sciences. 2002.
- [26] FENG, Guo Can; YUEN, Pong C. Multi-cues eye detection on gray intensity image. Pattern Recognition Society. Publicado por Elsevier Science Ltd. 2000.
- [27] GARCIA, C.; SIMANDIRIS, G.; TZIRITAS, G. A feature-based face detector using wavelet frames. Department of computer science, University of Crete. Greece. 2002
- [28] GARCIA, Christophe; TZIRITAS, Georgios. Face Detection Using Quantized Skin Color Regions Merging and Wavelet Packet Analysis. IEEE Transactions On Multimedia. Vol. 1. N^o. 3. 1999.
- [29] GARCIA, C.; ZIKOS, G.; TZIRITAS, G. Wavelet Packet Analysis for Face Recognition. Elsevier. Institute of Computer Science, Foundation for Research and Technology-Hellas (FORTH). Crete – Greece. 1998.
- [30] GROTH, Patrick; MICHEALS, Ross; PHILLIPS, P. Jonathon. Face Recognition Vendor Test 2002 Performance Metrics. National Institute of Standards and Technology, Gaithersburg MD 20899. EUA. 2003.
- [31] GÜL, Ahmet Bahtiyar. Holistic Face Recognition By Dimension Reduction. Department of Electrical and Electronics Engineering, The Middle East Technical University. India. 2003.
- [32] GUNDIMADA, Satyanadh; ASARI, Vijayan. Face Detection Technique Based on Rotation Invariant Wavelet Features. VLSI Systems Laboratory, Department of Electrical and Computer Engineering, Old Dominion University. International Conference on Information Technology: Coding and Computing (ITCC'04). IEEE. 2004.
- [33] HAFED, Ziad M.; LEVINE, Martin D. Face Recognition Using the Discrete Cosine Transform. International Journal of Computer Vision. Vol. 43. N^o 3, 167–188. Center for Intelligent Machines, McGill University. Montreal - Canada. 2001.
- [34] HEISELE, Bernd; HO, Purdy; WU, Jane; POGGIO, Tomaso. Face recognition: component-based versus global approaches. ELSEVIER. Computer Vision and Image Understanding 91 6-21. 2003.
- [35] HOWELL, Andrew J. Automatic Face Recognition using Radial Basis Function Networks. Cognitive Science. Research Papers. 1997.
- [36] HSU, Rein-Lien; ABDEL-MOTTALEB, Mohamed; ANIL, K. Jain. Face detection in color images. Dep. of computer science & Engineering, Michigan State University. 2002.

- [37] HUANG, Jeffrey; WECHSLER, Harry. Eye Detection Using Optimal Wavelet Packets and Radial Basis Functions (RBFs). International Journal of Pattern Recognition and Artificial Intelligence. Vol 13. No 7. Department of Computer Science, George Mason University. 1999.
- [38] IZURIETA, Fernando; SAAVEDRA, Carlos. Redes Neuronales artificiales. Departamento de Física, Universidad de Concepción. Chile. 2000.
- [39] JUNG, D.; LEE, Chang-Woo; LEE, Yeon-Chu; BAK, Sang-Yong; KIM, Jong-Bae; KANG, Hyun; KIM, Hang-Joon. PCA-Base Real-Time Face Detection and Tracking. Korea. 2002.
- [40] JUNG, Marcia Y. Biometric Market and Industry Overview. World Customs Organization. International Biometric Group. Bruselas – Bélgica. 2005.
- [41] KAWATO, Shinjiro; OHYA, Jun. Two-step Approach for Real-time Eye Tracking with a New Filtering Technique. SMC2000: IEEE Int. Conf. on Systems, Man & Cybernetics, pp.1366-1371. Tennessee – EUA. 2000
- [42] KUKHAREV, Georgi; MASICZ, Pawel; MASICZ, Piotr. A Fast and Accurate Faces Localization using Gradient Method. Technical University of Szczecin, Faculty of Computer Science and Information Technology. 2004.
- [43] LAWRENCE, Steve; GILES, C. Lee; TSOI, Ah Chung; BACK, Andrew D. Face Recognition: A Hybrid Neural Network Approach. Technical Report UMIACS-TR-96-16 and CS-TR-3608. Institute for Advanced Computer Studies University of Maryland College Park, MD 20742.
- [44] LEE, Young. Face Detection and Recognition using Linear Classifiers. Department of Computer Science. Calgary - Alberta. 2004.
- [45] LI, Wen-shu; ZHOU, Chang-le; XU, Jia-tuo. A novel face recognition method with feature combination. Journal of Zhejiang University SCIENCE. 2005.
- [46] LIN, K. H.; LAM, K.M.; SIU, W.C. Locating the eye in human face images using fractal dimensions. IEEE - Image Signal Process. Vol. 148. N° 6. 2001.
- [47] LU, Xiaoguang. Image Analysis for Face Recognition. Dept. of Computer Science & Engineering, Michigan State University. 2005.
- [48] MARCEL, Sebastien; MARCEL, Christine; BENGIO, Samy. A State-Of-The-Art Neural Network for robust face verification. Research Report: The Advent Of Biometrics On The Internet. Roma – Italia. 2002.
- [49] MARKS, Tim. Principal Component Analysis (PCA). Cognitive Science Department. 2002.
- [50] MARTIN, Luis. Introducción a las Redes Neuronales Artificiales. Departamento de Ciencias de la Computación, Facultad de Matemáticas y Computación. Universidad de La Habana. Curso de Inteligencia Artificial. 2003–2004.

- [51] MARTÍNEZ, Aleix. Recognizing Imprecisely Localized, Partially Occluded and Expression Variant Faces from a Single Sample per Class. School of Electrical and Computer Engineering, Purdue University. IEEE Transactions on Pattern Analysis and Machine Intelligence. 1999.
- [52] MOLINA, R. Introducción al Procesamiento y Análisis de Imágenes Digitales. Departamento de Ciencias de la Computación e I. A., Universidad de Granada . 2002.
- [53] MOSES, Y.; ADINI, Y.; ULLMAN, S. Face recognition: the problem of compensating for illumination changes . In Eklundh, J. O. (Ed.), Proceedings of European Conference on Computer Vision, Lecture Notes in Computer Science. Vol. 800, pp.286-296. Estocolmo - Suecia. 1994.
- [54] NEFIAN, Ara V. Statistical Approaches to Face Recognition. Georgia Institute of technology, School of Electrical Engineering. 1996.
- [55] PHILLIPS, P. Jonathon; MOON, Hyeonjoon; RIZVI, Syed A.; RAUSS, Patrick J. The FERET Evaluation Methodology for Face-Recognition Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 22. Nº 10. 2000.
- [56] PIETROWCEW, A. Face detection in colour images using fuzzy Hough transform. OPTO-ELECTRONICS REVIEW. Rev. 11(3), 247-251. Institute of Radioelectronics, Faculty of Electronics and Information Technology, Warsaw University of Technology. Warsaw - Polonia. 2003.
- [57] POGGIO, Tomaso; GIROSI, Federico. A Theory of Networks for Approximation and Learning. Massachusetts Institute of Technology Artificial Intelligence Laboratory and Center for Biological Information Processing Whitaker College. 2001.
- [58] RAMÍREZ, Carlos; PÉREZ, Manuel. Procesamiento de imágenes digitales. Detección de caras y análisis de expresiones faciales. 2004.
- [59] RASIWASIA, Nikhil. Color Space for Skin Detection – A Review. Fondazione Graphitech, University of Trento. Italia.
- [60] RODRÍGUEZ, Yann; CARDINAUX, Fabien; BENGIO, Samy; MARIÉTHOZ, Johnny. Estimating the Quality of Face Localization for Face Verification. 2004.
- [61] ROMDHANI, S. Face Recognition Using Principal Components Analysis. 1997.
- [62] ROMERO, Luis Alonso; CALONGE, Teodoro. Capítulo 1.- Redes Neuronales y Reconocimiento de Patrones. España. 2001.
- [63] ROSE GÓMEZ, César Enrique. Reconocimiento e Identificación de Rostros usando análisis de los Principales Componentes. Instituto Tecnológico de Hermosillo, Departamento de Sistemas y Computación. 2001

- [64] ROWLEY, Henry A.; BALUJA, Shumeet; KANADE, Takeo. Neural Network-Based Face Detection. IEEE. 1998.
- [65] SALINAS, Renato; LARRAGUIBEL, Luis. Red Neuronal de Arquitectura Paramétrica en Reconocimiento de Rostros. 2002.
- [66] SAVVIDES, Marios; KUMAR, Vijaya; KHOSLA, Pradeep. Face Verification using Correlation Filters. Electrical and Computer Engineering Department, University Carnegie Mellon. Pittsburg - EUA.
- [67] SINGH, S.K.; CHAUHAN, D. S.; VATSA, M.; SINGH, R. Design of Real Time Face Recognition System. ElectronicsLetters. 2003.
- [68] SMITH, Lindsay. A tutorial on Principal Components Analysis. 2002.
- [69] SPANEL, Michal. Face Representation and Tracking Using Gabor Wavelet Networks. CESC. Brno University of Technology, Faculty of Information Technology Computer Graphics and Multimedia Department. 2003.
- [70] TAKAYA, Kunio; CHOIÛ, Kyung-Yung. Detection of Facial Components In A Video Sequence By Independent Component Analysis. Electrical Engineering, University of Saskatchewan. Canada. 2001.
- [71] THILAK KUMAR, S.; RAJA, Kumar; RAMAKRISHNAN, A.G. Eye detection using color cues and projection functions. IEEE ICIP. Department of Electrical Engineering, Indian Institute of Science. Bangalore. 2002.
- [72] TURK, Matthew. Eigenfaces and Beyond. Computer Science Department University of California. Academic Press. 2005.
- [73] TURK, Matthew. A Random Walk through Eigenspace. Invited Paper Special Issue on Machine Vision Applications. IEICE TRANS. INF. & SYST. Vol. E84-D. Nº12. 2001.
- [74] TURK, M.; PENTLAND, A. Eigenfaces for Face Detection/Recognition. Journal of Cognitive Neuroscience. Vol. 3. Nº 1, pp. 71-86. 1991.
- [75] ULRICH, Markus; STEGER, Carsten; BAUMGARTNER, Albert; EBNER, Heinrich. Real-Time Object Recognition Using a Modified Generalized Hough Transform. 2002.
- [76] VARIOS. Face Recognition. National Science and Technology Council (NSTC), Committees on Technology and Homeland & National Security, Subcommittee on Biometrics. 2006.
- [77] VARIOS. Real Time Face Detection and Tracking on Personal Computer. 2002.
- [78] VARIOS. Syntactic Pattern Recognition Overview. EE 583 Pattern Recognition. 2001.

- [79] VARIOS. Syntactic Pattern Recognition via Parsing. EE 583 Pattern Recognition. 2001.
- [80] VEZHNEVETS, Vladimir; SAZONOV, Vassili; ANDREEVA, Alla. A Survey on Pixel-Based Skin Color Detection Techniques. Graphics and Media Laboratory, Faculty of Computational Mathematics and Cybernetics, Moscow State University. Moscú - Rusia. 2004
- [81] VILLEGAS, C.; QUEZADA, A.; KURI, F. Reconocimiento Holístico de Rostros a través de Análisis Multivariado y Algoritmos Genéticos: Resultados Preliminares. Centro de Investigación en Computación, Instituto Politécnico Nacional. 2001.
- [82] WANG, Y.H.; TAN, T.N.; ZHU, Y. Face Verification Based on Singular Value Decomposition and Radial Basis Function Neural Network. National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences. Beijing - China. 2001.
- [83] WISKOTT, Laurenz; FELLOUS, Jean-Marc; KRÜGER, Norbert; MALSBURG, Christoph von der. Face Recognition by Elastic Bunch Graph Matching. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol. 19. Nº. 7. 1997.
- [84] WISKOTT, Laurenz; MALSBURG, Christoph von der. A Neural System for the Recognition of Partially Occluded Objects in Cluttered Scene -A Pilot Study-. Int. J. of Pattern Recognition and Artificial Intelligence. Bochum - Germany. 1993.
- [85] XU, Xiaoyin. Image - Based Face Recognition using Global Features. Research Centre for Integrated Microsystems Electrical and Computer Engineering University of Windsor. 2005.
- [86] YAMBOR, Wendy S. Analysis of Pca-Based and Fisher Discriminant-Based Image Recognition Algorithms. Technical Report CS-00-103. Computer Science Department Colorado State University Fort Collins, CO 80523-1873. 2000.
- [87] YANG, Jar-Ferr; HAO, Shu-Sheng. Modified Hough Transforms for Object Feature Extraction. Journal of Information Science and Engineering. Nº 17, pp.133-145. Tainan - Taiwan. 2001.
- [88] YANG, Ming-Hsuan; KRIEGMAN, David J.; AHUJA, Narendra. Detecting Faces in Images: A Survey. IEEE Transactions On Pattern Analysis And Machine Intelligence. Vol. 24. Nº 1. 2002.
- [89] ZHANG, Bai-Ling; ZHANG, Haihong; SAM GE, Shuzhi. Face Recognition by Applying Wavelet Subband Representation and Kernel Associative Memory. IEEE Transactions on Neural Networks. Vol. 15. Nº. 1. 2004.
- [90] ZHANG, J.; YAN, Y.; LADES, M. Face Recognition: Eigenface, Elastic Matching, and Neural Nets. IEEE. Vol. 85. Nº. 9. 1997

- [91] ZHAO, W.; CHELLAPPA, R.; ROSENFELD, A.; PHILLIPS, P.J. Face Recognition: A Literature Survey. National Institute of Standards and Technology, University of Maryland. 2000.
- [92] ZHAO, WenYi; CHELLAPPA, Rama. Image based Face Recognition Issues and Methods. 2004.
- [93] ZHOU, Shaohua; KRUEGER, Volker; CHELLAPPA, Rama. Face Recognition from Video: A Condensation Approach. Fifth IEEE International Conference on Automatic Face and Gesture Recognition. IEEE. Computer Society. 2002.
- [94] ZHOU, Zhi-Hua; GENG, Xin. Projection Functions for Eye Detection. Nanjing University. China. 2003.

DIRECCIONES ELECTRÓNICAS

- [95] ACOSTA, Emiliano; TORRES, Luis; ALBIOL, Alberto; DELP, Edward.
http://gps-tsc.upc.es/imatge/pub/ps/ICASSP02_Acosta_Torres_Albiol_Delp.pdf
 An Automatic Face Detection And Recognition System For Video Indexing Applications. 2001
- [96] ANÓNIMO
<http://www.us.es/gtocoma/pid/pid10/RedesNeuronales.htm>
 Conceptos básicos sobre redes neuronales. 1999.
- [97] FERRER, Alba. Apuntes GMMD Grafismo I. "Resolución en la imagen digital". 2001.
- [98] GUTIÉRREZ González, L.
http://www.udlap.mx/~tesis/msp/gutierrez_g_l/
 Reconocimiento de rostros utilizando redes neuronales. 2000.
- [99] PENTLAND, Alex; CHOUDHURY, Tanzeem.
<http://vismod.media.mit.edu/tech-reports/TR-516/node12.html>
 Personalizing Smart Environments: Face Recognition for Human Interaction. Massachusetts Institute of Technology. 2000
- [100] Siete punto Cero.
<http://www.sietepuntocero.com/sieteinfo/articulos/ResumenLaImagenDigital.pdf>
 La Imagen Digital. Notas (Extracto).
- [101] SOSSA, Juan Humberto; RAYÓN, Patricia; FIGUEROA, Jesús.
<http://www.cic.ipn.mx/~project/identification/redi2.PDF>
 Arquitectura Basada en Redes Neuronales para el Reconocimiento de Rostros. México.

GLOSARIO

- **Agrupamiento o clustering:** Agrupación de objetos similares en un espacio multidimensional. Técnica empleada principalmente para construir características que constituyan abstracciones de las características existentes.
- **Análisis de Componentes Principales (PCA):** Técnica estadística empleada para la construcción de nuevas características, componentes principales de un conjunto de datos.
- **Biometría:** Conjunto de métodos para la identificación y autenticación de los seres humanos, se basan en uno o varios rasgos físicos o de comportamiento propios de cada persona. La biometría informática constituye la aplicación de técnicas matemáticas y estadísticas sobre dichos rasgos con el objetivo de verificar identidades o identificar personas. Entre los rasgos que se analizan están: rasgos físicos como la geometría de la mano, iris, retina, rostro, huellas digitales; rasgos de comportamiento como la firma o la voz.
- **Convolución:** Este proceso sirve como base para la mayoría de los algoritmos de filtrado espacial. Consiste en promediar el valor de cada píxel en una imagen con los valores de sus vecinos, de tal manera que cada píxel resultante es afectado por el área circundante a éste de una forma que se especifica matemáticamente con una malla, ventana o filtro.
- **Cuantización:** Pérdida de datos provocada por la conversión de éstos a una escala diferente que no coincide con exactitud en tamaño o intervalos a la escala inicial.
- **Error cuadrático medio:** En estadística el error cuadrático medio es el valor esperado para el cuadrado del error, a este error se lo define como la cantidad por la que el estimador difiere de la cantidad a ser estimada. Con relación a las redes neuronales artificiales esta medida se refiere al error existente entre las salidas actuales de la red respecto a las deseadas.

- **FAR** (*False Acceptance Rate*): Representa la tasa de reconocimientos falsos. Con relación a la verificación de rostros se refiere a rostros identificados como conocidos sin serlo.
- **FRR** (*False Rejection Rate*): Representa la tasa de rechazos falsos. Con relación a la verificación de rostros se refiere a rostros de usuarios conocidos que no han sido identificados.
- **Gradiente**: Es la variación espacial de un campo escalar, da como resultado un vector que apunta en la dirección en la que la variación del campo es mayor por unidad de desplazamiento. El módulo de este vector expresa el valor de la variación.
- **NTSC** (*National Television System(s) Committee*): Sistema de codificación y transmisión. Formato que utiliza una resolución de 480 líneas horizontales y una tasa de refresco de 30 cuadros por segundo. Esto significa que el video en NTSC es mas bajo en resolución que el video en formato PAL pero debido a la velocidad de actualización el movimiento se aprecia mejor en el video para el formato NTSC.
- **Oclusión**: Desaparición total o parcial de algún área en la imagen.
- **PAL** (*Phase Alternating Line*): Sistema de codificación y transmisión. Formato cuyas características son: 576 líneas horizontales de resolución y una velocidad de actualización de 25 imágenes por segundo.
- **Ruido**: El ruido en imágenes digitales es cualquier valor de un píxel que no corresponde exactamente con la realidad. Cuando se adquiere una imagen digital, ésta está contaminada por ruido. El ruido se debe en la mayoría de los casos al equipo electrónico utilizado para la adquisición de las imágenes (ruido de cuantificación de la imagen, efecto de niebla en la imagen, etc.) y al ruido añadido en los tramos de transmisión (posibles interferencias o errores al transmitir los bits de información).

- **Selección de características:** Se refiere a la eliminación de ciertas características consideradas irrelevantes para el modelado de datos.

- **Validación cruzada:** Método para evaluar un modelo estadístico o un algoritmo que tiene algunos parámetros libres. Consiste en dividir al conjunto de datos disponible en dos subconjuntos independientes entre sí, de tal manera que un grupo de datos forme el conjunto de entrenamiento y los datos restantes formen el conjunto de prueba.

- **Visión por computador:** Conjunto de técnicas y modelos que permiten el procesamiento, análisis y explicación de cualquier tipo de información espacial obtenida mediante imágenes digitales.

- **VFW** (*Video para Windows*): Conjunto de controladores independientes del hardware. Permiten el funcionamiento de dispositivos de diferentes fabricantes. Actualmente VFW es casi obsoleto, aunque todavía existen dispositivos que lo usan.

- **WDM** (*Modelo de Dispositivo de Windows*): Arquitectura compuesta por un conjunto de controladores independientes del hardware, denominados controladores de clase. Cada fabricante de un hardware proporciona minicontroladores específicos que implementan funciones propias de dicho dispositivo, en tanto que para funciones generales el minicontrolador llama al controlador de clase para que las realice.

ANEXOS

Anexo 1

ELEMENTOS DE LA PERCEPCIÓN VISUAL

1.1 ESTRUCTURA DEL OJO HUMANO

El ojo es frecuentemente considerado el órgano de la visión, sin embargo es el cerebro quien efectúa este proceso. La función real del ojo es traducir las ondas electromagnéticas de luz que recibe, en impulsos nerviosos que se envían al cerebro a través del nervio óptico.

La figura 1.1 muestra una sección transversal horizontal del ojo humano. El ojo es un órgano fotosensible que se encuentra alojado en una cavidad denominada órbita, su diámetro aproximado es 2.5 cm y se divide en dos segmentos anatómicos: una pared exterior y un contenido interno.

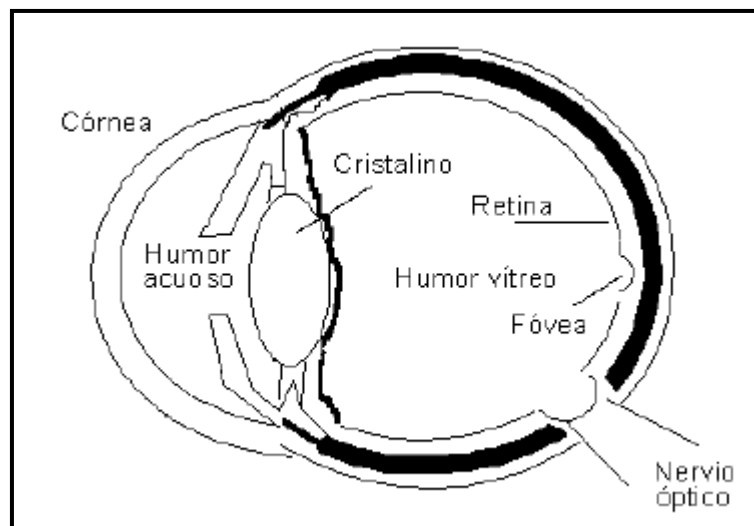


Figura 1.1. Ojo Humano

Fuente: ORTIZ, F. 2002. Pág. 14.

La pared exterior está a su vez formada por la esclerótica y la córnea. La esclerótica o esclera, es la membrana más externa y su función es proteger al globo ocular, mientras que la córnea es una capa transparente que cubre la parte frontal del ojo y permite la entrada de rayos luminosos al interior de éste. Detrás de la córnea existe un espacio lleno de líquido denominado humor acuoso que la separa del lente cristalino, y provoca la refracción de los rayos luminosos que lo atraviesan. El cristalino está ubicado detrás del iris, es un lente biconvexo constituido por una gran cantidad de fibras transparentes dispuestas en capas. Su función es enfocar la luz sobre la retina para que se visualice una imagen nítida. Detrás del cristalino se encuentra el cuerpo vítreo, cuya presión mantiene distendido el globo ocular.

El contenido interno se divide en dos capas. Una capa media compuesta por tres partes: la coroides que es la responsable de la nutrición de los tejidos del ojo, el cuerpo ciliar que está formado por tejido conectivo y por el músculo ciliar y se encarga de producir el humor acuoso, finalmente se encuentra el iris que se extiende por toda el área frontal del ojo y se encarga de regular la cantidad de luz que entra y de proporcionar el color a éste. La abertura circular en el centro del iris es la pupila.

La capa más interna, es la retina, lugar donde se origina el nervio óptico. La retina está formada por células receptoras denominadas bastones y conos. Los bastones son células sensibles a la intensidad luminosa y se los relaciona con la visión escotópica o visión de luz tenue (bajos niveles de luz), se encuentran localizados a lo largo de la retina y comparten entre ellos terminaciones nerviosas por lo que la visión es menos fina. Por su parte los conos son sensibles a los colores y se relacionan con la visión fotópica o visión de luz brillante (niveles mayores de luz), cada uno está conectado a su propia terminación nerviosa por lo que ayudan a la visión fina.

En la parte posterior de la retina se encuentra el punto ciego o papila óptica que es una zona insensible a la luz ya que no posee bastones ni conos, y una mancha amarilla denominada mácula lútea donde se encuentra la fovea, que

es la zona del ojo con mayor agudeza visual ya que en ésta se encuentran localizados los conos.

1.2 FORMACIÓN DE LAS IMÁGENES EN EL OJO

El ojo actúa como órgano receptor de los rayos luminosos. Una vez que dichos rayos son refractados por los elementos transparentes del ojo son dirigidos hacia la retina donde se proyectan como una imagen invertida y pequeña respecto a la real. En las células fotorreceptoras de la retina nace el estímulo que por las vías ópticas llega al cerebro, donde se elabora la interpretación del impulso nervioso.

Aunque la imagen se proyecta en la retina de forma invertida, existe un proceso de carácter psicológico que se inicia desde la niñez, en el cual por asociaciones diversas especialmente aquellas proporcionadas por el sentido del tacto, provoca el enderezamiento de la imagen.

1.3 ADAPTACIÓN AL BRILLO Y DISCRIMINACIÓN

Es importante conocer la capacidad de adaptación que tiene el ojo humano a los diferentes niveles de brillo, ya que esto es una consideración importante en la presentación de resultados de procesamiento de imágenes digitalizadas.

El rango de niveles de intensidad de luz a los cuales se puede adaptar el ojo humano está en el orden de 10^{10} desde el umbral de percepción escotópica (bajos niveles de iluminación) hasta el límite de encandilamiento. Sin embargo no se puede operar sobre todo ese rango de manera simultánea, por lo que el rango total de niveles de intensidad que puede ser discriminado simultáneamente es pequeño en comparación con el rango total.

La relación entre la intensidad de la luz que entra en el ojo y el brillo percibido no es lineal sino que tiende a ser logarítmica, es decir, a medida que la

intensidad de una fuente luminosa cambia no se percibirá un cambio igual en el brillo. Por ejemplo, mediante experimentos se ha demostrado que la intensidad de una fuente luminosa debe ser cercana al doble antes que el ojo detecte que ésta ha cambiado.

También se debe notar que los cambios tenues de intensidad son más perceptibles en áreas oscuras que en áreas brillantes de una imagen.

Esta forma no lineal en la que opera el sistema visual humano da lugar a fenómenos ópticos conocidos como *ilusiones ópticas*, dentro de este tipo de efectos se puede destacar las *bandas de Mach*.

Otros tipos de fenómenos ópticos producidos por las limitaciones y características propias del sistema visual humano son las *ambigüedades* y las *inconsistencias*.

1.3.1 Bandas de Mach

Las bandas de Mach muestran el efecto de contraste en un borde. Se trata de una imagen compuesta por un conjunto de franjas con distintos niveles de gris. El tono es uniforme para todas las franjas, pero el brillo varía progresivamente. Como se puede observar en la Figura 1.2 cualquier franja se ve más clara en su lado izquierdo y más oscura en su lado derecho aunque el nivel de gris en cada franja es uniforme. Las bandas o franjas no existen realmente son el resultado de una ilusión óptica provocada por la forma en la que el ojo se adapta a la variación del brillo.



Figura 1.2. Bandas de Mach

1.4 RESPUESTA AL COLOR

Las ondas electromagnéticas son ondas producidas por la aceleración de una carga eléctrica, se las puede ordenar en un espectro electromagnético de acuerdo a su frecuencia. Existen ondas de frecuencias muy elevadas hasta frecuencias muy bajas, así tenemos por ejemplo los rayos x, los rayos y, la luz ultravioleta, la luz visible, la luz infrarroja y las ondas radioeléctricas cortas y largas.

La luz visible o espectro visible es solo una parte pequeña del espectro electromagnético, se la denomina así porque es el segmento del espectro que puede ser percibido por el ojo; está en el rango que va desde los 380 nm de longitud de onda (color violeta) hasta los 780 nm (color rojo), con una mayor sensibilidad para el ojo humano en la longitud de onda que corresponde al color amarillo y la mínima en las longitudes correspondientes a los colores rojo y violeta, fuera de este rango el ojo no distingue ninguna clase de radiación.

Los colores que el ojo humano percibe en un objeto son el resultado de la luz que refleja dicho objeto. Por ejemplo si un objeto se ve de color verde es porque refleja luz que está en el rango de 500 a 570nm y absorbe el resto. Cuando se reduce la iluminación gradualmente, los objetos comienzan a perder color de acuerdo a un orden determinado: rojo, amarillo, verde o azul. Mientras mayor sea la reducción de iluminación los colores dejan de percibirse hasta distinguirse solo matices de gris, negro y blanco azulado. Así se explica por ejemplo que al mirar un jardín a plena luz del sol sean las flores amarillas las que más sobresalen mientras que en el crepúsculo son las flores de color azul las que más resaltan.

De acuerdo a la respuesta del ojo a las distintas longitudes de onda se tiene la visión fotópica y la escotópica.

Anexo 2

FENÓMENO DE LA LUZ

2.1 LA LUZ

La luz visible es el único tipo de radiación electromagnética que puede ser vista por el ojo humano.

Por esta razón el análisis se centrará en este segmento del espectro electromagnético y se utilizará el término *luz* para referirse a la luz visible.

La naturaleza de la luz ha sido explicada con diversas teorías, sin embargo las más destacadas son la *teoría corpuscular* y la *teoría ondulatoria*.

- La teoría corpuscular fue propuesta por Isaac Newton en el siglo XVII. De acuerdo con esta teoría las fuentes luminosas emiten corpúsculos muy livianos a gran velocidad y en línea recta. Dichas partículas pueden atravesar medios transparentes y ser reflejadas por materias opacas.
- Por su parte la teoría ondulatoria fue defendida por Christian Huygens en el año 1678, en ésta se describe a la luz como un movimiento ondulatorio semejante al producido con el sonido.

Actualmente se considera que la luz tiene una naturaleza doble, que permite explicar los diferentes fenómenos de propagación (naturaleza ondulatoria) y de interacción (naturaleza corpuscular).

A continuación se detallarán algunas propiedades de la luz así como los tipos de luz que existen.

2.2 PROPIEDADES ÓPTICAS DE LA LUZ

Cuando la luz se refleja sobre un objeto su comportamiento depende de la superficie y naturaleza de éste, lo cual da origen a ciertos fenómenos físicos que se consideran como propiedades ópticas de la luz:

- *Absorción:* Cuando un rayo de luz se refleja sobre una superficie opaca, negra o mate, se absorbe casi en su totalidad y se transforma en calor.
- *Reflexión:* Al reflejarse un haz de luz sobre un objeto, algunas de estas ondas serán absorbidas y otras reflejadas. En el caso de una superficie lisa y brillante el haz de ondas se refleja en un ángulo igual al de incidencia (*reflexión especular*), pero si la superficie no es totalmente lisa y brillante entonces se refleja únicamente parte de las ondas y en diferentes direcciones(*reflexión difusa*).
- *Transmisión:* Es el fenómeno mediante el cual la luz puede atravesar objetos no opacos. Cuando la transmisión es íntegra y directa, se dice que el medio es transparente y la transmisión es *directa*. Si el rayo de luz se dispersa en diferentes direcciones se dice que el medio es translucido y que la transmisión es *difusa*.
- *Refracción:* Es la desviación que experimenta un rayo de luz cuando pasa de un medio a otro de diferente densidad. Este fenómeno ocurre dentro del fenómeno de transmisión.
- *Dispersión:* Efecto mediante el cual el índice de refracción de una sustancia es función de la longitud de onda de la luz. El índice de refracción es mayor cuando las longitudes de onda son cortas, además esta variación también depende del tipo de sustancia.
- *Difracción:* Fenómeno por medio del cual los rayos de luz pueden bordear los obstáculos.

2.3 TIPOS DE LUZ

De acuerdo a ciertas características, la luz se puede clasificar como monocromática y cromática.

- La luz monocromática también conocida como acromática es aquella formada por componentes de un solo color, presenta como única característica su intensidad. Para definir la intensidad monocromática se utiliza el término *niveles de gris* , que va desde el color negro hasta el blanco pasando por diferentes matices de gris.

- La luz cromática abarca todo el espectro visible y tiene tres características que la describen: radianza, luminosidad y brillo.
 - La radianza es la cantidad total de energía proveniente de la fuente de luz, se mide en vatios.

 - La luminosidad mide la cantidad de energía emitida por la fuente de luz, que es perceptible por el ojo humano, se mide en lúmenes.

 - El brillo es una medida subjetiva que está relacionada con la noción acromática de intensidad.

RELACIONES BÁSICAS ENTRE PÍXELES

3.1 VECINOS DE UN PÍXEL

Considerando una relación en dos dimensiones, se dice que los vecinos de un píxel son aquellos que se encuentran ubicados a cierta distancia de éste y que comparten un lado o un vértice con dicho píxel.

- a. *Cuatro vecinos:* Se representan por $N_4(p)$, son los píxeles adyacentes a un píxel p , por lo tanto comparten con éste un lado y están a una unidad de distancia, (véase Figura 1.7 (a)).
- b. *Ocho vecinos:* Se representan por $N_8(p)$, son un conjunto de píxeles formado por los cuatro vecinos de un píxel p y sus vecinos en diagonal. Es decir, son los píxeles que comparten ya sea un lado o un vértice con p , (véase Figura 1.7 (b)).

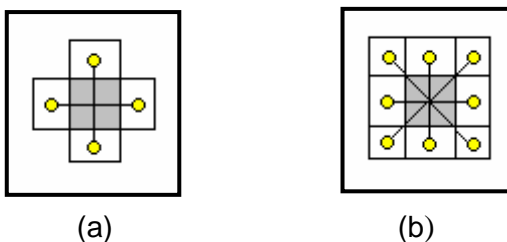


Figura 3.1. Vecinos de un píxel: (a) cuatro vecinos y (b) ocho vecinos

Para los píxeles que se encuentran en el borde de la imagen algunos de sus vecinos caen fuera de la misma.

3.2 CONECTIVIDAD

Describe una relación entre dos o más píxeles (sucesión de diversos píxeles conectados). Su objetivo es establecer los límites o bordes de objetos así como los objetos existentes en una imagen. Para que dos píxeles estén conectados deben cumplir ciertas condiciones en cuanto a similitud en su valor y proximidad.

Para imágenes en escala de grises los píxeles deben tener valores que estén dentro de un rango determinado de niveles de gris, en el caso de una imagen binaria (ceros y unos) los píxeles deben tener valor 1.

En lo referente a la proximidad se debe considerar el criterio de *vecinos de un píxel* descrito en la sección anterior. De acuerdo a este criterio se puede definir tres tipos de conectividad, los cuales están detallados en la Tabla 1.1.

Tipos de Conectividad	
<i>4-conectividad</i>	Píxeles adyacentes (4-vecinos) con valores similares.
<i>8-conectividad</i>	Píxeles que sean 8-vecinos, con valores similares.
<i>m-conectividad o conectividad mixta</i>	Píxeles que sean 4-vecinos o sean vecinos en diagonal pero que no tengan píxeles adyacentes en común. En ambos casos deben tener valores similares

Tabla 3.1. Resumen de los tipos de conectividad entre píxeles

3.3 ETIQUETADO DE COMPONENTES CONEXAS

Se denomina componente conexa al conjunto de píxeles en una imagen que están conectados entre sí, las componentes conexas distintas son disjuntas.

El etiquetado de componentes conexas se refiere al proceso de determinar todas las componentes existentes en una imagen y asignar a cada una de

ellas una etiqueta distintiva. Es importante mencionar que las componentes conexas existentes en una imagen dependen en gran medida del tipo de conectividad utilizado.

En el siguiente ejemplo se puede observar que si se utiliza 4-conectividad se tienen tres componentes mientras que si se utiliza 8-conectividad se tienen sólo dos. La Figura 1.8 representa a la matriz resultante del proceso de etiquetado utilizando 4-conectividad, los números mayores que cero son etiquetas que indican las componentes conexas encontradas en la imagen:

0	0	0	0	0	0	0	0
0	1	1	0	0	3	3	3
0	1	1	0	0	0	3	3
0	1	1	0	0	0	0	0
0	0	0	2	2	0	0	0
0	0	0	2	2	0	0	0
0	0	0	2	2	0	0	0

Figura 3.2. Matriz resultante del etiquetado utilizando 4-conectividad

Si la misma imagen es etiquetada utilizando 8-conectividad se obtienen diferentes componentes debido a la conexión diagonal, (véase Figura 1.9)

0	0	0	0	0	0	0	0
0	1	1	0	0	2	2	2
0	1	1	0	0	0	2	2
0	1	1	0	0	0	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0
0	0	0	1	1	0	0	0

Figura 3.3. Matriz resultante del etiquetado utilizando 8-conectividad

3.4 MEDIDAS DE DISTANCIA

Para determinar cuan cerca o lejos se encuentran dos píxeles en una imagen se recurre a una métrica o medida que indique la distancia entre ellos. La Tabla 1.2 presenta algunos métodos comúnmente usados para determinar la distancia entre dos píxeles.

Métodos para determinar distancia entre píxeles	
<i>Distancia euclidiana</i>	Indica la distancia en línea recta entre dos píxeles, es la métrica más cercana a la realidad pero es la más costosa de calcular. Se define por: $d(x,y) = \sqrt{\sum (x_i - y_i)^2}$
<i>Manhattan</i>	También es conocida como <i>City Block</i> , esta métrica calcula la ruta entre los píxeles basándose en 4-conectividad. Se define por: $d(x,y) = \sum x_i - y_i $
<i>Distancia de Tablero de Ajedrez</i>	También conocida como D_8 . Calcula la ruta entre los píxeles basándose en 8-conectividad. Se define por: $d(x,y) = \max(x_1 - x_2 , y_1 - y_2)$

Tabla 3.2. Métodos de cálculo de distancia más utilizados

3.5 OPERACIONES ARITMÉTICO/LÓGICAS

Dentro del tratamiento digital de imágenes, las operaciones aritméticas y lógicas entre píxeles son procesos básicos y que en muchas ocasiones están inmersos en procesos más complejos.

Las operaciones aritméticas además de poder aplicarse entre píxeles se pueden efectuar entre un píxel y un valor escalar. En cuanto a las operaciones lógicas, éstas se realizan únicamente en imágenes binarias

En la Tabla 1.3 se reseñan las operaciones aritmético/lógicas más importantes dentro del tratamiento digital de imágenes.

Operaciones Aritmético/Lógicas	
<i>Operaciones Aritméticas</i>	<ul style="list-style-type: none"> ○ Adición: Sean p y q píxeles, la adición se define como $p + q$. Esta operación se utiliza principalmente para la reducción de ruido en las imágenes.
	<ul style="list-style-type: none"> ○ Sustracción: Se define como $p - q$ Se puede utilizar para eliminar información irrelevante en una imagen, como por ejemplo un fondo.
	<ul style="list-style-type: none"> ○ Multiplicación: Se define como $p * q$ Se puede utilizar por ejemplo para corregir variaciones en los niveles de gris de una imagen.
	<ul style="list-style-type: none"> ○ División: Se define como p / q Al igual que la multiplicación, se puede utilizar para la corrección de iluminación.
<i>Operaciones Lógicas</i>	<ul style="list-style-type: none"> ○ AND: Y lógico. El resultado de esta operación será 1 sólo cuando el valor de los dos píxeles sea 1.
	<ul style="list-style-type: none"> ○ OR: Or lógico El resultado de esta operación será 1 cuando uno o ambos valores de los píxeles sea 1.
	<ul style="list-style-type: none"> ○ Complemento: Sea p un píxel su complemento es NOT p.

Tabla 3.3. Resumen de las operaciones aritméticas y lógicas que se pueden efectuar en imágenes.

Tanto las operaciones aritméticas como las lógicas pueden ser ejecutadas no solo píxel a píxel sino utilizando los píxeles circundantes. Para esto se hace referencia a las *máscaras o filtros*, que son ventanas o matrices de un tamaño determinado, generalmente mucho menor al de la imagen completa, que se centran sobre un píxel y utilizan los valores de los vecinos de éste para que mediante las operaciones aritméticas o lógicas se establezca un nuevo valor del píxel, de tal manera que el nuevo valor de ese píxel esté en función no solo de su valor inicial sino del de sus vecinos.

TRANSFORMACIONES ESPACIALES

Las transformaciones espaciales realizan un mapeo de los píxeles desde su posición en la imagen inicial hacia una nueva posición en la imagen resultante.

A continuación se describen algunas transformaciones básicas y términos relacionados con éstas.

4.1 INTERPOLACIÓN

Es el proceso de estimar valores desconocidos a partir de otros ya conocidos mediante la aplicación de ciertos algoritmos. Por ejemplo, si se quiere aumentar el tamaño de una imagen, se requiere un mayor número de píxeles, que originalmente no existen, por lo tanto se recurre a la interpolación para calcular los valores que cada uno de los píxeles adicionales debe tener.

Como se puede deducir, esta transformación afecta la calidad de la imagen, mientras mayor sea la interpolación se tendrá una mayor pérdida en la calidad.

Entre los métodos de interpolación más usados están:

- a. *Interpolación del punto más cercano*: Consiste en asignar al píxel de salida el valor del punto más cercano. No se consideran otros píxeles.
Cuando existen objetos con líneas rectas, este tipo de interpolación provoca un efecto de escalera en la imagen resultante.
- b. *Interpolación bilineal*: Obtiene el promedio de los 4 píxeles más cercanos asumiendo que la función de brillo es lineal con la distancia entre píxeles.

La interpolación bilineal puede provocar un efecto de difuminado en la imagen resultante.

- c. *Interpolación bicúbica*: Se considera el método estándar de interpolación, promedia 16 píxeles adyacentes (en un área de 4 x 4) y aproxima la función de brillo mediante un polinomio bicúbico. Este tipo de interpolación conserva los detalles finos de la imagen.

La Figura 1.10 muestra los resultados de aplicar los métodos de interpolación antes descritos, para ampliar el tamaño de una imagen.

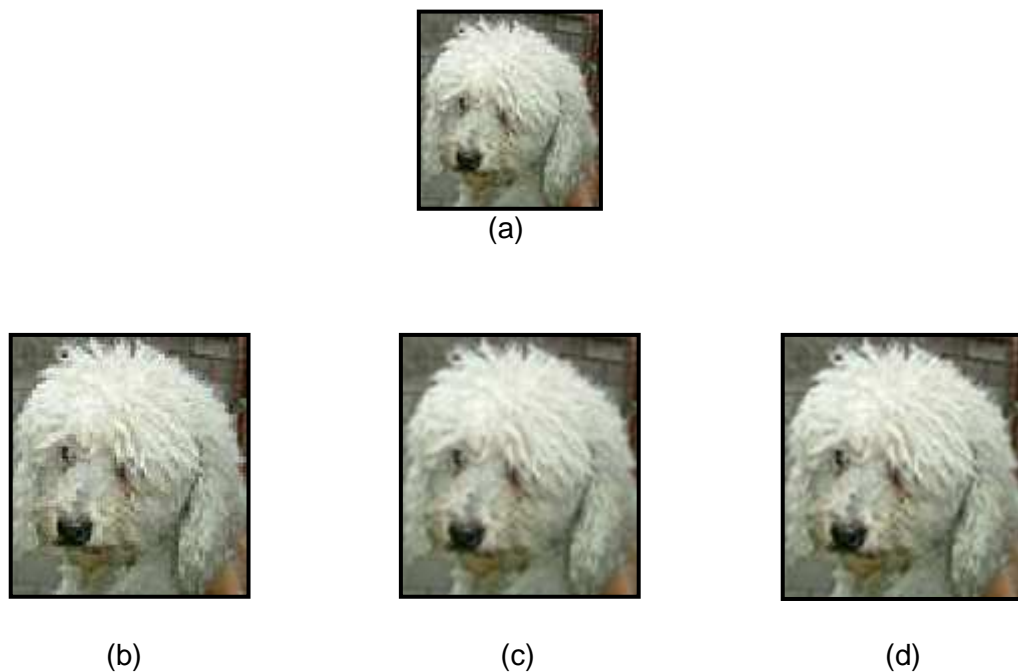


Figura 4.1. Imagen ampliada con diferentes métodos de Interpolación

(a) Imagen original (b) interpolación del punto más cercano, se distingue el efecto escalera en la imagen (c) interpolación bilineal, se aprecia el efecto de difuminado (d) interpolación bicúbica, preserva mejor los detalles.

Aunque el número de píxeles utilizados para la interpolación provoca un mayor esfuerzo computacional se debe tener en cuenta que mientras mayor sea el número de píxeles considerados, más precisa será la transformación.

4.2 TRANSFORMACIÓN DE ESCALA

Mediante esta transformación se puede reducir o aumentar el tamaño de una imagen.

- a. La reducción de una imagen o submuestreo consiste en el reemplazo de un conjunto de píxeles por un píxel del conjunto o en la interpolación de áreas locales de píxeles, (véase Figura 1.11). Al reducir el tamaño de una imagen se puede producir un efecto conocido como *aliasing* debido a la pérdida de información.

El efecto de *aliasing* provoca que ciertas curvas o líneas inclinadas en la imagen presenten un efecto visual tipo “sierra” o de “escalera”. Para evitar esto se puede aplicar previamente a la reducción, un filtro que permita disminuir el impacto del *aliasing* en la imagen resultante.

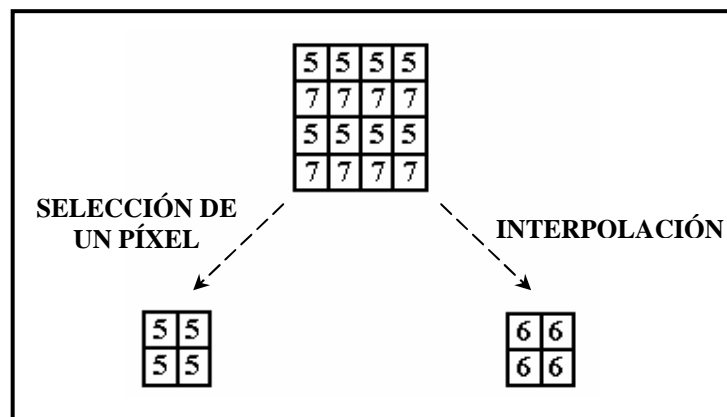


Figura 4.2. Métodos de submuestreo de una imagen

- b. El aumento del tamaño de una imagen se lo consigue mediante replicación o por interpolación de píxeles, (véase Figura 1.12).

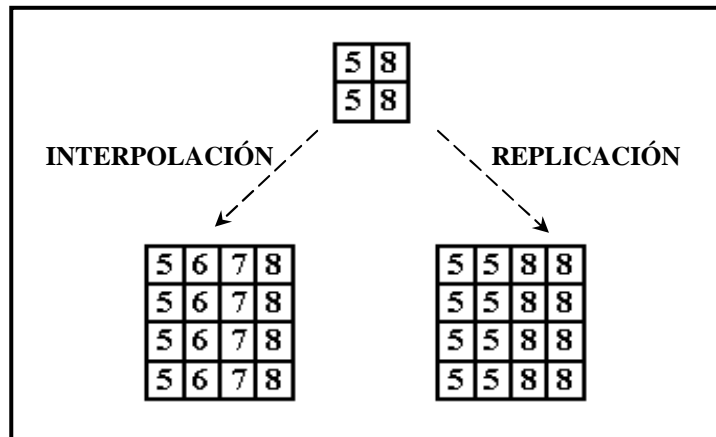


Figura 4.3. Métodos para ampliar una imagen

4.3 TRASLACIÓN

Esta transformación geométrica permite el desplazamiento de los píxeles de una imagen en una dirección determinada. Cada píxel localizado en una posición (x_1, y_1) en la imagen original es trasladado a una nueva posición (x_2, y_2) en la imagen resultante mediante ecuaciones previamente definidas, (véase Figura 1.13).



(a)



(b)

Figura 4.4. Ejemplo de traslación: (a) imagen original (b) imagen trasladada

El principal uso de la traslación es la alineación de imágenes para posteriormente efectuar operaciones de tipo aritmético.

4.4 ROTACIÓN

La rotación es una transformación geométrica que mapea la posición (x_1, y_1) de un píxel en una imagen inicial hacia una posición (x_2, y_2) en una imagen resultante, para esto utiliza un ángulo de rotación y un origen.

El principal uso de esta transformación es mejorar la apariencia visual de una imagen, es decir, generalmente se la emplea para corregir algún grado de inclinación de los objetos presentes en ésta, (véase Figura 1.14)

Cuando se realiza el proceso de rotación se obtienen píxeles localizados fuera de los bordes de la imagen original (fuera de las dimensiones de la imagen original), en estos casos se puede ignorar a dichos píxeles y obtener una imagen incompleta con relación a la inicial o bien se puede crear una imagen de dimensiones mayores a la original pero que contenga toda la información inicial y los píxeles que no pertenecen a la imagen original son rellenados con valor 0 (color negro).

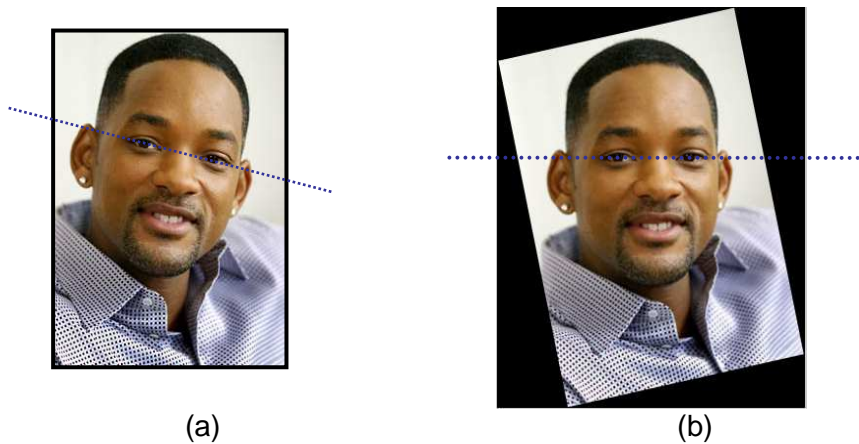


Figura 4.5. Ejemplo de rotación para corregir la inclinación de la línea de los ojos (a) imagen con el área del rostro levemente inclinada (b) imagen corregida

4.5 TRANSFORMACIÓN AFFINE

Es una transformación geométrica en la que los valores de los píxeles de la imagen original son mapeados a una imagen final o resultante mediante la aplicación de una combinación de otras transformaciones tales como: traslación, rotación y escalamiento. De esta manera las líneas rectas permanecen rectas y las líneas paralelas permanecen paralelas, pero la distancia y ángulos entre éstas líneas pueden cambiar.

Al aplicar esta transformación a una imagen distorsionada se puede corregir cierto nivel de distorsión.

4.6 EXTRACCIÓN DE ÁREAS DE LA IMAGEN

Es el proceso mediante el cual se remueven ciertos píxeles en una imagen ya sea para eliminarlos o utilizarlos como una nueva imagen.

La Figura 1.15 ilustra la extracción de un área específica en una imagen, este proceso es muy útil para aislar objetos de interés.

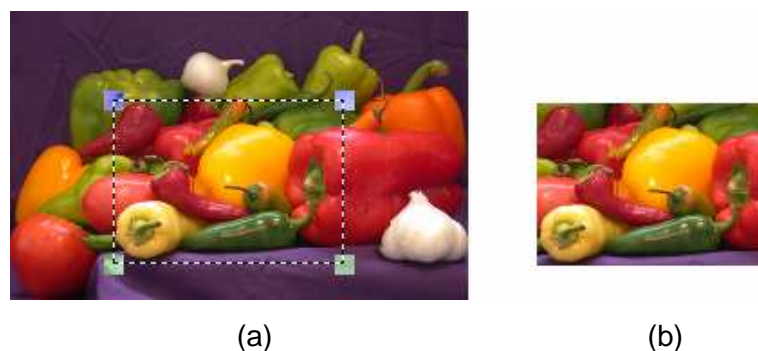


Figura 4.6. Extracción de un área en una imagen: (a) imagen original (b) área extraída

Anexo 5

MODELOS DE COLOR

A continuación se describen algunos de los espacios o modelos de color más utilizados.

5.1 MODELO RGB (ROJO-VERDE-AZUL)

Este modelo se basa en la síntesis aditiva, (véase Figura 1.17 (a)) por lo tanto trabaja con la combinación de los tres colores primarios: rojo, verde y azul.

Para representar un color en el sistema RGB se asigna un valor determinado para cada componente o color primario, respetando el orden establecido en el modelo (R,G,B). Este valor de acuerdo a la notación decimal debe estar entre 0 y 255 (correspondiendo el valor 0 al color negro) y de acuerdo a la notación hexadecimal entre 00 y FF.

Esto significa que las imágenes en este modelo están integradas por tres planos de imágenes independientes, (véase Figura 1.18), donde cada plano corresponde a un color primario. Cuando una imagen RGB es desplegada en un monitor, los tres planos son combinados para producir la imagen real.

La Figura 1.18 muestra una imagen en el espacio RGB y cada uno de sus planos o canales. El canal R presenta valores más altos que los otros canales, esto se debe a que el color predominante en la imagen utilizada es el rojo (la imagen tiene una tendencia rojiza); en tanto que el canal G presenta los valores más bajos con relación a los otros dos canales.



Figura 5.1. Imagen en modelo RGB y sus componentes (a) imagen original (b) canal R (c) canal G (d) canal B.

5.2 MODELO CMY

Este modelo es considerado sustractivo (véase Figura 1.17 (b)), puesto que se basa en la propiedad de ciertos elementos de absorber la luz que reciben. Si en la síntesis sustractiva se añade un componente de color negro se habla entonces de CMYK.

El modelo CMY se puede obtener mediante la transformación lineal del modelo RGB, para esto se recurre a la siguiente operación:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

5.3 MODELOS YIQ Y YUV

Estos modelos son utilizados como estándar por las emisoras de televisión. YIQ es utilizado como estándar para sistemas NTSC mientras que en los sistemas PAL se utiliza YUV.

En estos modelos se da una mayor importancia a la luminancia que a la crominancia, respondiendo así a características propias del sistema visual humano; además de permitir la compatibilidad con monitores de televisión monocromos en los que solo la información de luminosidad es procesada.

Los modelos YIQ y YUV pueden obtenerse a partir de transformaciones lineales del modelo RGB:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & 0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad \begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & 0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

En la Figura 1.19 se observa que si sólo se toma en cuenta al canal Y (luminosidad) de la imagen, lo que se tiene es una imagen en escala de grises, esto demuestra la capacidad de estos modelos de permitir compatibilidad con sistemas monocromos.



Figura 5.2. Componentes de una imagen en el modelo YIQ: (a) Canal Y (b) Canal I (c) Canal Q

5.4 MODELO HSV

También se lo conoce como modelo intuitivo porque se basa en la forma como el ojo humano percibe el color, por lo tanto utiliza las características más representativas: brillo, (véase Figura 1.20). Según el modelo HSV cualquier color se puede representar basándose en estos tres atributos, por ejemplo el color negro se obtiene con la ausencia de brillo, los grises con una saturación baja, entre otros.

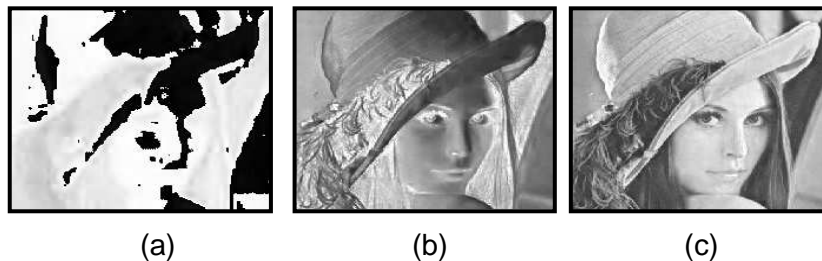


Figura 5.3. Componentes de una imagen en el modelo HSV: (a) Canal H (matiz) (b) Canal S (saturación) y (c) Canal V (brillo).

Al igual que varios modelos, éste puede ser calculado mediante una conversión del espacio RGB, sin embargo se debe mencionar que la conversión no es lineal.

Dentro del área de procesamiento de imágenes el modelo HSV es frecuentemente utilizado, una de las principales razones es porque que permite evitar en gran medida el efecto indeseable de brillo en una imagen. Esto es posible pues el brillo es procesado como un plano independiente sin afectar al color de la imagen.

5.5 MODELO L^*a^*b

Es un modelo de color tridimensional formado por un componente de luminosidad (L) junto a dos componentes cromáticos, uno que va de verde a rojo y otro que va de azul a amarillo.

Este modelo pretende mantener consistente el color independientemente del dispositivo usado.

En la Figura 1.21 se pueden observar los canales que componen este modelo. El canal de luminancia (L) tiene mayor definición en las formas que los otros dos canales.



Figura 5.4. Componentes de una imagen en el modelo L^*a^*b (a) Canal L (b) Canal $*a$ (c) Canal $*b$.

5.6 MODELO NRGB

El espacio RGB normalizado o NRGB es un espacio bidimensional que se forma independientemente de la intensidad del color, debido a esto es comúnmente utilizado dentro del procesamiento de imágenes.

Los componentes de este modelo se obtienen a partir del modelo RGB utilizando la siguiente transformación:

$$N_c = \frac{C}{R+G+B} \quad \text{para } C \in \{R,G,B\} \text{ y } R+G+B \neq 0$$

Sin embargo estos componentes son redundantes ya que $N_R + N_G + N_B = 1$, por lo tanto se suele considerar únicamente los componentes N_R y N_G .

Como se puede observar en este modelo se busca determinar la proporción de rojo, verde y azul en la imagen, en lugar del nivel de intensidad de cada color.

La conversión reversa, es decir de NRGB a RGB, no es posible ya que durante la conversión a NRGB existe pérdida de la información de intensidad.

TRANSFORMADAS SEPARABLES

En esta sección se tratará de forma general algunas transformadas que tienen la propiedad de separabilidad. Esta propiedad permite crear algoritmos más eficientes, que a su vez logran un decremento del costo computacional de las transformadas.

6.1 TRANSFORMADA DE FOURIER

La transformada de Fourier es una herramienta de gran importancia dentro del procesamiento de imágenes. Con su ayuda se puede descomponer a una imagen en sus componentes seno y coseno, esto significa que el resultado obtenido es la representación de dicha imagen en el *dominio de la frecuencia*³⁴, donde cada punto representa una frecuencia particular contenida en el dominio real de la imagen también conocido como *dominio espacial*³⁵.

Como se mencionó anteriormente sólo se considerará la Transformada de Fourier Discreta (DFT, por sus siglas en inglés).

Transformada de Fourier Discreta (DFT)

La transformada de Fourier Discreta es en sí la transformada de Fourier luego de un muestreo, lo que significa que utiliza un número finito de muestras discretas, en este caso píxeles que componen la imagen.

De esta manera el resultado de la transformada contiene un conjunto de muestras de frecuencias que es lo suficientemente grande como para

³⁴ El dominio de la frecuencia dentro del contexto de procesamiento de imágenes, representa la variación de los píxeles de la imagen en función de las coordenadas espaciales.

³⁵ El dominio espacial representa a las imágenes por intensidades en puntos dados en el espacio. Es la representación más común para las imágenes.

describir la imagen en el dominio espacial. El número de frecuencias corresponde al número de píxeles de la imagen.

La expresión matemática que define a DFT es:

$$F(p, q) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) e^{-j\left(\frac{2\pi}{M}\right)pm} e^{-j\left(\frac{2\pi}{N}\right)qn} \quad \begin{matrix} p = 0, 1, \dots, M - 1 \\ q = 0, 1, \dots, N - 1 \end{matrix}$$

donde $f(m, n)$ es la imagen en el dominio espacial, los valores m y n corresponden a las muestras utilizadas para la transformada (píxeles en la imagen) y el término exponencial es la función base correspondiente a cada punto $F(p, q)$ en el dominio de la frecuencia. Así se concluye que el valor de cada punto $F(p, q)$ se obtiene al multiplicar la imagen representada en el dominio espacial por la correspondiente función base y sumando los resultados. Las funciones base son senos y cosenos con frecuencias incrementales.

La transformada de Fourier produce como valores de salida números complejos, donde la parte real está asociada con la magnitud que se la conoce también como *espectro de Fourier* y la parte imaginaria con el ángulo de fase, (véase Figura 1.22).

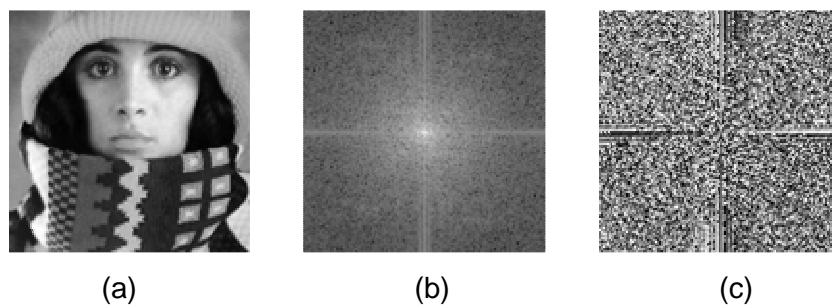


Figura 6.1. Transformada de Fourier (a) imagen original (b) logaritmo de la transformada (magnitud en forma escalada) (c) fase de la transformada

Fuente: YOUNG, I.T.; GERBRANDS, J.J.; VAN VLIET, L.J.

El valor $F(0,0)$ también se denomina *componente DC*, es el coeficiente de frecuencia cero, en tanto que el valor $F(M-1,N-1)$ representa a la frecuencia más alta.

Transformada de Fourier Inversa

Una imagen representada en el dominio de frecuencia puede ser re-transformada al dominio espacial aplicando la inversa de la transformada de Fourier.

La ecuación que describe este proceso es:

$$f(m,n) = \frac{1}{MN} \sum_{p=0}^{M-1} \sum_{q=0}^{N-1} F(p,q) e^{j\left(\frac{2\pi}{M}\right)pm} e^{j\left(\frac{2\pi}{N}\right)qn} \quad m = 0,1,\dots, M-1$$

$$n = 0,1,\dots, N-1$$

Dentro del procesamiento de imágenes generalmente es la magnitud la que se utiliza, sin embargo para aplicar la transformada de Fourier Inversa se requiere tanto la magnitud como el ángulo de fase, (véase Figura 1.23).

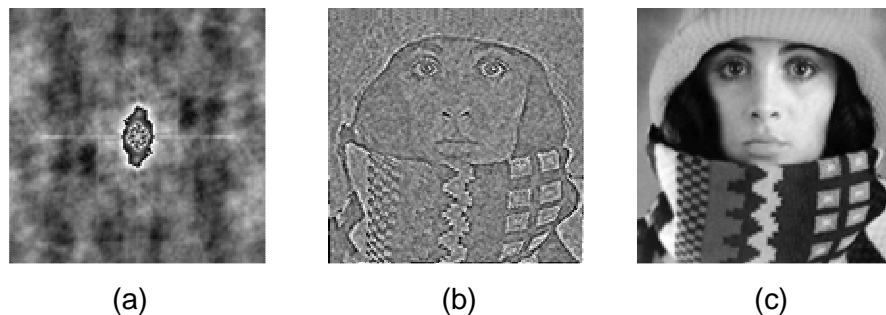


Figura 6.2. Reconstrucción de una imagen a partir de su Transformada de Fourier

- (a) imagen reconstruida sólo en base a la información de la magnitud (fase = 0)
- (b) imagen reconstruida sólo en base a la información de la fase (c)
- imagen reconstruida utilizando la información de la magnitud y de la fase.

Fuente: YOUNG, I.T.; GERBRANDS, J.J.; VAN VLIET, L.J

Transformada Rápida de Fourier

La transformada de Fourier discreta es computacionalmente costosa. Por ejemplo si se tiene una imagen de tamaño $M \times M$ se requerirá M^4 multiplicaciones complejas.

Por esta razón y considerando que la transformada de Fourier es separable, se calcula la transformada discreta de longitud N como la suma de dos transformadas de Fourier de longitud $N/2$. Esto se puede aplicar de manera recurrente al conjunto de puntos hasta alcanzar transformadas de solamente dos puntos. El proceso descrito se conoce como *Transformada rápida de Fourier* (FFT, por sus siglas en inglés), reduce la complejidad de N^2 a $N \log_2 N$ para un conjunto de N puntos. Se debe notar que la FFT no es otro tipo de transformada sólo se diferencia de DFT en el algoritmo utilizado.

De igual manera la FFT es separable, por lo que se puede reducir esta operación bidimensional, en dos operaciones unidimensionales, (véase Figura 1.24). Primero se calcula la FFT de las filas de la imagen y luego con lo obtenido la transformada de las columnas, o viceversa (primero la transformada para las columnas y luego para las filas). Así para una imagen de tamaño $M \times N$ se requiere calcular $M+N$ FFTs con una complejidad de $NM \log_2 NM$.

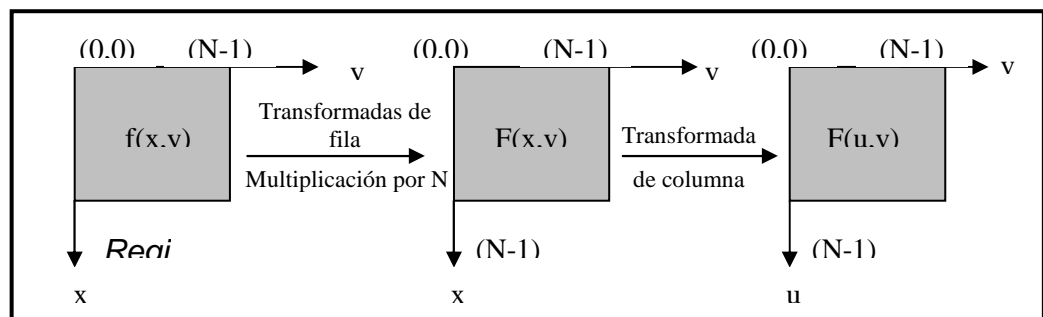


Figura 6.3. Obtención de la transformada de Fourier bidimensional como una serie de transformadas unidimensionales.

Fuente: GONZALES, R.; WOODS, R. 1996. Pág. 103.

Este mismo principio puede ser aplicado para la Transformada de Fourier Inversa, lo que se conoce como *Transformada Rápida Inversa de Fourier* (IFFT , por sus siglas en inglés).

Aplicaciones

A continuación se describen a breves rasgos, sólo algunas de las posibles aplicaciones de la Transformada de Fourier dentro del procesamiento de imágenes, sin embargo se debe recordar que la gama de aplicaciones es mucho mayor.

- Filtrado: Si se aplica DFT a una imagen y se eliminan ciertos componentes de frecuencia muy altos o muy bajos, al aplicar la transformada inversa se logra filtrar la imagen para así remover ruido o cualquier otro tipo de artefacto presente. Esta clase de operaciones se tratará en la Sección 1.1.9.4.
- Compresión: Se aplica la transformada a la imagen y se descartan los coeficientes de las frecuencias altas ya que se asume que éstas no son perceptibles. Para la descompresión se debe calcular la transformada inversa basándose en el número de coeficientes de Fourier previamente reducidos.
- Convolución: Es una operación utilizada para implementar filtros lineales, sin embargo en términos computacionales se puede emplear la transformada de Fourier para calcular de manera más rápida y eficiente la convolución, sobre todo cuando se trata de entradas muy grandes, ya que la multiplicación de dos transformadas de Fourier equivale a la convolución de la función espacial asociada.
- Localización de características en la imagen: La transformada de Fourier también puede ser utilizada para implementar una operación conocida como *Correlación*, que en ocasiones se emplea para localizar características u objetos en una imagen.

6.2 OTRAS TRANSFORMADAS SEPARABLES

Como se mencionó en una sección anterior, la transformada de Fourier es separable, esto quiere decir que puede calcularse en dos partes, cada una de las cuales involucra una transformada unidimensional.

En esta sección se tratará de forma general algunas transformadas que también tienen esta propiedad de separabilidad. Propiedad que permite crear algoritmos más eficientes, que a su vez logran un decremento del costo computacional de las transformadas.

6.2.1 TRANSFORMADA DISCRETA COSENO

La Transformada Discreta Coseno (DCT, por sus siglas en inglés) es una variación de la Transformada Discreta de Fourier. En esta transformada la imagen se descompone exclusivamente en sumas de cosenos.

La expresión matemática para la DCT bidimensional se define como:

$$C(u, v) = \alpha(u)\alpha(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cos\left(\frac{(2x+1)u\pi}{2M}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right), \quad 0 \leq u \leq M-1$$

$$0 \leq v \leq N-1$$

$$\alpha u = \begin{cases} 1/M, & u = 0 \\ \sqrt{2/M}, & 1 \leq u \leq M-1 \end{cases} \quad \alpha v = \begin{cases} 1/N, & v = 0 \\ \sqrt{2/N}, & 1 \leq v \leq N-1 \end{cases}$$

y la inversa como:

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \alpha(u)\alpha(v) C(u, v) \cos\left(\frac{(2x+1)u\pi}{2M}\right) \cos\left(\frac{(2y+1)v\pi}{2N}\right), \quad 0 \leq x \leq M-1$$

$$0 \leq y \leq N-1$$

$$\alpha u = \begin{cases} 1/M, & u = 0 \\ \sqrt{2/M}, & 1 \leq u \leq M-1 \end{cases} \quad \alpha v = \begin{cases} 1/N, & v = 0 \\ \sqrt{2/N}, & 1 \leq v \leq N-1 \end{cases}$$

Algunas consideraciones que se deben hacer con relación a esta transformada son:

- La transformada coseno es real y ortogonal³⁶.
- Esta transformada no es la parte real de la transformada de Fourier.
- Tiene una excelente compactación de energía para datos altamente correlacionados.
- Es una transformada rápida.

Como se puede ver, una de las características más importantes que tiene la DCT es su capacidad de compactación. Así, al aplicar esta transformada a una imagen, gran parte de la información visual significativa se concentra en sólo unos pocos coeficientes. Por tal razón su principal aplicación dentro del procesamiento de imágenes está en la compresión de video e imágenes estáticas; es utilizada en normas de codificación como JPEG, H.261, H.263, MPEG-1, MPEG-2 y MPEG-4, entre otras.

6.2.2 TRANSFORMADA DE WALSH

La transformada de Walsh es una transformada real por lo tanto no requiere ninguna función exponencial o trigonométrica para su cálculo y necesita menor capacidad de almacenamiento por píxel que otras transformadas.

La expresión matemática que la define es:

$$W(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) \prod_{i=0}^{k-1} (-1)^{b_i(x)b_{k-1-i}(u) + b_i(y)b_{k-1-i}(v)}$$

donde:

$u = 0, 1, \dots, N-1$, $v = 0, 1, \dots, N-1$ y $b_j(z)$ es el j -ésimo bit de la representación binaria de z .³⁷

³⁶ Una matriz ortogonal es necesariamente cuadrada e invertible: $A^{-1} = A^T$

La transformada inversa de Walsh es equivalente a la expresión anterior intercambiando W y f , esto se debe a que los vectores base para la transformada bidimensional inversa son idénticos a los de la directa.

$$f(x, y) = \frac{1}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} W(u, v) \prod_{i=0}^{k-1} (-1)^{b_i(x)b_{k-1-i}(u)+b_i(y)b_{k-1-i}(v)}$$

Aunque a simple vista las ecuaciones parezcan complejas, las funciones base de la transformada involucran sólo al producto de términos que son +1 o -1.

La transformada de Walsh se puede aplicar únicamente en imágenes cuyas dimensiones sean potencias enteras de 2.

Su principal aplicación dentro del procesamiento digital de imágenes es la compresión.

6.2.3 TRANSFORMADA DE HADAMARD

La transformada de Hadamard está estrechamente relacionada con la transformada de Walsh, incluso en ocasiones se las trata como una sola transformada (Walsh-Hadamard o WHT, por sus siglas en inglés).

La expresión matemática que representa a la transformada de Hadamard bidimensional es:

$$H(u, v) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} f(x, y) (-1)^{\sum_{i=0}^{k-1} b_i(x)b_i(u) + b_i(y)b_i(v)}$$

donde:

$f(x,y)$ es una imagen de dimensiones $N \times N$ y $N = 2^k$

$u = 0, 1, \dots, N-1$, $v = 0, 1, \dots, N-1$ y

$b_j(z)$ es el j -ésimo bit de la representación binaria de z .

³⁷ Por ejemplo si $n = 4$, $z = 5$ (0101 en binario) se tiene que $b_0(z) = 1$, $b_1(z) = 0$, $b_2(z) = 1$, $b_3(z) = 0$.

Su transformada inversa es:

$$f(x, y) = \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} H(u, v) (-1)^{\sum_{i=0}^{k-1} b_i(x)b_i(u) + b_i(y)b_i(v)}$$

Como se puede observar tanto la transformada directa como la inversa pueden calcularse con un único algoritmo. Así, si se aplica dicho algoritmo un par de veces sobre una imagen se obtiene como resultado la misma imagen.

Las propiedades más importantes de esta transformada son:

- Es real, simétrica y ortogonal.³⁸
- Es una transformada rápida. En Hadamard la imagen es descompuesta en funciones básicas que son esencialmente +1 y -1 en lugar de funciones trigonométricas como senos y cosenos que son más complejas, como es el caso de la transformada de Fourier. Por lo tanto el número de sumas y restas se reduce de N^2 hasta $N \log_2 N$ aproximadamente.
- A diferencia de la transformada de Fourier en la cual el espectro es la información más utilizada dentro del procesamiento de imágenes en Hadamard el logaritmo de la magnitud no tiene una interpretación física útil.
- Presenta una buena compactación de energía, por lo que es utilizada para compresión y filtrado de imágenes.

6.2.4 WAVELETS

Conocidas también como ondículas u ondeletas son funciones que a diferencia de la transformada de Fourier no solo tienen asociada frecuencia sino además una escala.

³⁸ $A=A^*=A^T=A^{-1}$

En la última década esta transformada ha generado gran interés dentro del campo de procesamiento de imágenes. Una de las principales razones es la flexibilidad que se tiene para trabajar con dicha transformada ya que existe un gran número de bases (wavelets) que se pueden utilizar dependiendo del tipo de aplicación al que esté orientado el procesamiento.

Otra razón está en su complejidad computacional que es lineal con el número de coeficientes calculados ($O(N)$), mientras que en otras transformadas aún en su implementación más rápida se tiene una complejidad de $N \log_2 N$. Adicionalmente, las wavelets permiten descomponer una imagen en versiones escaladas móviles de la wavelet original o wavelet madre, lo que permite un análisis multiresolución de la imagen.

La transformada wavelet discreta (DWT, por sus siglas en inglés) en su caso bidimensional es usualmente ejecutada mediante la aplicación de un filtro de paso bajo y un filtro de paso alto, seguidos de un submuestreo en un factor de 2 (reducción de la imagen) a lo largo de las dos dimensiones de forma separada, (véase Figura 1.25). De esta manera la imagen es descompuesta en sub-bandas que representan diferentes propiedades de la misma.

La convolución con el filtro de paso bajo da como resultado una imagen conocida como *aproximación* o *sub-banda LL* mientras que la convolución con el filtro de paso alto en direcciones específicas produce las *imágenes detalle* o *sub-bandas LH, HL y HH*. Las sub-bandas LH y HL registran los cambios en la imagen a través de las direcciones horizontal y vertical respectivamente y la banda HH muestra el componente de alta frecuencia de la imagen.

En cuanto a los filtros utilizados en esta transformada son generados empleando una función de escala para el filtro de paso bajo y una función wavelet para el filtro de paso alto y deben estar relacionados entre sí.

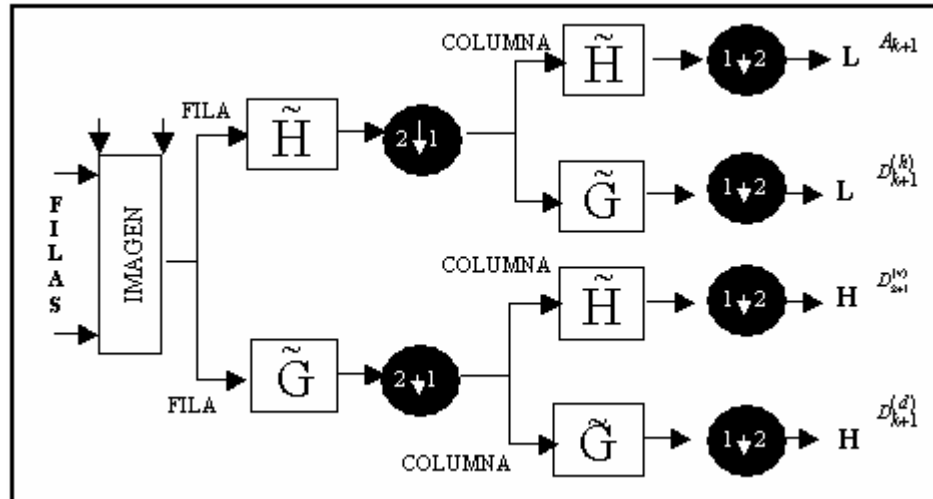


Figura 6.4. Transformada Wavelet discreta bidimensional

Fuente: POLIKAR, R. 2002.

La descomposición de una imagen a través de wavelets puede realizarse de dos formas:

- Descomposición clásica:* La imagen es descompuesta en una imagen aproximación e imágenes detalle (una por cada dirección en la que se aplica el filtro). En un segundo nivel sólo la imagen aproximación es descompuesta en una nueva imagen aproximación e imágenes detalle y así sucesivamente (proceso recursivo). Cada nivel se refiere a una escala, así la imagen original puede ser representada por un conjunto de subimágenes en algunas escalas.
- Descomposición de wavelets en paquete:* Es una generalización de la descomposición clásica, puesto que las imágenes detalle también son descompuestas de la misma manera que las aproximaciones. El resultado final se conoce como *árbol de wavelets en paquete*.

Las principales aplicaciones de la transformada wavelet son la compresión de imágenes³⁹, detección y reconocimiento de patrones, reducción de ruido, entre otros.

³⁹ Wavelets son el eje central del estándar JPEG-2000 así como del método WSC (cuantización escalar de wavelets) que es utilizado por el FBI para comprimir su base de datos de huellas dactilares.

Anexo 7

TRANSFORMADA DE HOUGH Y SU APLICACIÓN EN LA DETECCIÓN DE ELIPSES

7.1 CONSIDERACIONES DE LA TRANSFORMADA DE HOUGH

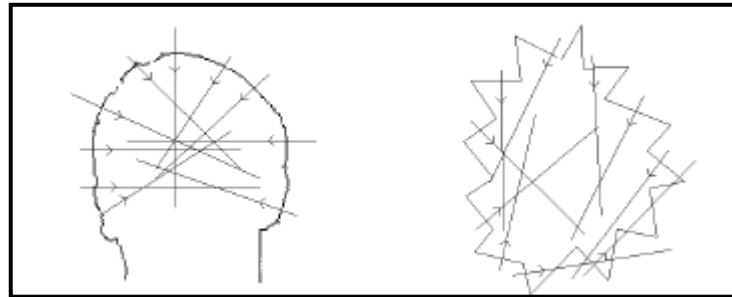
Existen diversos métodos para implementar la transformada de Hough, los mismos que están sujetos al tipo de patrón o grupo de patrones que se desean buscar en una imagen así como al tipo de aplicación para la cual se va a utilizar la transformada.

Sin embargo existen ciertas consideraciones generales que se deben tomar en cuenta al momento de trabajar con la transformada de Hough:

- Orientación del patrón: Existen patrones como las líneas rectas para los que la orientación es un parámetro que debe ser considerado mientras que en casos como la detección de círculos, la orientación no es necesaria.

En otros tipos de patrones, por ejemplo elipses, la inclusión de la orientación como un parámetro depende del tipo de aplicación para el se que requiere la transformada. Sin embargo se debe tener en cuenta para estos casos que si no se utiliza la orientación como un parámetro sólo se podrá encontrar objetos en una orientación específica, en tanto que si se utiliza esta información se pueden encontrar objetos con diferentes orientaciones pero implica un mayor requerimiento computacional.

- Normal del gradiente: Cuando la normal del gradiente es relevante se puede considerar esta información para reducir la complejidad de la transformada, (véase Figura 3.1).



(a)

(b)

Figura 7.1. Diferentes orientaciones de la normal del gradiente: (a) imagen en la cual la normal del gradiente se aproxima hacia el centro del objeto – Detección de elipses (b) imagen en la que la normal del gradiente no aporta información relevante.

Fuente: SEGUIER, R.; LE GLAUNEC, A.; BERNARD, L.. Francia. Pág. 2.

- Discretización del espacio de parámetros: El espacio de parámetros no es continuo por lo que la granularidad con la que se lo discretice determina cuán precisa será la localización de los objetos en la imagen.
- Normalización del Acumulador: Para lograr una cierta tolerancia en la detección de objetos cuyo contorno no tiene la forma exacta del patrón, ya sea por presencia de ruido u otros artefactos, se puede recurrir a un voto a nivel de áreas y no de puntos, es decir, cada punto de borde no genera un voto sólo para una celda del acumulador sino también para las celdas cercanas.

Otra forma de normalización es la de barridos en el acumulador una vez que se ha finalizado el proceso de votación, con el objetivo de fusionar celdas cercanas.

La forma en la que se normalice el acumulador depende de cuán cercanos pueden estar los objetos o cuántos objetos se van a detectar.

- DetECCIÓN DE MÁXIMOS: La forma en la que se seleccionan los máximos (celdas con mayor votación) en el espacio de los parámetros depende del número de objetos a detectar. Por ejemplo, si se desea detectar sólo un objeto entonces se puede seleccionar la celda con el valor máximo y luego mapear esta información al espacio de la imagen. Si se requiere detectar más de un objeto se deben encontrar máximos locales, para esto existen diferentes métodos descritos en la literatura.
- USO DE LA INTENSIDAD: Existen métodos que no consideran la magnitud del gradiente como parte de la transformada, es decir solo trabajan con la representación binaria de la imagen luego que se han detectado los bordes. En otros casos el voto es proporcional a la magnitud del gradiente en el punto de borde.

7.2 TRANSFORMADA DE HOUGH PARA DETECCIÓN DE ELIPSES

El tipo de detección más frecuentemente analizado en la literatura es la detección de líneas rectas, sin embargo en el presente trabajo se analizará la aplicación de la transformada de Hough para la detección de elipses.

Como se ha mencionado en apartados anteriores la transformada de Hough requiere información previa sobre el patrón que va a detectar, por esta razón se va a explicar el proceso de localización de elipses en su caso más simple, es decir considerando una elipse con una orientación específica a fin de reducir el número de parámetros que la describen.

De esta manera la elipse estará definida por cuatro parámetros: las coordenadas del centro (f,c) y sus semiejes (a,b) , (véase Figura 3.2).

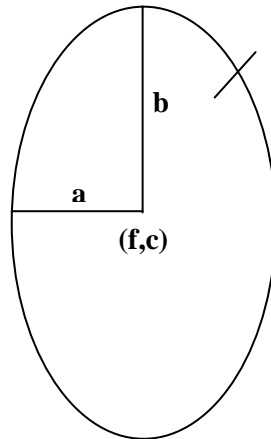


Figura 7.2. Parametrización de la elipse.

Cada parámetro debe ser cuantizado para así discretizar el nuevo espacio:

- Para el caso de los semiejes se asignan rangos discretos que están en función de los tamaños de las elipses que se desean detectar en la imagen. Mientras más cercanos sean los valores que componen los rangos se logrará una mayor precisión pero implicará un mayor costo computacional.
- En cuanto a las coordenadas del centro de la elipse pueden tomar valores que estén dentro del rango discreto asociado con las dimensiones de la imagen a analizar, es decir, si se considera a la imagen como una matriz o cuadrícula en la cual cada píxel ocupa una celda, entonces el centro de una elipse puede estar ubicado en cualquiera de estas celdas. De esta manera las coordenadas del centro de una elipse pueden ser las coordenadas de cualquiera de los píxeles de la imagen, considerando ciertas restricciones como en el caso de píxeles que están muy cercanos a los márgenes de la imagen ya que se buscan elipses que estén completas.

El espacio de parámetros tal como se ha definido es 4-dimensional lo que lo hace difícil de manejar a nivel computacional, por esta razón existen varias técnicas que han sido propuestas para reducir los requerimientos de tiempo y memoria que demanda el cálculo de operaciones en un espacio de este tipo.

Una de estas técnicas es dividir las operaciones a realizar dentro de una cascada: la votación se realiza en una estructura de dos dimensiones, (véase Figura 3.3), las mismas que corresponden a los dos primeros parámetros de la elipse (coordenadas del centro), en tanto que los dos parámetros restantes se mantienen fijos (semiejes de la elipse). Este proceso de votación se repite para cada pareja de valores discretos que representan a los semiejes de la elipse.

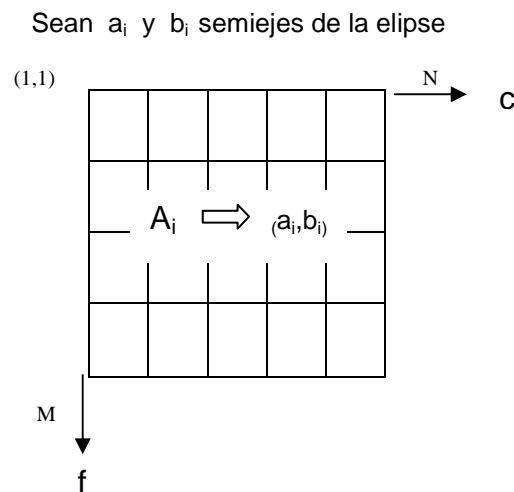


Figura 7.3. Célula acumulador A_i , sus dimensiones son las mismas que las de la imagen en la que se busca el patrón.

Así el proceso se reduce al cálculo de las coordenadas del centro de una elipse de semiejes conocidos y que pasa por un punto determinado (punto de borde).

Por cada combinación de valores de los semiejes de la elipse (a_i, b_i) se utiliza una nueva estructura o matriz en la que se acumulan los votos para esa combinación específica.

El conjunto de estas matrices compone el espacio de parámetros, por lo que una vez finalizado todo el proceso de votación se deben buscar los máximos en el espacio de parámetros. Estos máximos estarán ubicados en una celda de una matriz determinada, las coordenadas de la celda indicarán la ubicación del centro de la elipse en el dominio espacial, en tanto que la matriz a la que pertenece esa celda estará asociada a un par de valores que corresponden a los semiejes de la elipse.

Las ecuaciones utilizadas para el cálculo de las coordenadas del centro de la elipse se definen a continuación:

Sea a = semieje menor de la elipse,
 b = semieje mayor de la elipse,
 f_p = fila en la que está ubicado el punto de borde P ,
 c_p = columna en la que está ubicado el punto de borde P ,
 Δf_p y Δc_p = valores de los gradientes vertical y horizontal en el punto de borde P .

$$f_i^c = f_p \pm \operatorname{sgn}(\Delta f_p) \frac{b_i}{\sqrt{1 + \frac{a_i^2}{b_i^2} \left(\frac{\Delta c_p}{\Delta f_p} \right)^2}}$$

$$c_i^c = c_p \pm \operatorname{sgn}(\Delta c_p) \frac{a_i}{\sqrt{1 + \frac{b_i^2}{a_i^2} \left(\frac{\Delta f_p}{\Delta c_p} \right)^2}}$$

donde (f_i^c, c_i^c) son calculados por cada punto de borde para cada célula acumulador $A_i \rightarrow (a_i, b_i)$ independientemente y representan las coordenadas del centro de la elipse de semiejes (a_i, b_i) que pasa por el punto P .

El algoritmo que detalla este proceso es el siguiente:

1. Encontrar todos los puntos de borde o contornos de los objetos presentes en la imagen (Detección de bordes).
2. Por cada valor del semieje mayor
3. Por cada valor del semieje menor
4. Por cada punto de borde
5. Calcular las coordenadas del centro de la elipse que pasa por el punto de borde.
6. Incrementar el valor de la celda que tiene las coordenadas halladas. $A_i(f_c, c_c) = A_i(f_c, c_c) + 1$
7. Repetir los pasos 5 y 6 para cada par de valores de los semiejes, utilizando un nuevo acumulador A_j en cada iteración.
8. Encontrar máximos en el nuevo espacio (A_1, A_2, \dots, A_n)
9. Si se requiere, mapear cada máximo encontrado hacia el espacio de la imagen.

El algoritmo propuesto es sólo una forma de resolver el problema de detección de elipses, como se ha mencionado anteriormente existen otros métodos para aplicar la transformada.

De acuerdo a lo descrito en esta sección se puede notar que si bien el costo computacional de la transformada depende en gran medida del número de parámetros utilizados y de su cuantización, también influye el tamaño de la imagen a analizar y la cantidad de píxeles de borde de dicha imagen.

Anexo 8

[74] TURK, M.; PENTLAND, A. Eigenfaces for Face Detection/Recognition. Journal of Cognitive Neuroscience. Vol. 3. N° 1, pp. 71-86. 1991.

ESTANDARES DE COMPRESIÓN DE IMÁGENES DIGITALES

9.1 ESTÁNDARES DE COMPRESIÓN DE IMÁGENES ESTÁTICAS

Existen diferentes tipos de estándares de compresión para las imágenes estáticas, que se pueden clasificar de la siguiente manera:

- *Formatos bitmap:* TIFF, JPEG, GIF, PNG, PCX, BMP, PICT, PSD, RAW, TGA, entre otros.
- *Formatos vectoriales:* DXF, CDR, etc.
- *Formatos híbridos:* CGM, WMF, WPG.

En el presente documento se abarcará solo aquellos formatos que actualmente son los más utilizados para el procesamiento de imágenes:

TIFF (Tagged Image File Format): Es un formato flexible que puede trabajar con o sin pérdida de información. Es independiente de la plataforma hardware y del sistema operativo sobre el que se lo ejecute, además de ser el formato de mayor calidad en una cámara digital. Los detalles del algoritmo de almacenamiento de la imagen forman parte del fichero.

JPEG (Joint Photographic Experts Group): Este estándar utiliza técnicas de compresión con pérdidas, sin embargo permite obtener un nivel de compresión alto, manteniendo una elevada calidad de la imagen. Es

adecuado para imágenes de tonos continuos y con una gama de color muy amplia, trabaja con escala de grises a 8 bits y en color a 24 bits de profundidad.

GIF (Graphics Interchange Format): Este formato solo puede desplegar un máximo de 256 colores en una imagen. Cuando la imagen tiene menos de 256 colores GIF puede almacenarla sin pérdida, por el contrario cuando la imagen tiene una mayor cantidad de colores, este formato busca un conjunto de 256 colores adecuado para la imagen (reduce el número de colores).

PNG (Portable Network Graphics): Utiliza técnicas de compresión sin pérdida por lo que no existe variación en la calidad. Permite el manejo de grandes áreas de color uniforme y con más de 256 colores (soporta una profundidad máxima de 24 bits).

BMP (Bitmapped File Format): Es el formato de mapa de bits de Microsoft. No guarda las imágenes comprimidas, por lo que requiere un mayor espacio para almacenamiento.

DXF (Drawing Exchange Format): Es el formato vectorial utilizado por defecto en AutoCAD, trabaja con una profundidad de píxel de hasta 8 bits.

WMF (Windows MetaFiles): Este tipo de formato copia en un archivo los comandos para crear la imagen que se desea, logrando así un ahorro considerable de espacio.

9.2 ESTÁNDARES DE COMPRESIÓN DE VIDEO

Norma H.261: Es parte del grupo de estándares H.320, diseñado para una tasa de datos múltiplo de 64 Kbit/s. Soporta dos tipos de resolución 352 x 288 píxeles y 176 x 144 píxeles.

Este estándar está organizado de forma jerárquica en cuatro capas: imagen, grupo de bloques (GOB), macrobloques y bloques.

H.261 divide cada imagen en macrobloques de 16 x 16 píxeles, cada uno de estos macrobloques se predice de un cuadro anterior, estimando la cantidad de movimiento durante el tiempo transcurrido entre los cuadros.

MPEG (Moving Pictures Experts Group): Este formato utiliza información de varios cuadros para codificar las imágenes de una secuencia, tomando no solo la primera y la última imagen sino también diferentes imágenes de la misma secuencia en intervalos regulares.

MPEG-1 y MPEG-2 se basan en la estimación del movimiento entre cuadros sucesivos. MPEG-1 trabaja con una velocidad de transferencia de 1.5 Mbps y una resolución de 352 x 240 píxeles. Por el contrario MPEG-2 ofrece un mayor ancho de banda (entre 2 y 50 Mbps) y 720 x 486 píxeles de resolución.

MPEG-3 fue una propuesta de estándar para televisión de alta resolución sin embargo no está actualmente en uso ya que MPEG-2 mostró resultados similares. MPEG-4 utiliza un ancho de banda muy bajo y está orientado para videoconferencias sobre Internet.

En la actualidad se está trabajando sobre nuevos estándares MPEG, tales como: MPEG-7 que es un estándar de descripción de contenidos multimedia

para archivos digitales audiovisuales y MPEG-21 que es un estándar para integración de tecnologías multimedia.

Norma H.263: Este estándar de compresión de video definido por la UTI (International Telecommunication Union) fue desarrollado como una mejora al estándar H.261 pero considerando además otros estándares como MPEG-1 y MPEG-2.

Posteriormente se desarrollaron H.263+ y H.263++ que son versiones optimizadas del estándar para mejorar la transmisión de video.

Actualmente existe el estándar H.264 que presenta mejoras significativas sobre H.263 y está incluido en muchos de los nuevos productos para videoconferencias. Una de sus mayores ventajas es que ofrece una mejor calidad de imagen sin que esto involucre un mayor costo computacional.

Anexo 10

ESTUDIO PRELIMINAR

OBJETIVO

- Identificar el problema así como su entorno, considerando las limitaciones actuales que pueden influir en la definición de la solución.

10.1 ANTECEDENTES Y ESTADO ACTUAL DE LA TEMÁTICA

El reconocimiento de rostros⁴⁰ es un área de la biometría que ha sido estudiada desde los inicios de la visión por computador. A pesar de existir otras formas biométricas más confiables para la identificación (huellas digitales, escaneo de iris, etc.), el reconocimiento de rostros siempre ha generado un gran interés especialmente por ser poco intrusivo, por requerir una menor inversión para su implementación que otros sistemas biométricos, por aportar información adicional (por ejemplo el estado de ánimo), además de constituir la forma principal como las personas identifican a otras.

El primer sistema parcialmente automático para reconocimiento de rostros fue desarrollado en los años 60, éste requería la ubicación manual de puntos de referencia del rostro (ojos, boca, etc.) en la imagen que se le proporcionaba para así calcular distancias que luego servirían para ser comparadas con otros datos y de esta manera realizar el reconocimiento [76]. En 1991, Turk y Pentland descubrieron que la técnica de eigenfaces, a través del error residual, podía ser usada para detectar rostros en las

⁴⁰ Para mayor detalle sobre el proceso de reconocimiento y las metodologías utilizadas, véase Capítulo 1 , sección 1.3 en adelante.

imágenes. Este descubrimiento permitió sistemas confiables para reconocimiento automático de rostros en tiempo real.

En los últimos años el uso del reconocimiento automático de rostros como una forma de identificación ha ido en progreso, en gran parte gracias a los avances tecnológicos que han permitido crear algoritmos más complejos y robustos, incluso se ha logrado desarrollar sistemas para reconocimiento en tiempo real con técnicas diferentes a las propuestas inicialmente basándose en el gran potencial que tiene el video y en general las herramientas de procesamiento de imágenes con las que se cuenta actualmente. Así por ejemplo existen métodos que emplean el modelado de rostros en 3D como una forma de reconocimiento, otros se basan en la información temporal utilizando modelos probabilísticos [93], avances tecnológicos como el uso de fuentes de luz infrarroja han permitido el desarrollo de técnicas que detectan la distribución de la temperatura del rostro mediante sensores térmicos⁴¹, el uso de redes neuronales ha sido ampliamente difundido y analizado en especial por su gran poder de generalización, sin embargo esta técnica es sensible a ciertas variaciones como por ejemplo variación de luz en la imagen lo que provoca que el nivel de confiabilidad que ofrece no llegue al cien por ciento. Los algoritmos genéticos en combinación con otros métodos componen una nueva forma de reconocimiento de rostros, que si bien no ha sido extensamente analizada si existe en la literatura algunos trabajos que la emplean con este objetivo.

Los ejemplos antes citados son una muestra de la gran variedad de métodos que se pueden utilizar para llevar a cabo la tarea de reconocimiento de rostros, sin embargo ninguno de estos métodos es totalmente confiable y robusto. En cuanto a la detección de rostros, parte fundamental del reconocimiento automático, hasta mediados de los años 90 las investigaciones realizadas en esta área se enfocaban en la

⁴¹ YOSHITOMI, Y.; MIYAURA, T.; TOMITO, S.; KIMURA, S. Face Identification Using Thermal Image Processing. IEEE International Workshop on Robot and Human Communication, pp. 374-379. IEEE, 1997.

detección de un solo rostro en la imagen analizada, para esto se recurría a técnicas como plantillas, segmentación basada en el color de la piel, etc. En los últimos años se ha incursionado en la detección de múltiples rostros utilizando técnicas como redes neuronales (Rowley, 1998), SVM (Osuna, 1997), técnicas basadas en el movimiento, entre otras.

Debido al rápido crecimiento del mercado de tecnologías biométricas, dentro de las que se incluye el reconocimiento automático de rostros, la evaluación del desempeño de los métodos empleados en este campo ha adquirido una mayor importancia. Es así que actualmente existen varios esquemas que determinan cómo probar un método para medir su desempeño, algunos de ellos están dirigidos a estandarizar los mecanismos de evaluación, tal es el caso de la *prueba FERET* [55] que es una evaluación general diseñada para medir el desempeño en laboratorio de los diferentes métodos para reconocimiento de rostros, generalmente utilizada con la base de imágenes *FERET* que contiene diferentes categorías de imágenes (variaciones en iluminación, adquiridas en diferentes días, etc) y que fue creada con este fin aunque no existen restricciones para usar imágenes que no pertenezcan a esa base. Esta prueba no evalúa aspectos fuera del ámbito de laboratorio, es decir no considera factores como rapidez de ejecución, usabilidad, tiempo requerido para implementar el método empleado, relación entre confiabilidad y rapidez, entre otros aspectos que están más orientados hacia el dominio de la aplicación.

Otro protocolo de evaluación muy utilizado es el FRVT [30] que es un conjunto de métricas usadas para cuantificar el desempeño de los sistemas biométricos, tanto en el ámbito de laboratorio como del dominio de la aplicación. Existen esquemas menos formales a los que se les denomina *evaluaciones empíricas*, que son realizadas en muchas de las investigaciones que se pueden encontrar en la literatura y responden en general a formas de verificar el desempeño de un método específico de acuerdo al criterio de sus desarrolladores.

Usualmente todos los esquemas de evaluación utilizan conjuntos de imágenes de prueba para determinar el desempeño de los métodos de reconocimiento de rostros y expresan el resultado obtenido en base a dos métricas: FRR y FAR.

La probabilidad que un rostro conocido no sea identificado se denomina tasa de *rechazos falsos* o FRR. En tanto que la probabilidad que un rostro sea identificado como conocido sin serlo se denomina tasa de *reconocimientos falsos* o FAR. Durante las pruebas estos valores siempre están correlacionados, por lo tanto es necesario especificar ambos valores para evaluar el desempeño. El valor de estas métricas se determina de acuerdo al tipo de reconocimiento que se efectúa (identificación o verificación)⁴².

Para la identificación, el valor de FRR se define mediante el número de imágenes de rostros conocidos que no fueron identificados. Para FAR, su valor se define en relación al número de imágenes que fueron identificadas incorrectamente, es decir se les asignó una identidad que no les corresponde.

En el caso de la verificación, FRR se determina en base al número de verificaciones correctas realizadas en el conjunto de imágenes. Mientras que FAR se determina mediante el número de verificaciones falsas, para esto se deben utilizar imágenes de rostros y proporcionar identificaciones que no correspondan con esas imágenes, es decir simular la verificación de un impostor. Los resultados de desempeño pueden ser presentados de diversos modos como por ejemplo diagramas de frecuencia, curvas FAR-FRR, diagramas ROC, entre otros.

En conclusión, a pesar de los avances realizados en el campo de reconocimiento de rostros, éste tiene hoy por hoy muchas limitaciones y no

⁴² En la identificación el algoritmo recibe la imagen de un rostro y debe retornar la identidad de la persona en la imagen, mientras que en la verificación el algoritmo recibe una imagen y una identificación y debe retornar como resultado si la identificación proporcionada realmente corresponde a la imagen. Véase Capítulo 1, Sección 1.3.5.2.

se ha logrado desarrollar un sistema que realice esta tarea de manera confiable bajo diferentes condiciones y circunstancias, tal como lo haría un ser humano. Además se debe tener en cuenta que factores como el tipo de reconocimiento a realizar, los métodos seleccionados, el tamaño de la base de imágenes utilizada, el tipo y calidad de las imágenes, entre otros factores, influyen en los resultados de desempeño; por lo tanto una solución que trabaja bien bajo un conjunto de condiciones puede no presentar el mismo desempeño bajo otras condiciones.

Información más detallada sobre el estado de arte del reconocimiento y detección de rostros se encuentra en [44, 76, 81].

10.2 IDENTIFICACIÓN DEL PROBLEMA

10.2.1 Problemas y Limitaciones de la Realidad Actual

Debido a la naturaleza dinámica de las imágenes de rostros humanos, el proceso de reconocimiento enfrenta diversos problemas, tales como:

- *Posición del Rostro (rotación)*

Es el factor que afecta más significativamente la apariencia del rostro, es por esta razón que actualmente la mayoría de las aplicaciones de reconocimiento utilizan imágenes que abarcan el rostro completo visto de frente. De esta manera se obtienen todas las características representativas del rostro, lo que permite alcanzar un mayor porcentaje de aciertos. Recientemente se están desarrollando técnicas para el reconocimiento a partir del perfil del rostro, pero son técnicas mucho menos confiables que las actuales ya que están en etapa experimental.

- *Escala*

La distancia focal entre el rostro de una persona y una cámara puede variar de una imagen a otra, mientras la distancia sea menor la imagen del rostro será más grande. Esto provoca imágenes de rostros a diferente

escala y por lo tanto es un problema que puede influir en el proceso de reconocimiento.

- *Variaciones en la Iluminación*

Los cambios de iluminación (intensidad y dirección) también afectan el proceso de reconocimiento de rostros, una persona bajo distintas condiciones de iluminación puede verse completamente diferente.

El asegurar condiciones estables puede reducir el riesgo de errores durante el proceso de reconocimiento. Si esto no es posible se deben considerar al menos aquellas variaciones que afectan en mayor grado al reconocimiento y evitarlas. De acuerdo con el documento "*Face Recognition Format For Data Interchange*"⁴³, se debe evitar sombras fuertes en los lados de la cara así como en el área de los ojos.

Las variaciones de iluminación no solo pueden afectar el reconocimiento sino también pueden influir en el proceso de detección de rostros. No obstante existen técnicas para reconocimiento bajo variación de luz, tal es el caso del método PCA compuesto, que es utilizado cuando existe una variación considerable en la iluminación, en las expresiones faciales y también existen pequeñas variaciones en la posición de la cabeza. Sin embargo este método requiere de implementación a nivel de hardware para lograr un tiempo de procesamiento razonable.

Muchos métodos de reconocimiento buscan minimizar el efecto de la iluminación mediante preprocesos de la imagen, técnicas como la modificación del histograma o corrección de gama se emplean frecuentemente con este objeto. Pese a los resultados que se han obtenido en diversos trabajos con relación a la corrección de iluminación, actualmente no se cuenta con técnicas de corrección completamente adaptivas (que realicen una corrección adecuada en función del grado de iluminación que cada área de la imagen presenta).

⁴³ GRIFFIN, Paul. FACE RECOGNITION FORMAT FOR DATA INTERCHANGE. Jersey – EUA. 2003.

- *Resolución de la Imagen*

La resolución de la imagen puede afectar los resultados del reconocimiento o influir en el tiempo de procesamiento. A una mayor cantidad de píxeles por pulgada la imagen tendrá una mejor calidad y por ende su definición será mejor pero esto también puede significar mayor tiempo de análisis de la imagen. Por otro lado una imagen con baja resolución puede no ser suficientemente clara para el reconocimiento.

Por esta razón la resolución con la que se va a trabajar depende de las características del método empleado y el seleccionar la resolución adecuada constituye todavía un problema. Sin embargo existen referencias⁴⁴ que definen que la tecnología de reconocimiento de rostros trabaja bien con gran parte de las cámaras de PC, generalmente requiriendo una resolución de 320 x 240 en imágenes estáticas y en imágenes dinámicas con 3 a 5 cuadros por segundo con la misma resolución.

- *Cambios en la apariencia de las personas*

Otro de los mayores problemas para el reconocimiento de rostros son los cambios en la apariencia. Una persona puede cambiar drásticamente su aspecto en periodos de tiempo muy cortos (por ejemplo de un día para otro), el uso de accesorios como lentes, gafas, gorras, diferentes peinados, etc. Todos estos cambios pueden hacer que el reconocimiento de una persona sea muy difícil de efectuar.

- *Diferentes expresiones faciales y oclusiones*

Los cambios en las expresiones faciales o la oclusión de un área del rostro también pueden afectar el proceso de reconocimiento.

- *Presencia de ruido en las imágenes*

Algunos dispositivos de captura como por ejemplo las cámaras web pueden generar cierto ruido en las imágenes. La presencia de ruido afecta

⁴⁴ HENRY, Verónica. BIOMETRICS: FACE RECOGNITION TECHNOLOGY. SANS Institute. 2000–2002.

directamente al proceso de reconocimiento de rostros ya que altera en algún grado el contenido de la imagen.

De manera general las variaciones o cambios antes mencionados pueden ser tan grandes entre imágenes de la misma persona como las variaciones que existen entre una persona y otra [53], es por esta razón que constituyen los principales problemas para el reconocimiento de rostros (tienen un impacto significativo en el reconocimiento). Adicionalmente existen limitaciones relacionadas a los métodos que actualmente son empleados, como por ejemplo: el número de imágenes que se pueden obtener para el proceso de reconocimiento es habitualmente pequeño (métodos como las redes neuronales requieren de un mayor número de imágenes de entrenamiento para lograr un mejor desempeño), la capacidad de procesamiento de los sistemas computacionales actuales que no permiten procesamiento paralelo para lograr respuestas en tiempo real para ciertos métodos.

En conclusión los actuales sistemas para reconocimiento de rostros no son totalmente confiables bajo diversas condiciones (iluminación, movimiento, gestos, rotación, etc.) y presentan limitaciones derivadas de los métodos que emplean para llevar a cabo el proceso de reconocimiento.

10.2.2 Definición del Problema

¿Cómo realizar el reconocimiento automático de un rostro en una imagen digital con un nivel de confiabilidad adecuado, empleando métodos ya existentes?

Anexo 11

ETAPA EXPERIMENTAL DE RECONOCIMIENTO

OBJETIVO

- Identificar el conjunto de métodos o técnicas más adecuado para llevar a cabo el reconocimiento automático de rostros en el contexto de la solución computacional propuesta.

11.1 ESQUEMA EMPLEADO PARA EL RECONOCIMIENTO

Establecer el esquema que se utilizará para realizar el reconocimiento automático de rostros permitirá definir claramente los subprocesos que se deberán llevar a cabo como parte del proceso de reconocimiento.

Cada subproceso tiene un objetivo diferente y puede efectuarse mediante diversos métodos o técnicas, algunas de las cuales serán analizadas en esta etapa experimental.

Considerando la referencia teórica provista en el Capítulo 1, sección 1.3 del presente trabajo se establecerá que el esquema a seguir para realizar el reconocimiento automático de rostros constará de dos etapas:

- Enrolamiento
- Reconocimiento

De igual manera cada etapa constará a su vez de un conjunto de fases:

- Enrolamiento
 - Preprocesamiento
 - Extracción de Características
 - Entrenamiento

- Reconocimiento
 - Preprocesamiento
 - Extracción de Características
 - Verificación

La Figura 11.1 muestra el esquema planteado para la solución computacional a desarrollarse.

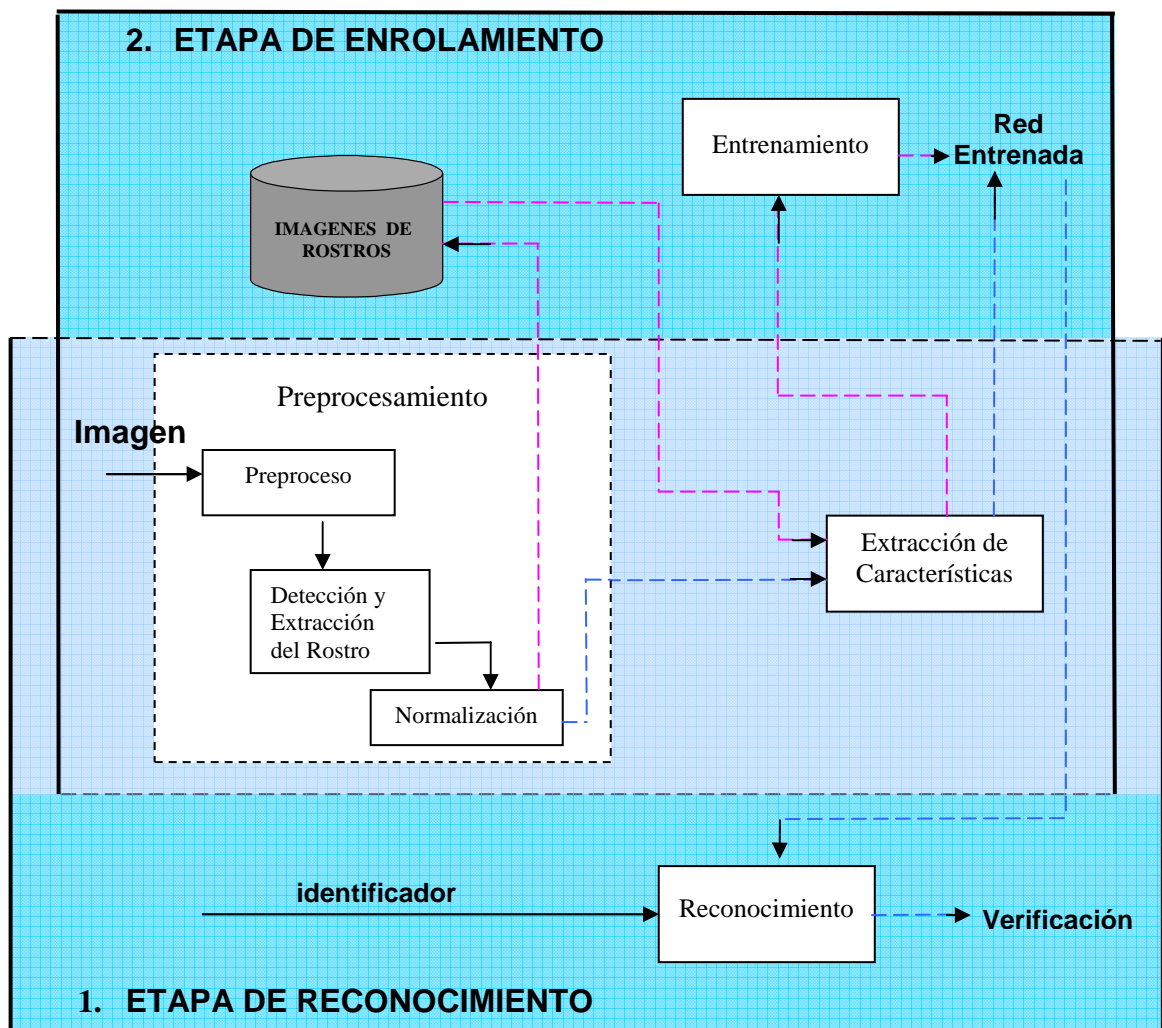


Figura 11.1. Esquema para el Reconocimiento automático de Rostros

Como se puede observar en la Figura 11.1 la etapa de enrolamiento consistirá en el almacenamiento de las imágenes de los rostros de los usuarios. Estas imágenes serán previamente sometidas a un preproceso, extraídas de imágenes más grandes a fin de eliminar elementos irrelevantes para el reconocimiento y finalmente normalizadas.

Una vez que las imágenes de todos los usuarios han sido almacenadas se extraerá las características más relevantes de cada imagen, estas características constituirán el patrón que representará a esa imagen, por lo tanto cada imagen tendrá asociado un patrón. El conjunto de patrones de las imágenes almacenadas será enviado a una red neuronal para entrenarla. De esta manera concluye el proceso de enrolamiento.

En cuanto al reconocimiento, éste consistirá en preprocesar una imagen, localizar el rostro que contiene, extraerlo y normalizarlo. Una vez que se cuente con la imagen sólo del rostro se extraerán las características más relevantes de éste y se las enviará a la red neuronal para que identifique a qué patrón de los utilizados durante el entrenamiento se parece más el nuevo patrón.

La red enviará el resultado y utilizando el identificador inicialmente proporcionado por el usuario se decidirá si el usuario es quien dice ser. Así termina el proceso de reconocimiento (en este caso de verificación).

De la descripción anterior se puede concluir que existen dos fases comunes entre el enrolamiento y la verificación, éstas son:

- El preprocesamiento de una imagen cuyo objetivo final para este caso específico es la obtención de una imagen normalizada del área de un rostro.
- La extracción de características relevantes de un rostro previamente localizado (para el enrolamiento la extracción será realizada para cada

imagen del conjunto almacenado en tanto que para la verificación sólo se realizará para una imagen).

Por lo tanto el siguiente análisis se enfocará en tres fases bien diferenciadas:

1. Preproceso de una imagen con el objetivo de obtener el área del rostro contenido en ésta.
2. Extracción de características relevantes de un rostro.
3. Entrenamiento de una red neuronal para la tarea de verificación de rostros.

Se considerará sólo estas fases por ser las más complejas dentro del proceso de reconocimiento automático de rostros.

11.2 MÉTODOS PROPUESTOS

Como se ha mencionado anteriormente los métodos empleados para realizar el reconocimiento automático de rostros son muy variados y cada uno presenta ventajas y desventajas con respecto a los demás, adicionalmente hoy por hoy no se cuenta con métodos totalmente confiables y robustos bajo diferentes condiciones (cambios en iluminación, diferentes expresiones faciales o rotaciones del rostro, etc.)

Esto lleva a plantear restricciones que permitan reducir la variabilidad en las imágenes a fin de limitar los aspectos que puedan influir en el proceso de reconocimiento de rostros. Dichas restricciones fueran planteadas como parte del análisis de requerimientos efectuado en el Capítulo 2, sección 2.1.

A manera de síntesis se recogen las restricciones más relevantes⁴⁵:

- La imagen debe contener un rostro completo y visto de frente.

⁴⁵ Para un mayor detalle sobre las características que deben tener las imágenes véase Capítulo 2, sección 2.1

- Cada imagen debe contener el rostro de una sola persona.
- No se deben utilizar fondos complejos.
- La iluminación en la imagen de ser normal y uniforme.
- Se admiten cambios en expresiones faciales pero no contorsiones bruscas.

Estas y otras restricciones permiten limitar el espectro de posibles problemas que se deben manejar durante el reconocimiento de rostros y por ende delimitar mejor las opciones de métodos que pueden ser empleados.

A continuación se realiza un análisis superficial de algunos métodos considerados dentro de esta etapa experimental de reconocimiento para cada fase identificada como relevante. Cabe destacar que varios de éstos métodos fueron implementados como prototipos para probar su desempeño mientras que otros sólo se analizaron en función de las referencias existentes en la literatura.

11.2.1 Preproceso de la imagen

En el contexto de la solución computacional propuesta se entenderá por preproceso de la imagen al conjunto de tareas que permitan obtener como resultado una imagen que contenga sólo el área del rostro, normalizada en cuanto a iluminación, orientación del rostro y tamaño de la imagen.

Por lo tanto se debe separar las diferentes tareas que este subproceso involucra:

- Corrección de iluminación en la imagen inicial
- Localización del rostro
- Normalización de la imagen para corregir orientación del rostro
- Extracción del rostro y normalización de la nueva imagen en cuanto a tamaño.

De igual manera se considerará solo las tareas que son más complejas.

11.2.1.1 Corrección de iluminación en la imagen inicial

A pesar de establecer ciertas características que deben cumplir las imágenes es difícil lograr tener un ambiente totalmente controlado durante la adquisición de dichas imágenes. Por esta razón es conveniente emplear algún método que permita mejorar de alguna manera la imagen en cuanto a la iluminación presente en ésta.

No existen métodos universales, por lo tanto cada método asume presunciones diferentes en relación a la iluminación y realizan la corrección en función de estas suposiciones. Por ende el objetivo es determinar el método o métodos que mejor se acoplen a los requerimientos de la solución computacional.

- Gray World

Este método se basa en la hipótesis que el color promedio que se refleja sobre las superficies es acromático, es decir es gris. Por lo tanto los canales en una imagen RGB con una correcta distribución de colores, deben tener igual promedio.

De acuerdo a la literatura el método Gray World es efectivo si se desea corregir la iluminación de una imagen capturada en condiciones cercanas a una iluminación estándar, mientras que en casos de condiciones extremas de iluminación (lugares con poca iluminación por ejemplo) no es de gran utilidad.

En esta etapa experimental se empleó este método (en su forma estándar) para corregir la iluminación en imágenes que no presentaban condiciones extremas de iluminación, sin embargo se observó que si bien en algunos casos realiza una buena corrección tiende en general a sobre compensar el canal B (azul) de las imágenes cuando éstas tienen áreas con una mayor iluminación (sin

ser extrema). Mientras que en imágenes con poca iluminación tiende a sobre compensar la iluminación de toda la imagen.

- White Patch Retinex

Este método es una modificación de Gray World, se basa en la hipótesis que la mayor intensidad que existe en una imagen debería corresponder al color blanco, por ende reescala linealmente cada canal en una imagen RGB considerando a la mayor intensidad encontrada como blanco.

De acuerdo a la literatura este método es más efectivo para imágenes con poca iluminación pero en imágenes con una suficiente iluminación o con exceso de iluminación no realiza una corrección adecuada.

- Compensación adaptativa de luz [21]

Es una adaptación de Gray World, es una técnica particular propuesta como parte de un trabajo de investigación para desarrollar un detector de regiones de color de la piel para la extracción de rostros.

En este método se utiliza el modelo de color RGB normalizado en lugar del modelo RGB. Además se consideran los píxeles que no tienen color negro con el objetivo de solucionar el problema de sobre compensación que realiza el método estándar para imágenes con poca iluminación. Calcula también el valor máximo y mínimo para cada píxel en el modelo RGB normalizado con el objetivo de obtener un valor adaptativo de gris promedio para toda la imagen.

En esta etapa experimental se empleó este método sin embargo en las pruebas realizadas se observó que al igual que el método estándar tiende a sobre compensar el canal azul de las imágenes cuando éstas tienen áreas con mayor iluminación.

- Normalización de Iluminación [71]

Técnica propuesta como parte de un algoritmo para la detección de ojos mediante información del color y funciones de proyección. Consiste en normalizar la iluminación de una imagen mediante una operación de estiramiento del histograma. Como paso previo a la modificación del histograma se calcula la suma cumulativa de éste y se descartan los píxeles que están por debajo del 0.1% de la suma cumulativa. Finalmente el histograma residual es estirado sobre el rango completo. Esto se ejecuta por separado para cada canal de la imagen RGB.

Esta técnica fue empleada dentro de la etapa experimental mostrando buenos resultados. Sin embargo puede provocar errores al eliminar valores de luminosidad que si bien no están presentes en alto grado en la imagen son importantes, por ejemplo en el caso de una imagen del rostro de una persona con ojos color verde, ésta técnica elimina el aporte del canal G (verde) en esta área por ser muy bajo dando como resultado que el área de los ojos tome un color totalmente diferente (tiende a ser violeta).

- Compensación de Iluminación [36]

Esta técnica es desarrollada como parte de un algoritmo para la detección de rostros en imágenes a color. Consiste en definir un valor como “blanco de referencia” para luego ajustar los canales R, G, B de una imagen en función de este valor. Para definir el blanco de referencia se selecciona los píxeles que estén sobre el 95% del rango absoluto de la luminancia de la imagen. La luminancia se refiere al canal Y (luminancia con corrección de gamma) del modelo YCbCr.

De acuerdo a esta técnica, sólo se aplica la compensación de iluminación si el número de píxeles que se encuentran dentro de este rango es mayor a 100. En tal caso, el blanco de referencia se

calcula como el promedio del conjunto de píxeles que están en el rango antes definido y los canales R, G, B de la imagen son ajustados de modo que el blanco de referencia sea linealmente escalado al blanco absoluto.

Esta forma de compensación ofrece buenos resultados en imágenes con variación en la iluminación, sin embargo en ambientes externos donde no existe ningún control de la iluminación esta técnica puede no resultar adecuada.

Considerando las restricciones establecidas para el presente proyecto se utilizó esta técnica durante la etapa experimental, obteniendo buenos resultados, sin embargo existen ciertos aspectos que pueden ser mejorados como por ejemplo el considerar el número de píxeles que deben estar como mínimo dentro del rango de luma como un valor en función del tamaño de la imagen y no un valor fijo (en la literatura este valor es fijo: 100 píxeles)

La Figura 11.2 muestra una imagen corregida su iluminación con algunas de las técnicas o métodos antes mencionados.

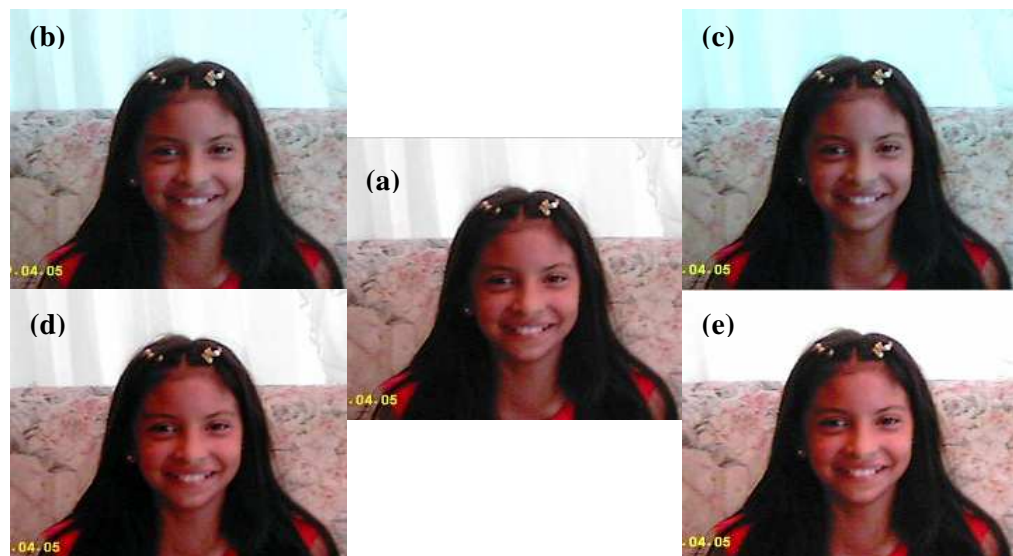


Figura 11.2. Corrección de la iluminación de una imagen utilizando diferentes técnicas (a) Imagen Original (b) Gray World (c) Compensación adaptativa de luz (d) Normalización de Iluminación (e) Compensación de Iluminación (con modificaciones a la técnica original).

11.2.1.2 Localización del Rostro

La localización de un rostro en una imagen es una de las tareas más complejas dentro del proceso de reconocimiento automático de rostros además de ser una de las más importantes.

De acuerdo a la literatura se entiende por localización de rostros a una especialización de la detección de rostros, en la cual se conoce a priori el número de rostros que existen en la imagen.

Para el presente proyecto el número de rostros a detectar por imagen es conocido, puesto que se limitó este número a un rostro por imagen. De tal manera que el análisis se centrará en métodos para localizar un rostro en una imagen y no múltiples rostros.

- o Localización de áreas del color de la piel

Una de las técnicas más comúnmente utilizadas como parte de la localización de rostros es el uso de la información del color para hallar regiones en una imagen que correspondan a posibles áreas de la piel, esto se basa en la hipótesis que el color de la piel de todos los seres humanos sin importar la raza, es similar en cuanto a la crominancia y lo que varía ampliamente es la luminancia. Por lo que el definir un modelo adecuado para el color de la piel es la estrategia primordial de esta técnica.

El objetivo de definir un modelo es determinar una forma para discriminar entre píxeles que pueden pertenecer a áreas de piel de aquellos que no. Las formas para definir un modelo son básicamente cuatro: Definir explícitamente las regiones de la piel, modelado no paramétrico de la distribución del color de la piel, modelado paramétrico de la distribución del color de la piel y modelados dinámicos de la distribución del color de la piel.

En el presente trabajo se utilizó la primera forma citada para definir un modelo para el color de la piel, es decir definir explícitamente las regiones de la piel. De esta manera cada píxel de la imagen se clasifica independientemente utilizando umbralización.

La principal razón para utilizar esta forma de modelado es la posibilidad de construir un algoritmo para una rápida localización de rostros debido a la simplicidad de las reglas que se utilizan para discriminar entre píxeles de piel de aquellos que no lo son. Sin embargo la principal dificultad está en determinar un espacio de color adecuado y la definición empírica de las reglas de decisión.

En la literatura existe una gran variedad de análisis sobre qué espacio de color es en el que mejor se compactan los píxeles relacionados al color de la piel, ya que al formar grupos compactos se puede establecer de manera más confiable rangos de valores que permitan eliminar de una imagen todas aquellas regiones que no representen potenciales áreas de la piel. Sin embargo estos análisis presentan diversas conclusiones y no existe un consenso sobre el espacio de color más adecuado para esta tarea.

El determinar valores útiles para realizar la detección en función del color requiere disponer de información representativa, caso contrario se pueden establecer valores que no constituyan la real distribución del color de la piel en un espacio concreto. Por ejemplo en [71] se menciona que los valores utilizados para realizar el proceso de segmentación se escogieron luego de una gran cantidad de experimentos en una amplia variedad de imágenes de rostros de personas de diferentes razas.

Por esta razón se consideró conveniente recurrir a investigaciones ya realizadas y en base a estas experiencias definir qué regla de decisión sería la más adecuada para este proyecto. En la literatura

se encontró algunos trabajos en los que se presentan las reglas de decisión halladas experimentalmente, éstas están definidas en tres espacios de color diferentes:

1. RGB: Este espacio se presenta como una buena opción para la detección de píxeles de piel cuando las condiciones de iluminación son considerablemente buenas, sin embargo muchos trabajos que analizan la utilidad de este espacio para la detección de áreas de piel argumentan que debido a la alta correlación entre sus canales y a la mezcla de crominancia y luminancia que tiene este espacio, no es una opción favorable para emplearla como parte de un algoritmo de localización de rostros.

En [80] se especifica la siguiente regla de decisión basada en el espacio RGB:

(R,G,B) se clasifica como piel si:
 $R > 95$ y $G > 40$ y $B > 20$ y
 $\max\{R,G,B\} - \min\{R,G,B\} > 15$ y
 $|R-G| > 15$ y $R > G$ y $R > B$.

2. YCbCr: La información de la luminancia está dada por el canal Y mientras que la crominancia se describe mediante los canales Cb y Cr. Esta separación de luminancia y crominancia, según varias investigaciones, hace de este espacio de color bastante apropiado para ser utilizado en la detección de áreas de la piel. Además de esto, los canales Cb y Cr se encuentran altamente correlacionados para los píxeles de piel.

En la literatura se encontró dos reglas de decisión:

YCbCr se clasifica como piel si:

- $102 < Cb < 128$ y $125 < Cr < 160$
- $105 < Cb < 135$ y $140 < Cr < 165$

3. RGB normalizado: Al ser este espacio bidimensional a diferencia de la mayoría de espacios de color existentes permite reducir el tiempo de procesamiento requerido para la detección de áreas de piel en una imagen. Además de acuerdo a varios trabajos descritos en la literatura los píxeles de piel se muestran más compactos en este espacio que en otros espacios.

Otra característica importante de este espacio es la normalización que efectúa del espacio RGB original, que permite obtener componentes con una menor dependencia de la iluminación.

En [71] se establece la siguiente regla de decisión para este espacio:

$$(norm_r, norm_g) \text{ se clasifica como piel si:} \\ 90 < norm_r < 145 \quad \text{y} \quad 60 < norm_g < 100$$

Las reglas de decisión antes mencionadas fueron empleadas dentro de esta etapa experimental, considerando imágenes con una adecuada iluminación.

La regla definida para el espacio RGB muestra buenos resultados aunque tiende a eliminar áreas de la piel que en algunos casos pueden ser relevantes, la primera regla definida para el espacio YCbCr no realiza una buena discriminación especialmente con regiones claras que las considera como áreas de piel, mientras que la segunda regla definida para este espacio es más rígida y puede eliminar áreas mayores de piel impidiendo una correcta localización de estas áreas en una imagen. En cuanto a la regla definida para el espacio RGB normalizado presenta buenos resultados y el tiempo de procesamiento que requiere es menor en comparación con las

demás reglas (considerando todas las operaciones que involucran inclusive transformaciones entre espacios).

Finalmente se debe mencionar que un factor importante para la selección de una u otra regla es su grado de flexibilidad, es decir se requiere que la regla de discriminación elimine dentro de lo posible la mayor cantidad de píxeles que no pertenecen a las áreas de la piel pero que a su vez no elimine regiones que realmente formen parte de la piel. Este análisis debe ser realizado experimentalmente.

La Figura 11.3 muestra un ejemplo que forma parte del análisis realizado con las reglas de decisión citadas.

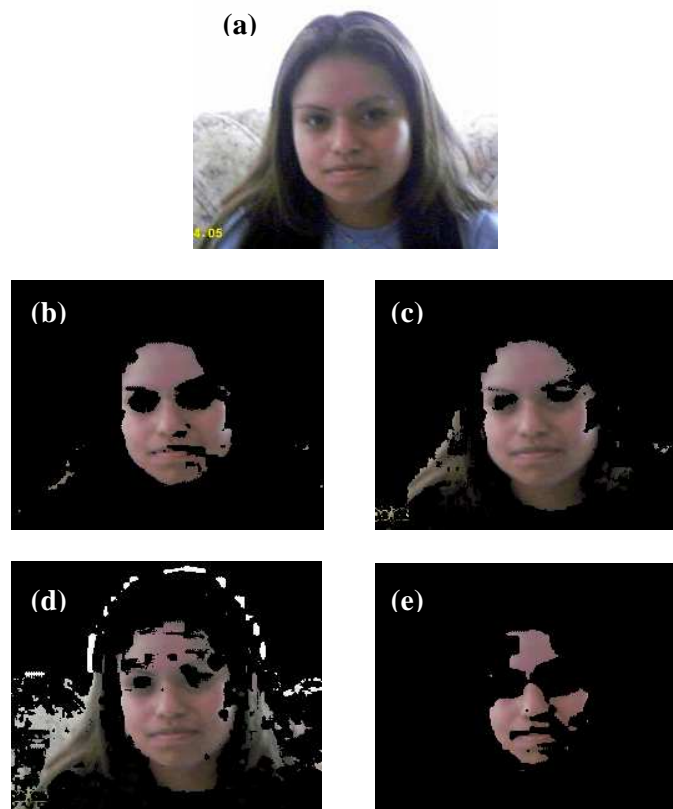


Figura 11.3. Detección de áreas de la piel en diferentes espacios de color

(a) Imagen Original (b) Regla en RGB (c) Regla en RGB normalizado (d) Regla en YCbCr : $102 < Cb < 128$ y $125 < Cr < 160$ (e) Regla en YCbCr: $105 < Cb < 135$ y $140 < Cr < 165$.

- o Caracterización del rostro por su contorno

Las técnicas que caracterizan al rostro por su contorno son aquellas que utilizan generalmente patrones geométricos para buscar en la imagen figuras que encajen con éstos y que puedan representar el perímetro del rostro. Estas técnicas suelen ser empleadas conjuntamente con otras técnicas para validar que la figura detectada es realmente un rostro.

Se analizaron dos técnicas de este tipo:

1. Localización de Rostros usando el método de gradiente [42]

Este algoritmo utiliza el método de gradiente para detectar objetos ovalados. Su objetivo es localizar en una imagen el contorno de un rostro utilizando como patrones de búsqueda circunferencias de distintos radios⁴⁶. Para lograr que el contorno de un rostro sea ovalado en lugar de elíptico reescala la imagen usando diferentes proporciones a las que ésta tiene.

Se implementó el algoritmo como parte de esta etapa experimental, sin embargo no se logró obtener resultados significativos para la localización de un rostro en una imagen.

Si bien tiene como ventaja el requerir menos tiempo de procesamiento que otros algoritmos de este tipo debido a que utiliza como patrón de búsqueda una circunferencia en lugar de una elipse, esto también provoca que cometa más errores al localizar rostros. No todas las personas tienen el contorno del rostro completamente elíptico o la forma del cabello también puede influir en la distorsión del contorno real por lo que al realizar el reescalamiento de la imagen en la forma propuesta, el contorno del rostro en ocasiones toma una forma diferente a la ovalada y se producen detecciones incorrectas.

⁴⁶ Para mayor detalle sobre el principio de esta técnica, véase Capítulo 1, sección 1.1.8.3

2. Transformada de Hough para la detección de elipses

En la literatura existen varios trabajos sobre detección de rostros que utilizan esta técnica [56, 77], algunos emplean imágenes de rostros vistos de frente mientras que otros adicionan el uso de imágenes de perfiles.

El **Anexo 7** presenta una breve descripción sobre la aplicación de la transformada de Hough para la detección de elipses. Esta es la base sobre la que se realizó los experimentos en esta etapa.

Inicialmente se realizaron dos pruebas. En la primera prueba se utilizaron 16 imágenes de la base de rostros ORL⁴⁷ (imágenes en escala de grises) y 30 imágenes a color obtenidas de Internet. El resultado de la localización de rostros fue 45% de localizaciones correctas, 13% localizaciones parciales (no se localizó el área completa del rostro) y el 42% de fallos (no se localizó el área del rostro en la imagen). En la Figura 11.4 se muestran algunos resultados de esta prueba.

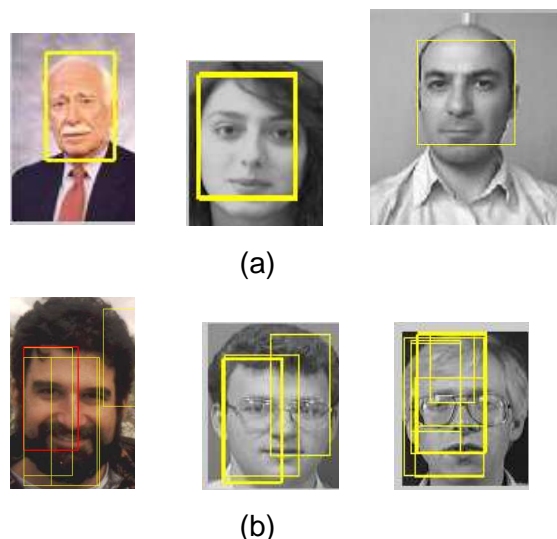


Figura 11.4. Prueba inicial para la detección de rostros mediante la transformada de Hough

(a) Ejemplos de localizaciones correctas (b) Ejemplos de fallos

⁴⁷ Base de rostros del laboratorio de Investigación Olivetti. 1994. <http://www.cam-orl.co.uk/facedatabase.html>

La segunda prueba se realizó luego de implementar ciertas mejoras al algoritmo original. Se utilizó 59 imágenes, algunas obtenidas de Internet y otras son imágenes de bases de rostros. Cabe destacar que el conjunto de imágenes no tiene todas las características requeridas para este proyecto, es decir existen imágenes con rostros con cierto grado de rotación o el área del rostro es muy pequeña con relación al tamaño de la imagen. El resultado de la localización de rostros fue 66% de localizaciones correctas y 34% de fallos. En la Figura 11.5 se muestran algunos resultados de esta prueba.



Figura 11.5. Segunda prueba realizada para la detección de rostros mediante la transformada de Hough

(a) Ejemplos de localizaciones correctas (b) Ejemplos de fallos

Los resultados obtenidos son aceptables y sirven como punto de partida para desarrollar posteriormente un algoritmo más confiable, sin embargo el tiempo de procesamiento es un factor que se debe considerar por lo tanto se debe buscar formas de optimizar el algoritmo original.

- **Análisis de características faciales**

Existen métodos que emplean características locales para detectar rostros en imágenes. Estas características pueden ser ojos, boca, nariz, cejas, entre otros.

A continuación se detallan las técnicas investigadas dentro de esta etapa experimental.

1. Localización del área de los ojos utilizando un filtro de varianza

Esta técnica forma parte de un algoritmo para detección de ojos en imágenes en escala de grises utilizando múltiples claves de detección [26]. Para su aplicación el área del rostro debe estar localizada previamente, no requiere una localización exacta pero sí bastante aproximada. Además dentro del rostro localizado se debe contar con un conjunto de candidatos potenciales a áreas de los ojos. Esto significa que esta técnica en sí es empleada para comprobar si un área es o no el área de los ojos, más que para detectar el área de los ojos sin ninguna información precedente.

Se basa en la idea que el cambio de intensidad en la región de los ojos es más obvio que en otras regiones del rostro humano, por esta razón emplea la varianza como medida para establecer la variación de la intensidad de gris. Se crea un filtro de varianza considerando las características del ojo, dicho filtro tiene como particularidad ser invariante a la rotación y reflejar la variación de intensidad en lugar de la forma exacta de un objeto.

Para esta etapa experimental se creó un filtro de varianza de acuerdo a lo establecido en [26]. Se seleccionaron 26 imágenes de ojos en escala de grises de tamaño 28 x 28 con diferentes orientaciones y de diferentes personas. En la Figura 11.6 se muestran las imágenes empleadas como conjunto de entrenamiento para crear el filtro de varianza.

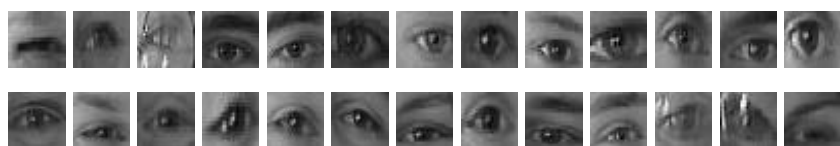


Figura 11.6. Imágenes de ojos utilizadas para crear el filtro de varianza

Cada una de estas imágenes es transformada a una imagen de varianza (de tamaño 7 x 7) y el filtro se forma del promedio de dichas imágenes. La Figura 11.7 muestra el filtro de varianza creado a partir de las imágenes de entrenamiento.

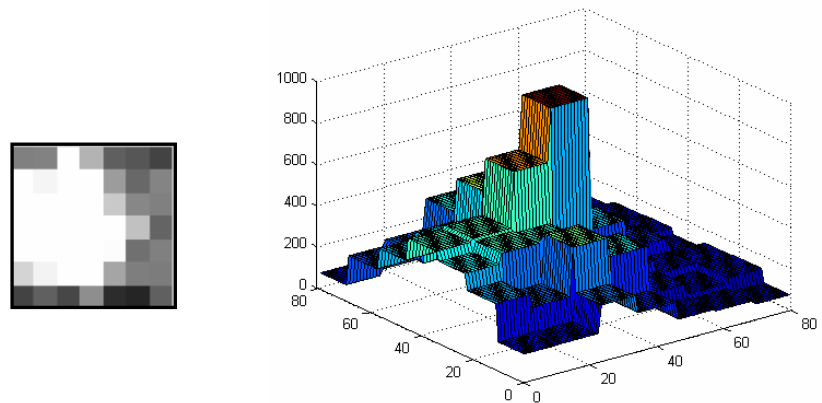


Figura 11.7. Filtro de varianza del ojo (reescalado a 80 x 80) y su gráfico en 3D

Una vez creado el filtro se selecciona un conjunto de imágenes formado por imágenes de ojos e imágenes de diferentes objetos, la finalidad es calcular la correlación entre el filtro creado y la imagen de varianza de cada imagen del nuevo conjunto para así establecer un valor umbral que permita discriminar las imágenes de ojos de aquellas que no lo son. La Figura 11.8 muestra el resultado de la correlación donde se fijó el valor umbral.

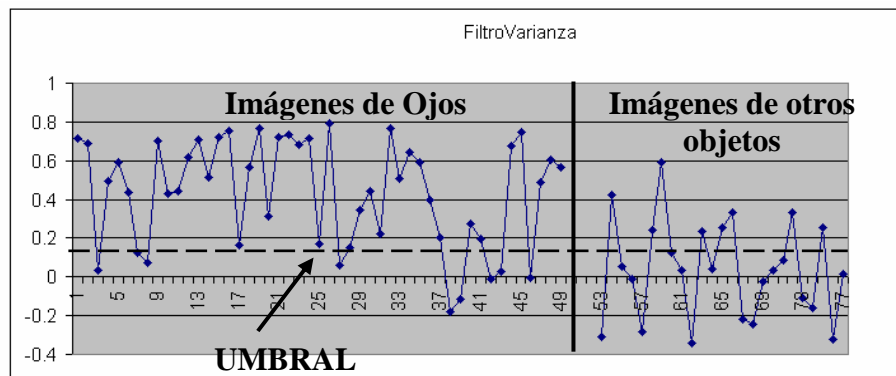


Figura 11.8. Resultado de la correlación entre el filtro de varianza y las imágenes de prueba.

Para las imágenes de ojos que se encuentran bajo el valor umbral se observó que correspondían a imágenes donde el ojo no está claramente definido (por ejemplo son imágenes de ojos parciales o con la presencia de lentes), esto significa que el filtro requiere que el área de los ojos se observe fácilmente y sin oclusiones. En tanto que el resultado de la correlación de algunas imágenes de objetos diferentes al ojo presentan valores por encima del umbral, lo que significa que la distribución de varianza del ojo no es única.

Esta última característica muestra que esta técnica no es confiable si se la utiliza como única forma para determinar la ubicación del área de los ojos y por ende del rostro, por lo que puede ser utilizada solamente como una técnica auxiliar.

Una vez establecido el valor umbral se aplica el filtro sobre la imagen de varianza de una imagen de un rostro para determinar el área de los ojos. La figura 11.9 muestra un resultado obtenido en esta etapa experimental.

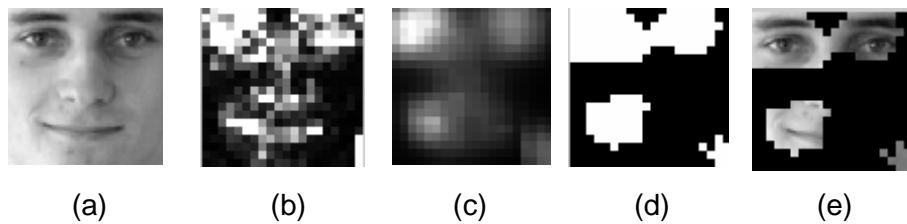


Figura 11.9. Resultado de la aplicación del filtro de varianza en la imagen de un rostro

(a) Imagen original (80x80) (b) Imagen de varianza del rostro (reescalada a 80x80)
(c) Respuesta del filtro de varianza del ojo (reescalada a 80x80) (d) Respuesta binarizada (e) Enmascaramiento de la imagen del rostro con la respuesta binarizada

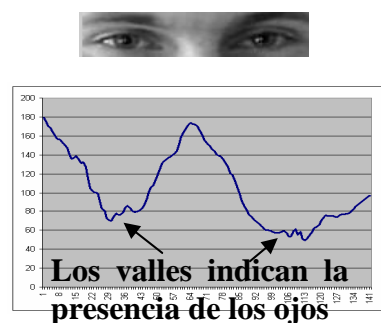
Como se puede observar en la Figura 11.9e si bien el filtro detecta el área de los ojos también presenta otras áreas como posibles áreas de los ojos. Asimismo se debe mencionar que esta técnica consume cierto tiempo de procesamiento para

calcular la imagen de varianza de un rostro lo que aumenta el tiempo total requerido para la localización de un rostro.

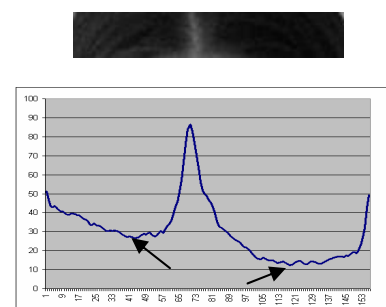
2. Funciones de proyección para detección de ojos

Las funciones de proyección se utilizan para detectar la posición de los ojos en una imagen que contiene sólo el área del rostro, es decir al igual que la técnica analizada antes ésta técnica también requiere que el área del rostro se haya localizado de forma aproximada. En la literatura se utilizan principalmente dos tipos de funciones de proyección: la función de proyección integral que se refiere al promedio de cada fila y de cada columna de una imagen (promedio de la intensidad), y la función de proyección de varianza que calcula la varianza de filas y de columnas en la imagen (variación de la intensidad). En [94] se utiliza una función de proyección adicional, la función de proyección generalizada que es una combinación lineal de las dos proyecciones anteriores.

Durante los experimentos realizados con esta técnica se observó que a pesar de que el área localizada del rostro sea bastante confiable (que no contenga otros objetos), la presencia de elementos propios del rostro como por ejemplo cabello, cejas, etc., pueden provocar detecciones falsas, (véase Figura 11.10), mientras que elementos como lentes pueden dificultar la detección, (véase Figura 11.11).



(a)



(b)

Figura 11.10. Función de Proyección integral horizontal

(a) del área de los ojos (b) de un área del cabello

Por esa razón se considera que esta técnica no es útil para la solución computacional propuesta.

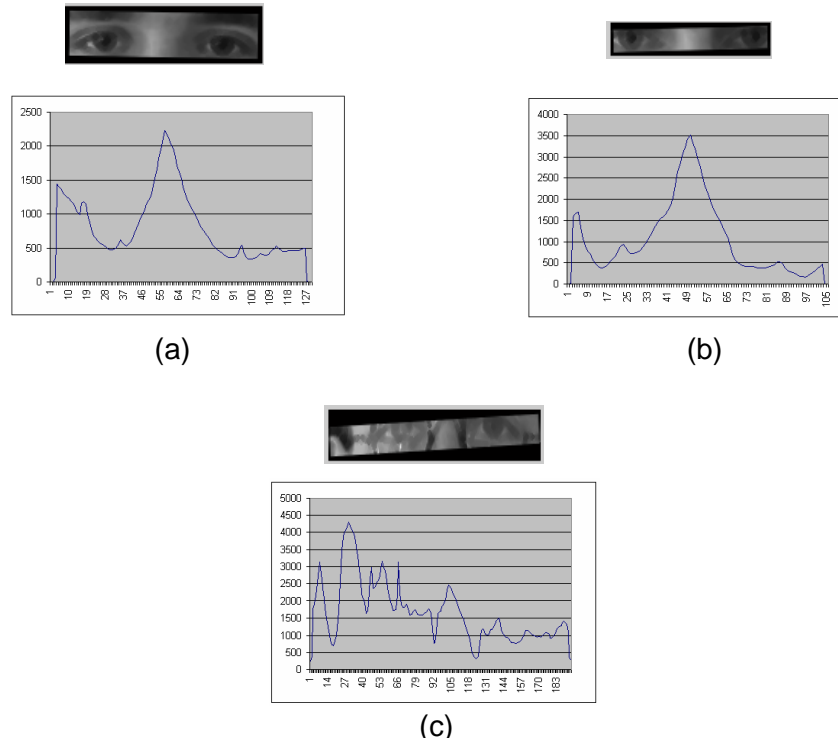


Figura 11.11. Función de Proyección generalizada horizontal

(a) del área de los ojos incluyendo las cejas (b) sólo del área de los ojos

(c) solo del área de los ojos con presencia de lentes

3. Detección de ojos utilizando wavelets

Otra de las técnicas analizadas en esta etapa fue el uso de wavelets para la detección de ojos, propuesta en [27] como parte de un algoritmo para detección de rostros.

De acuerdo a esta técnica se debe realizar primero la descomposición de la imagen con la que se va a trabajar en un árbol de wavelets de tres niveles⁴⁸. Luego de la descomposición se utiliza la banda AHH para la detección de

⁴⁸ Para mayor detalle sobre la descomposición utilizando wavelets, véase Capítulo 1, sección 1.1.8.2.4

ojos. En esta banda se realiza el análisis para detectar ojos empleando una ventana desplazable de tamaño 40x40. Cada área que es abarcada por esta ventana se analiza por separado, se extrae una señal unidimensional de dicha área aplicando un filtro paso bajo horizontal.

Si la señal que se obtiene corresponde a la imagen de un ojo, entonces debe tener tres máximos (picos) y dos mínimos (valles). Para esto se debe contar con un patrón de la señal de la imagen de un ojo, calculada a partir de un conjunto de entrenamiento de imágenes de ojos. Se compara la señal obtenida con el patrón utilizando la distancia de Mahalanobis y un valor umbral que permita decidir si el área contenida por la ventana es o no un ojo.

Al analizar esta técnica se observó que presenta el mismo problema descrito en la técnica anterior, es decir que para ciertas áreas que no corresponden a un ojo presenta una señal unidimensional similar a la del patrón⁴⁹, provocando falsas detecciones y dificultando la localización del rostro en la imagen, por lo tanto esta técnica debería ser utilizada en conjunto con otras técnicas (tal como el algoritmo que la emplea propone) para lograr una mayor robustez en la detección. Adicionalmente restringe el tamaño que pueden tener los rostros a localizar ya que al utilizar una ventana de tamaño fijo limita que la distancia ocular sea máximo de 60 píxeles que corresponde al tamaño máximo de rostros que puede localizar, mientras que la distancia mínima es 20 píxeles. Por lo que si se desea localizar un rostro cuya distancia ocular sea mayor de 60 píxeles esta técnica no es útil.

⁴⁹ Se debe mencionar que en la literatura no se encontró referencias sobre los filtros empleados para la descomposición wavelet, por lo que se utilizó una aproximación de los filtros. Echo que pudo influir en los resultados obtenidos.

Otro factor a considerar es el tiempo que requiere para analizar cada área de la imagen AHH con la ventana desplazable.

4. Localización de características faciales [36]

El método propuesto en [36] utiliza la información del color en el espacio YCbCr para localizar las características faciales de un rostro. Al igual que otros métodos analizados requiere que el área del rostro se haya localizado de forma aproximada antes de emplearlo.

La localización de características se divide en dos partes, la detección de ojos y la detección del área de la boca. Para la detección de ojos se crean dos mapas, uno para los componentes de crominancia (Cb y Cr) y otro para el componente de luminancia (Y), luego los mapas se combinan en uno solo de manera que el área de los ojos se muestre claramente mientras que las otras áreas faciales se suprimen. Esto se basa en dos características encontradas generalmente en las áreas de los ojos: la presencia de valores altos en el canal Cb y valores bajos en el canal Cr además de la presencia tanto de píxeles claros como de oscuros en el canal Y (luma).

Por su parte la detección del área de la boca se fundamenta en la siguiente característica: esta área contiene un mayor porcentaje de componente rojo que de componente azul, que otras regiones faciales. Por lo tanto se busca áreas donde el componente de crominancia Cr sea mayor que Cb.

Finalmente se combinan los resultados obtenidos para realizar una validación de las características geométricas faciales⁵⁰, es decir se analiza cada triángulo ojos-boca formado por un par de

⁵⁰ Para mayor detalle sobre la construcción de los mapas empleados así como la validación geométrica aplicada en este método, véase [36]

candidatos a ojos (obtenidos en la detección de ojos) y un candidato a área de la boca (obtenido en la detección del área de la boca) para verificar si cumple ciertas restricciones geométricas y de orientación. El triángulo que alcance el mayor puntaje en este análisis se considera que representa al área de los ojos y de la boca del rostro localizado.

Las principales ventajas de este método son su facilidad para detectar características faciales en múltiples escalas y que requiere poco tiempo de procesamiento ya que utiliza principalmente información del color.

Durante las pruebas realizadas en esta etapa se observó que para la detección de ojos el mapa de las componentes de crominancia es más estable que el mapa de luminancia. La figura 11.12 muestra los mapas para la crominancia y la luminancia de una imagen, la combinación de éstos en un solo mapa y su umbralización y binarización. Se puede notar que la imagen binaria que corresponde al mapa de crominancia muestra mejor el área de los ojos que la imagen binaria que corresponde a la combinación de los mapas de crominancia y luminancia.

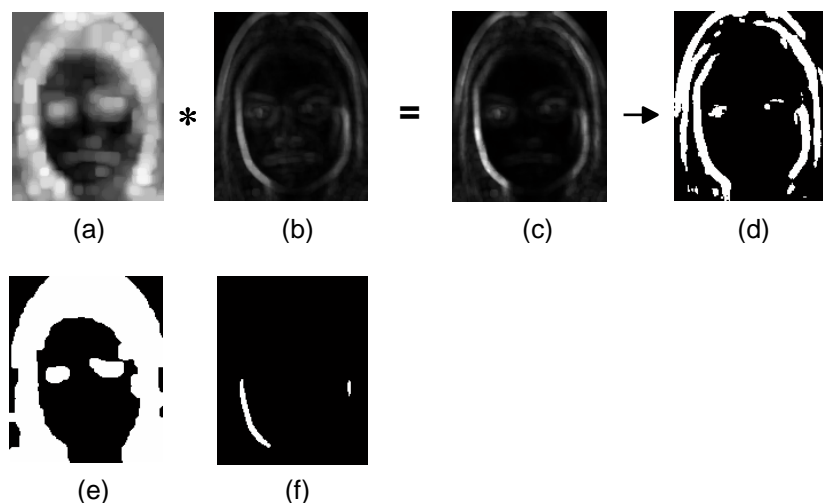


Figura 11.12. Mapas creados para la detección de ojos

(a) Mapa para la crominancia (b) Mapa para la luminancia (c) Combinación de los mapas (a) y (b). (d) Imagen binaria del mapa en (c). (e) Imagen binaria del mapa de crominancia (f) Imagen binaria del mapa de luminancia

En cuanto a la detección del área de la boca, el mapa creado inicialmente no permite una localización correcta, (véase Figura 11.13), por lo que se requiere mayor refinamiento del algoritmo implementado durante esta etapa a fin de obtener una mejor localización. Mientras que para la validación geométrica se debe establecer experimentalmente el porcentaje que cada puntuación parcial debe aportar para el puntaje total de un triángulo ojos-boca para obtener resultados más confiables.

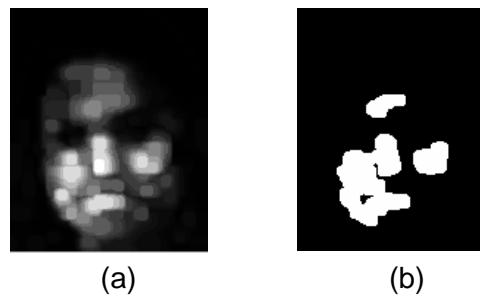


Figura 11.13. Mapa empleado para la detección del área de la boca
(a) Mapa para la crominancia (b) Imagen umbralizada y binarizada del mapa de crominancia.

5. Detección de ojos utilizando información del color [71]

La técnica propuesta en [71] requiere trabajar sólo en el área aproximada del rostro, realiza la detección de ojos en tres fases. En la primera las posibles áreas de los ojos son localizadas mediante umbralización en los espacios de color HSV y RGB normalizado secuencialmente, en la segunda fase se efectúa un análisis de componentes conectados para eliminar regiones que no pueden corresponder a áreas de los ojos y finalmente en la tercera etapa se emplean funciones de proyección en cada pareja de candidatos a ojos para validar que pareja corresponde realmente al área de los ojos.

Para esta etapa experimental se analizó sólo la primera fase de la técnica propuesta.

Las posibles regiones de los ojos se obtienen identificando las áreas de baja intensidad y eliminando las áreas de color de la piel.

Para identificar las áreas de baja intensidad la imagen debe estar normalizada en cuanto a iluminación luego de lo cual se realiza la detección de estas áreas mediante la umbralización del canal V de la imagen en el espacio de color HSV, adicionalmente se realiza la umbralización de la imagen en el espacio de color RGB normalizado para eliminar áreas de piel de tonalidad oscura que pueden presentar baja intensidad en el espacio HSV ⁵¹ y ser confundidas como posibles áreas de los ojos.

Una vez identificadas las regiones de baja intensidad que son potenciales candidatos a áreas de los ojos se elimina cualquier área que tenga color de la piel (cualquier tonalidad de piel) y de esta manera se obtienen los candidatos a áreas de los ojos.

En las pruebas efectuadas esta técnica mostró ser bastante confiable para detectar candidatos a áreas de los ojos en imágenes que no presentan degradación del color y con una iluminación cercana a la normal (características semejantes a las requeridas para este proyecto), además esta técnica requiere poco tiempo de procesamiento. Sin embargo no puede ser utilizada como única forma de detección puesto que presenta un conjunto de candidatos y no las áreas definitivas, adicionalmente las imágenes resultantes necesitan ser

⁵¹ De acuerdo a [71] las tonalidades oscuras de piel presentan baja intensidad en el canal V del espacio HSV al igual que las áreas de los ojos, por lo tanto se debe eliminar las áreas que puedan corresponder a estas tonalidades de piel para así evitar falsas detecciones.

sometidas a operaciones morfológicas para mejorar la conectividad entre píxeles cercanos y limpiarlas de ruido.

La Figura 11.14 muestra el proceso efectuado para la localización de las posibles áreas de los ojos en imágenes de personas con diferentes tonalidades de piel.

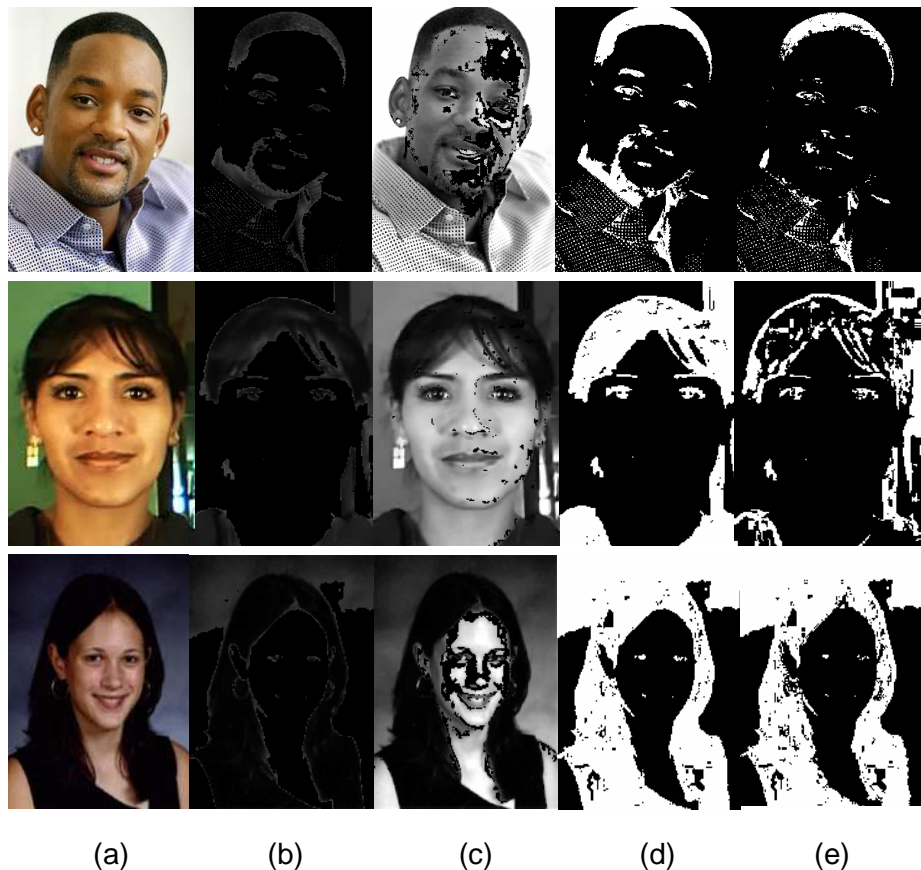


Figura 11.14. Localización de posibles áreas de los ojos en imágenes de personas con diferentes tonalidades de piel

(a) Imagen original (b) Imagen que muestra las regiones de baja intensidad (c) Imagen en la que se ha eliminado áreas relacionadas a tonalidades oscuras de piel (d) Posibles candidatos a áreas de los ojos ((b) AND (c)) (e) Candidatos a áreas de los ojos después de eliminar posibles áreas de piel de cualquier tonalidad.

- o Detección basada en la imagen

Este tipo de métodos o técnicas emplean la información contenida en toda la imagen, es decir la tratan como un todo a partir del cual se puede extraer características que permitan diferenciarla.

Al igual que gran parte de los métodos mencionados requieren que el área del rostro sea previamente localizada o en su defecto se debe dividir a la imagen que contiene al rostro en varios segmentos y analizar cada uno de éstos por separado. Por lo tanto para que estos métodos sean efectivos necesitan que la imagen de la que extraen las características para analizarlas contenga sólo el área del rostro.

Dentro de este grupo de técnicas o métodos se consideró como parte de esta etapa experimental el Análisis de Componentes Principales o PCA⁵² y las redes neuronales.

1. PCA

Para aplicar PCA en la detección de rostros se debe contar con un conjunto de imágenes de rostros (conjunto de entrenamiento) a partir del cual se forma el espacio de rostros o eigenspacio donde se proyectarán las nuevas imágenes para determinar si forman parte o no de este espacio.

La forma más sencilla para determinar si la imagen forma parte del espacio de rostros es calcular la distancia euclidiana entre la imagen centrada del posible rostro y su proyección en el espacio de rostros, si esta distancia es menor a un valor umbral se considera que la imagen contiene un rostro.

Para esta etapa experimental se construyó un espacio de rostros a partir de un conjunto de doce imágenes de rostros de personas diferentes. Algunas de estas imágenes fueron recolectadas de una base de rostros existente, en tanto que otras imágenes fueron encontradas en Internet.

⁵² Para mayor detalle sobre PCA, véase Capítulo 1, sección 1.1.8.4. También se puede consultar el Anexo 5.

Para reducir tiempo de procesamiento se normalizó el tamaño de las imágenes a 20 x 13. La Figura 11.15 muestra el conjunto de imágenes empleadas para la construcción del espacio de rostros, así como el rostro promedio obtenido a partir de este conjunto.

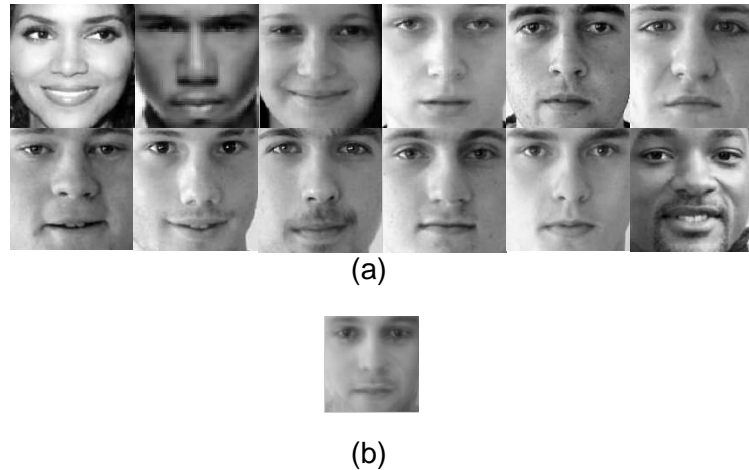


Figura 11.15. (a) Imágenes utilizadas para crear el espacio de rostros para PCA (b) rostro promedio obtenido.

Una vez construido el espacio de rostros y calculado el rostro promedio se realizó varias pruebas con imágenes que contenían rostros e imágenes que contenían otros objetos, con el fin de determinar si este método es confiable para la localización de rostros. A continuación se muestran algunos ejemplos de estas pruebas, (véase Figura 11.16).

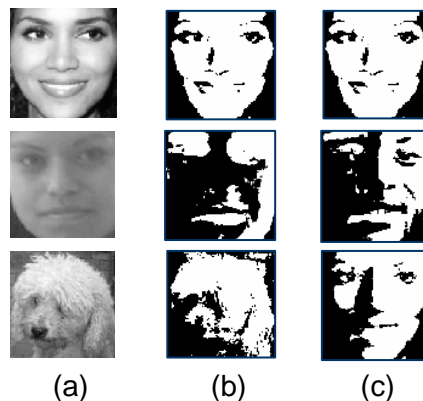


Figura 11.16. PCA para analizar imágenes (detección rostros)

(a) Imagen original (b) Imagen centrada con respecto al rostro promedio
(c) Proyección de la imagen centrada en el espacio de rostros.

Como se puede ver en la Figura 11.16 las imágenes que representan rostros humanos presentan una imagen centrada y una proyección más similares entre sí (menor distancia euclidiana) que las imágenes que no contienen rostros humanos. La primera imagen presentada en este ejemplo forma parte de las imágenes empleadas para crear el espacio de rostros por tal razón su imagen centrada y su proyección son semejantes entre sí y con la imagen original. Por otra parte la segunda imagen tiene una proyección e imagen centrada algo similares entre sí pero diferentes con la imagen original (esto se debe a que este rostro no forma parte del conjunto de entrenamiento), es decir la imagen corresponde a un rostro humano pero no es uno de los rostros empleados para crear el espacio de rostros. Finalmente en la tercera imagen la proyección y la imagen centrada son completamente diferentes lo que significa que la imagen no contiene un rostro humano.

Los resultados de las pruebas realizadas permiten concluir que este método es bastante confiable siempre y cuando la imagen contenga sólo el área del rostro. Adicionalmente el análisis de resultados efectuado muestra que se puede emplear PCA para verificar identidades.

2. Redes Neuronales

Su uso para la detección de rostros ha sido ampliamente analizado en la literatura [23, 37, 64] mostrando buenos resultados, sin embargo requiere contar con una gran cantidad de ejemplos sobre lo que es y lo que no es un rostro humano (ejemplos positivos y negativos) para su entrenamiento, puesto que el resultado final de la detección dependerá de la calidad y cantidad de los ejemplos proporcionados. Por esta razón se considera que este método de detección no es el más adecuado para el presente trabajo.

11.2.2 Extracción de características relevantes de un rostro

La extracción de características relevantes tiene como objetivo lograr diferenciar mediante estas características a un rostro de otro sin requerir toda la información contenida en la imagen de cada rostro.

Algunos de los métodos empleados con este fin y que se analizaron dentro de esta etapa experimental son: espectro de Fourier de la imagen de un rostro, filtro de correlación de promedio mínimo de energía [66], análisis de wavelets y PCA.

Los dos primeros métodos se analizaron en función de las referencias existentes en la literatura por lo que no se cuenta con resultados experimentales que permitan establecer un criterio sobre la robustez o confiabilidad de estos métodos.

A continuación se presenta un breve análisis de los métodos restantes.

- Análisis de wavelets

En la literatura existen diferentes métodos que emplean el análisis de wavelets para la extracción de características y el posterior reconocimiento de un rostro contenido en una imagen.

En esta etapa experimental se analizaron dos técnicas descritas en la literatura:

1. Análisis de wavelets en paquete para reconocimiento de rostros [29]
El algoritmo propuesto en [29] utiliza la descomposición de wavelets en paquete⁵³ para extraer las características más relevantes de un rostro y realizar su reconocimiento. Los filtros empleados para la descomposición wavelet son definidos experimentalmente por los autores pero no están descritos en la literatura.

⁵³ Para mayor detalle sobre la descomposición de wavelets, véase Capítulo 1, sección 1.1.8.2.4

Para llevar a cabo el análisis de este algoritmo se creó una aproximación de los filtros basándose en la información provista por la literatura.

Adicionalmente en [29] se propone analizar diferentes áreas de la imagen de un rostro: la parte superior, la parte inferior y el área de borde del rostro. Debido a las características de las imágenes empleadas para las pruebas no se analizó el área de borde de los rostros (las imágenes utilizadas no presenta el borde del rostro).

En las pruebas realizadas cada imagen fue sometida a una descomposición de wavelets en paquete de dos niveles empleando los filtros creados. A partir de esto se obtuvo un conjunto de 16 imágenes en el segundo nivel de descomposición (una imagen aproximación y 15 imágenes detalle), estas imágenes se utilizaron para crear el vector de características del rostro, el cual está formado por la varianza y el promedio de cada área (superior e inferior) de la imagen aproximación de segundo nivel y por la varianza de cada una de las imágenes detalle, dando un total de 19 características por rostro.

Para comprobar si las características extraídas realmente son relevantes se utilizó la distancia de Bhattacharyya. A una mayor distancia entre dos vectores de características se considera que pertenecen a rostros de personas diferentes, mientras que si la distancia es pequeña se considera que pertenecen a la misma persona.

La Figura 11.17 muestra algunas imágenes que forman parte del conjunto que se utilizó para verificar la confiabilidad de las características extraídas de un rostro mediante wavelets. Estas imágenes corresponden a dos personas diferentes con diferentes expresiones faciales.

Para cada imagen se extrajo el vector de características correspondiente y luego se calculó la distancia entre este vector y los vectores de las otras imágenes.

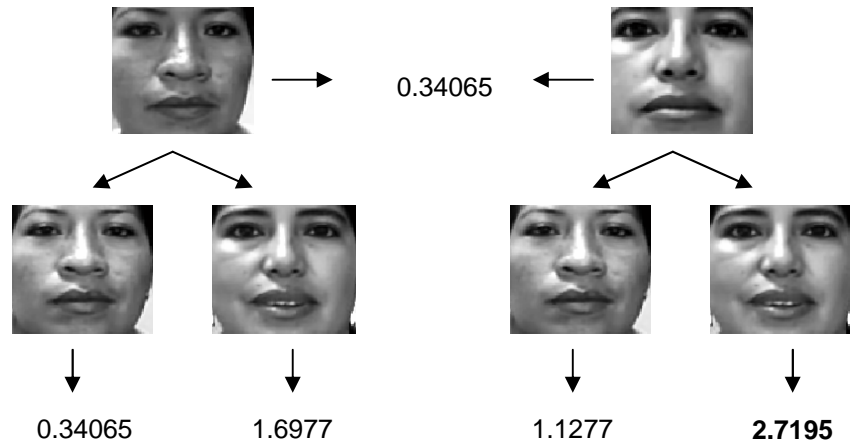


Figura 11.17. Extracción de características mediante descomposición de wavelets en paquete

Se puede observar que las imágenes que corresponden a la segunda persona tienen la mayor distancia entre sus vectores de características lo que podría interpretarse como vectores de personas diferentes aunque no lo son. Este problema puede ser provocado por la diferencia en el acercamiento de las dos imágenes.

Sin embargo en los resultados generales de las pruebas realizadas se observó que los vectores de características obtenidos con esta técnica no permiten una buena discriminación entre una persona y otra.

Se debe recordar que el algoritmo implementado se basa en [29] pero es en sí una adaptación debido a las restricciones antes mencionadas (filtros utilizados y el área del borde del rostro), por lo que los resultados de esta prueba no deben ser considerados como referencia del método descrito en la literatura.

2. Análisis de wavelets en paquete (Haar)

El algoritmo propuesto en [32] es empleado para la detección de rostros por lo que se requirió hacer una adaptación para emplearlo en la extracción de características para el reconocimiento de rostros.

En el algoritmo original se sugiere realizar una descomposición de wavelets en paquete (tres niveles) de la imagen que contiene un rostro, para esto se utiliza la wavelet Haar. Luego de la descomposición sólo se utilizan las bandas HL y LH de cada nivel para construir el vector de características.

Este vector está compuesto por el promedio y la desviación estándar de las bandas HL y LH en conjunto (por cada división de la descomposición) a fin de lograr características invariantes a la rotación, enseguida se aplica la transformada discreta de Fourier al conjunto de características para que sean también invariantes a la escala.

Para realizar la detección de rostros en el algoritmo original [32] se cuenta con un vector patrón creado a partir de un conjunto de imágenes de rostros (conjunto de entrenamiento). Cada nuevo vector calculado se compara con el vector patrón para establecer si es o no un rostro humano (utilizando la distancia de Bhattacharrya).

En las pruebas efectuadas durante esta etapa experimental se utilizó el mismo principio del algoritmo, pero en lugar de crear un patrón que represente las características de los rostros humanos en general, se creó un patrón para cada persona cuyo rostro formaba parte del conjunto de imágenes que se empleó.

Los resultados fueron aceptables, de 20 imágenes empleadas para la prueba el 60% de los vectores extraídos (uno por cada imagen) mostraron contener características discriminatorias que permitieron

diferenciar a una persona de otra. No obstante los valores umbrales para discriminar entre diferentes personas son difíciles de establecer puesto que al aplicar la distancia de Bhattacharyya entre vectores de características distintos se obtienen valores muy pequeños con diferencias de centésimas o milésimas, lo que hace que esta tarea sea bastante compleja y larga (para cada persona se debe establecer un valor umbral), factor que debe ser considerado durante la selección del método más adecuado.

- o Eigenfaces (PCA)

Durante las pruebas efectuadas con los métodos para localización de rostros el Análisis de Componentes Principales o PCA mostró buena capacidad de discriminación para imágenes de rostros de diferentes personas. Por esta razón se consideró analizar el PCA como método para extracción de características⁵⁴.

En esta etapa experimental se realizaron tres pruebas utilizando un conjunto de 10 personas, las imágenes sólo contenían el área del rostro.

Para la primera prueba se utilizó imágenes de rostros localizados de forma aproximada, es decir en algunas imágenes los rostros no aparecen completamente centrados, algunas contienen sólo parte del rostro mientras que otras contienen además del área del rostro una pequeña parte del área del cuello. El espacio de rostros se creó con una imagen de cada persona (10 imágenes en total) y las pruebas se efectuaron con dos imágenes por persona (diferentes a las utilizadas para crear el espacio). Una vez creado el espacio de rostros se seleccionó los eigenvectores que representaran el 95% de la varianza, obteniendo un conjunto de siete eigenvectores.

⁵⁴ Para mayor detalle sobre el uso de PCA para el reconocimiento de patrones, véase Capítulo 1, sección 1.1.8.4.1

Cada imagen utilizada en la construcción del espacio fue proyectada sobre éste para obtener el vector de características asociado a cada persona (cada vector contiene sólo siete características), posteriormente las imágenes de prueba fueron proyectadas sobre el mismo espacio de rostros para obtener sus correspondientes vectores. Finalmente se empleó la distancia euclidiana entre vectores para saber cuáles se encontraban más cercanos entre sí lo que implicaría que corresponden a la misma persona. La prueba logró un 45% de aciertos contra un 55% de fallos⁵⁵, sin embargo se consideró que el factor que más influyó en estos resultados es la localización de los rostros, que en este caso fue sólo aproximada, (véase Figura 11.18).

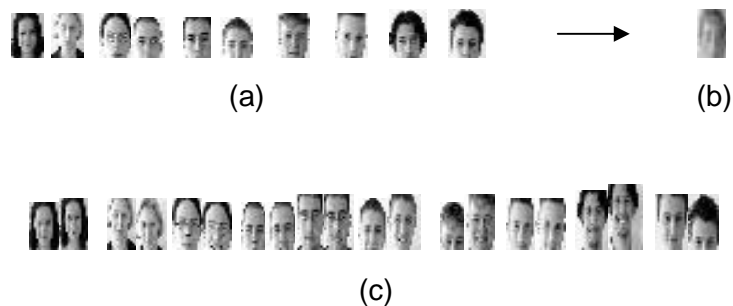


Figura 11.18. Extracción de características mediante PCA

(a) Conjunto de imágenes para crear el espacio de rostros (tamaño imágenes 20 x 20)
 (b) Rostro Promedio (c) Conjunto de imágenes de prueba (tamaño imágenes 20 x 20).

En la segunda prueba se empleó el mismo conjunto de imágenes pero con el área del rostro mejor localizada, evitando rostros parciales o la presencia del área del cuello en las imágenes. El resultado de esta prueba fue 65% de aciertos, es decir que al emplear imágenes con el área del rostro mejor localizada los vectores de características extraídos mediante PCA permiten una mejor discriminación entre diferentes personas. En esta prueba cada vector también contenía sólo siete características, por lo que no se requirió mucho tiempo para la extracción ni para el reconocimiento.

⁵⁵ Al emplear todos los eigenvectores del espacio de rostros inicial (10 eigenvectores) se obtuvo 40% de aciertos, lo que indica que los últimos eigenvectores no son discriminantes sino que pueden contener información relativa a la presencia de ruido en las imágenes, cambios de iluminación u otros elementos poco relevantes y que pueden provocar errores en el reconocimiento.

En la última prueba realizada durante esta etapa se consideró emplear imágenes sintéticas, que se obtienen a partir de la imagen de un rostro pero con cierto grado de traslación de éste o con mayor o menor acercamiento del rostro en la imagen. El uso de estas imágenes adicionales se fundamenta en el hecho que la localización de rostros siempre tiene asociado un error de localización sea cual sea el método empleado, esto se debe a que los métodos que existen no pueden encontrar cada rostro en cada imagen con precisión de píxeles.

De acuerdo a [51] el error de localización puede ocasionar que un vector de características de una imagen de prueba esté más cerca al vector de características de una imagen de entrenamiento de una persona diferente que al vector de características de una imagen de entrenamiento de la persona a la que realmente pertenece el vector de la imagen de prueba. Esto se puede comprobar con los resultados que se obtuvieron en la primera prueba efectuada, donde el error de localización era alto lo que ocasionó un mayor porcentaje de errores (la imagen de una persona era asociada incorrectamente con la imagen de otra persona diferente mediante sus vectores de características).

Se crearon seis imágenes sintéticas a partir de la imagen de entrenamiento: traslación superior, extracción de un área menor a partir de la traslación superior, traslación inferior, traslación inferior abarcando una mayor área, leve acercamiento del rostro, leve alejamiento del rostro, (véase Figura 11.19). Se debe notar que las imágenes sintéticas no deben tener cambios bruscos con respecto a la original porque esto podría empeorar el reconocimiento en vez de mejorarlo.



Figura 11.19. Ejemplo de imágenes sintéticas empleadas en PCA

(a) Imagen Original (b) Imágenes sintéticas

En [51] se sugiere emplear las imágenes sintéticas para realizar la modelación de la distribución gaussiana que aproxime al clúster que represente a cada persona, sin embargo el número de imágenes por persona es demasiado pequeño para obtener una mezcla de gaussianas (el número de ejemplos es crítico para la confiabilidad de la mezcla). Por esta razón en la tercera prueba se utilizó las imágenes originales para crear el espacio de rostros y se proyectó estas imágenes y sus imágenes sintéticas sobre dicho espacio obteniendo un conjunto de siete vectores de características por cada persona, luego se calculó la distancia euclidiana de un vector de características nuevo contra todos los conjuntos de vectores y se seleccionó la distancia mínima. El resultado de esta prueba fue 82% de aciertos contra un 18% de error.

El resultado final muestra que el método de PCA para la extracción de características obtiene buenos resultados y requiere poco tiempo de procesamiento, además de lograr una reducción de dimensionalidad óptima. Sin embargo se debe considerar emplear un método de localización de rostros que tenga un error de localización mínimo así como también el uso de imágenes sintéticas para obtener mejores resultados.

11.2.3 Entrenamiento de una red neuronal para la tarea de verificación de rostros

Una de las principales restricciones de la solución computacional propuesta es el número de imágenes con el que se cuenta por cada persona, este problema es muy común en la mayoría de trabajos descritos en la literatura. Por esta razón se considera que el utilizar un método de clasificación estadístico tradicional para realizar la verificación de rostros no es adecuado con un número tan reducido de ejemplos. Métodos como las redes neuronales o métodos de tipo no-estadístico por ejemplo KNN, SVM, entre otros son más adecuados para esta situación.

En el caso de las redes neuronales consiguen más rápidamente mejores modelos que los métodos estadísticos tradicionales. Si bien las redes neuronales alcanzan un mayor grado de aciertos en la verificación al contar con un mayor número de ejemplos para su entrenamiento pueden alcanzar buenos resultados con un menor número de ejemplos gracias a su capacidad de generalización.

En esta etapa experimental se consideraron dos tipos de redes neuronales, la red perceptrón multicapas y la red RBF⁵⁶, el análisis de éstas se realizó en función de las referencias existentes en la literatura.

De acuerdo a la información analizada la red perceptrón multicapas que es comúnmente entrenada utilizando la retropropagación de error, es capaz de realizar una buena generalización para problemas complejos pero el asegurar una convergencia global de todo el entrenamiento de la red es difícil debido al tipo de neuronas ocultas que la componen (no lineales) y a la naturaleza del mapeo entrada-salida que conducen a una gran cantidad de mínimos locales. Además el tiempo de entrenamiento de esta red puede ser muy grande.

Según algunos trabajos descritos en la literatura [35] la red perceptrón multicapas es efectiva para la tarea de reconocimiento cuando las imágenes no presentan variaciones, por ejemplo si no existe error de localización de rostros o no existe variación en la posición de los rostros, etc., pero no es tan efectiva como otras redes cuando el tipo de mapeo no es lineal como por ejemplo en el reconocimiento de rostros en presencia de rotación de la cabeza. Este hecho se demuestra en [48] donde se utiliza una red perceptrón multicapas para la verificación de rostros, sin embargo dicho trabajo parte del supuesto que el proceso de detección de rostros ha sido efectuado perfectamente para esto el área del rostro es extraída utilizando las coordenadas de los ojos localizadas manualmente (se asume una detección perfecta del rostro).

⁵⁶ Para mayor detalle sobre estos tipos de redes, véase Capítulo 1, sección 1.5

En cuanto a la red RBF presenta como principales ventajas su simplicidad computacional, realiza una robusta generalización, utiliza la aproximación para manejar la presencia de ruido en la información utilizada para el entrenamiento, presenta baja tasa de falsos positivos en el reconocimiento y de acuerdo a varios trabajos descritos en la literatura [35, 82] es una herramienta de clasificación adecuada para tareas en tiempo real.

Considerando todas las características antes mencionadas se optó por emplear redes RBF como parte de esta solución computacional, la arquitectura final de la red será discutida en la fase de evaluación del desempeño de la solución computacional.

11.3 ESTRATEGIA DE DISEÑO CONCEPTUAL DE LA SOLUCIÓN

En base al esquema planteado para el reconocimiento de rostros, (véase Figura 11.1) y al análisis realizado para los diferentes métodos considerados dentro de esta etapa experimental, se propone el siguiente diseño de la solución computacional descrito a un nivel algorítmico.

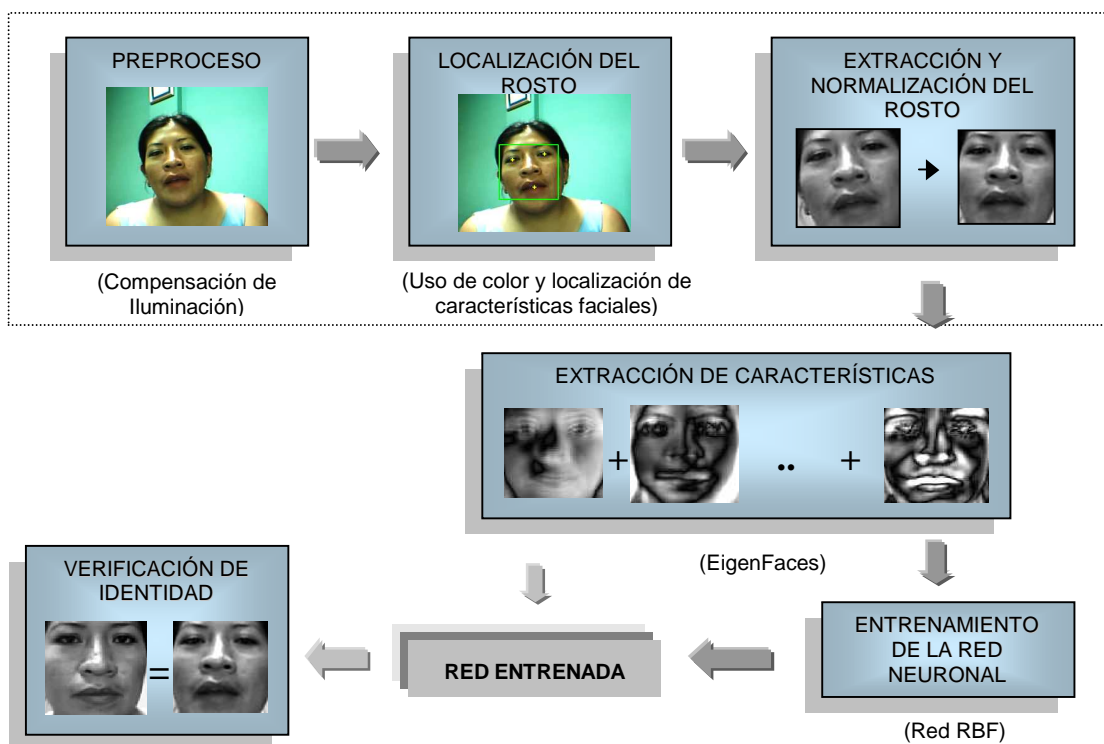


Figura 11.20. Estrategia de diseño de la solución computacional (algoritmo)

Como se puede observar en la Figura 11.20 el algoritmo propuesto se deriva de varias ideas de diversos trabajos descritos en la literatura, cada uno de los cuales se cita dentro del análisis realizado en la sección anterior.

A continuación se describe el algoritmo por fases presentando los motivos por los que se lo estructuró de esa forma:

1. Preprocesamiento.

Para el preproceso de la imagen como tal se deben ejecutar dos tareas:

- Eliminar el ruido que pueda existir en la imagen por efectos del dispositivo de adquisición aplicando un filtro de paso bajo.
- Corregir la iluminación de la imagen mediante la técnica de *compensación de iluminación* cuando el número de píxeles que están sobre el 95% del luma en su rango absoluto (referencia de blancos) están entre el 1% y el 30% del total de píxeles en la imagen, caso contrario mejorar el contraste de la imagen mediante la expansión de su histograma. La razón por la que no se realiza siempre la corrección de iluminación mediante la técnica de compensación es que si la referencia de blancos es muy pequeña se asume que la imagen no requiere corrección de iluminación, mientras que si la referencia es mayor al 30%, considerando las características que deben tener las imágenes empleadas, los píxeles del rostro podrían formar parte de dicha referencia por lo que al realizar la compensación se empeoraría la apariencia de la imagen. En este caso es mejor emplear la expansión del histograma para obtener un mejor contraste en la imagen.

Para la localización del rostro el algoritmo se basa en un conjunto de técnicas con diferente complejidad, ejecutadas secuencialmente de tal manera que cada técnica restringe progresivamente los posibles candidatos a rostros dentro de la imagen previamente preprocesada. El

objetivo de la secuencia es reducir el tiempo de procesamiento de las técnicas subsiguientes, además asumiendo que las técnicas son independientes en el sentido de tener diferentes características para la localización se reduce la tasa de localizaciones erróneas, el inconveniente de esta arquitectura está en el riesgo de reducir la tasa de localización por lo que las técnicas empleadas deberán ser más flexibles durante su ejecución al momento de eliminar áreas que no consideren candidatos a rostros.

Esta cadena de técnicas está compuesta por:

- *Localizar áreas del color de la piel empleando las reglas de decisión utilizadas para el espacio RGB normalizado.* Se emplea como primera técnica por ser rápida de aplicar además de mostrar ser bastante confiable en las pruebas efectuadas (considerando las características de la solución computacional), mostrando un buen balance entre la retención de ruido y la eliminación de áreas del color de la piel (no es una técnica rígida ni muy flexible). Una vez localizadas dichas áreas, se debe someter a la imagen a operaciones morfológicas básicas para remover elementos que por su tamaño no pueden ser rostros y también para mantener las posibles áreas del rostro concatenadas.
- Detectar el área aproximada del rostro. Después de segmentar la imagen empleando la información de color (localización de áreas del color de la piel) se la transforma a escala de grises y en esa nueva imagen se aplica la *transformada de Hough para la detección de elipses*. Debido a que esta técnica es computacionalmente costosa se la emplea bajo ciertas restricciones (véase Anexo 7), además se emplea la imagen en escala de grises reescalada a la cuarta parte.

La transformada de Hough trabaja sólo con píxeles de borde por lo que antes de utilizarla se debe realizar la detección de bordes. Para

esta tarea se seleccionó el filtro de Sobel por ser un filtro que logra un equilibrio entre tiempo de procesamiento y precisión; este filtro se aplica sobre la imagen en sentido horizontal, vertical y diagonal (45° y 135°). Una vez obtenidos los bordes se procede a efectuar el adelgazamiento de éstos para evitar la redundancia de píxeles de borde, se aplica sobre la imagen el filtro de Canny cuya característica es detectar bordes con grosor de un píxel y se eliminan los píxeles de borde obtenidos por el filtro de Sobel que no aparecen como píxeles de borde con el filtro de Canny.

Se calcula la transformada de Hough empleando los bordes detectados, el proceso de votación de la transformada emplea la propagación de un voto estimado [56] hacia las elipses cercanas en el espacio de los parámetros. Una vez terminado todo el proceso de votación se fusionan las elipses cercanas y de semi-ejes similares, finalmente se seleccionan las elipses con más alta probabilidad.

Para cada elipse resultante se realizan las tareas descritas a continuación:

1. Los parámetros de la elipse (coordenadas del centro y longitud de los semi-ejes) son reescalados con referencia al tamaño original de la imagen en escala de grises.
2. Se extrae de esta imagen el área que abarca la elipse y se la trata como una nueva imagen.
3. Para la nueva imagen se detecta el área de los ojos, el área de la boca, se realiza la validación geométrica del triángulo ojos-boca .
4. Si la puntuación obtenida por el triángulo es mayor a un valor umbral se normaliza la imagen en cuanto a orientación del rostro y luego se extrae el área del rostro como una nueva imagen.
5. Se ajusta la intensidad de la imagen del rostro.

6. Se proyecta esta imagen en el espacio de rostros (PCA) y se calcula la distancia entre la proyección y la imagen para confirmar si es un rostro.
 7. Si la distancia es menor a un valor umbral el proceso de localización del rostro finaliza, caso contrario se repiten los pasos 1-7 con la siguiente elipse hasta encontrar un rostro o utilizar todas las elipses en cuyo caso se concluye que la imagen no contiene un rostro humano.
- o Detectar el área de los ojos. Primero se localizan las posibles áreas de los ojos con ayuda de los espacios de color HSV y RG normalizado empleando la técnica *detección de ojos utilizando información del color*, luego se elabora un mapa de posibles áreas de los ojos basándose en la técnica localización de características faciales pero sólo se analizan los componentes de crominancia para la construcción del mapa.

Se obtienen los posibles candidatos a ojos combinando (AND) las dos imágenes resultantes en una sola imagen donde se realizan operaciones morfológicas básicas para conectar áreas cercanas.

Inmediatamente se realiza un análisis de componentes conectados en la imagen. Para cada componente se analiza su ubicación en la imagen, la relación entre el alto y ancho de éste, y la relación entre su área y el área de la imagen. Sólo los componentes que alcancen los valores umbrales preestablecidos para estas relaciones se mantienen como posibles candidatos a ojos, el resto de componentes se eliminan.

Los componentes que pasen el análisis anterior son sometidos a un análisis de características geométricas, donde se crean parejas de componentes para analizarlas. Sólo se conservan las parejas cuyos componentes se encuentran a una distancia adecuada como para

representar las áreas de los ojos y que el grado de inclinación no sea mayor a 10 grados (restricción de la solución computacional). Finalmente se obtiene un conjunto de parejas candidatas a áreas de los ojos.

- Detectar el área de la boca. Para obtener los candidatos para el área de la boca se emplea el mapa propuesto en la técnica *localización de características faciales*. El mapa original es sometido a operaciones morfológicas básicas, mejora del contraste y umbralización con el objetivo de obtener mejores resultados.

Una vez obtenido el mapa se realiza un análisis de componentes conectados. Para cada componente se analiza su ubicación en la imagen, la relación entre el alto y ancho de éste, y la relación entre su área y el área de la imagen. Sólo los componentes que alcancen los valores umbrales preestablecidos para estas relaciones se mantienen como posibles candidatos, el resto de componentes se eliminan.

- Validar los triángulos ojos-boca formados por los candidatos. Cada pareja de candidatos a ojos forma un triángulo con cada candidato para el área de la boca (los vértices del triángulo están formados por los centroides de los candidatos).

Primero se valida que el vértice asociado al candidato a área de la boca se encuentre ubicado bajo los otros dos vértices, no al mismo nivel ni sobre éstos. También se valida que los vértices formen un triángulo, que el triángulo sea agudo (pero que ninguno de sus ángulos sea menor a $\pi/7$) y que su área no sea demasiado grande o muy pequeña con relación al área de la imagen.

Sólo los triángulos que cumplen estas condiciones son evaluados para obtener una puntuación que determine que tan bien representan al área de los ojos y la boca.

Esta evaluación consiste de cuatro pruebas cada una con su propia puntuación. Las pruebas analizan la orientación y simetría del triángulo, la relación entre el área del triángulo y el área de la imagen, la relación entre las áreas de los candidatos a ojos y la relación de la distancia entre las áreas de los ojos y la distancia entre el punto medio de estas áreas y el área de la boca.

Finalmente se realiza una combinación lineal de las puntuaciones de las pruebas, este valor constituye el puntaje alcanzado por el triángulo.

Para la extracción y normalización del rostro se deben ejecutar las siguientes tareas:

- Extraer el área del posible rostro. Se selecciona el triángulo con el mayor puntaje siempre y cuando sobrepase un valor umbral. Las coordenadas de los vértices del triángulo son transformadas para obtener las coordenadas reales en la imagen original (todo el análisis de las características faciales se realiza en una imagen extraída de la imagen original), una vez obtenidas las coordenadas se rota la imagen original de acuerdo al ángulo de inclinación entre las áreas de los ojos (este ángulo se calculó como parte del análisis de características geométricas para la detección de ojos) con el objetivo de corregir cualquier inclinación del rostro.

En la imagen rotada se calculan las nuevas coordenadas de los centroides (vértices del triángulo) y se extrae sólo el área del rostro. Esta nueva imagen se encuentra normalizada en cuanto a orientación del rostro.

- Ajustar la intensidad de la imagen normalizada del rostro.
- Verificar si la imagen realmente representa a un rostro. Todo el proceso de localización antes descrito puede dar resultados incorrectos bajo ciertas circunstancias, por ejemplo si en la imagen existen objetos elípticos con un color similar al color de la piel, la localización podría dar como resultado uno de estos objetos. Por esta razón, se realiza una verificación final de la imagen del rostro empleando la técnica de detección de rostros con PCA para reducir la tasa de localizaciones erróneas (esto no significa que no pueden existir localizaciones falsas).

El resultado del preprocesamiento es una imagen que contiene sólo el área del rostro, es importante mencionar que la detección tanto de los ojos como de la boca es aproximada lo que significa que los centroides que forman los vértices del triángulo no coinciden exactamente con los centros de los ojos o de la boca, pero en el caso de los ojos generalmente es bastante cercana la aproximación.

2. Extracción de características relevantes de un rostro.

La imagen obtenida como resultado del preprocesamiento se encuentra normalizada en cuanto a orientación y a iluminación. Luego de esto se la normaliza en cuanto a tamaño (92x92) para extraer sus características más relevantes.

La técnica seleccionada para la extracción de características es *eigenfaces* que se basa en *PCA*. Cada imagen de un rostro tendrá asociado un vector de características, el tamaño de este vector dependerá del número de *eigen*vectores seleccionados para formar el espacio de rostros. Estos valores serán analizados durante la etapa de evaluación del desempeño de la solución computacional.

3. Entrenamiento de la red neuronal para la tarea de verificación de rostros

Una vez que se cuenta con el conjunto de vectores de características que representan al conjunto de imágenes de entrenamiento se crea una *red neuronal de tipo RBF* y se la entrena con dichos vectores. El objetivo de entrenar a la red con los vectores de características y no con las imágenes de los rostros es evitar un sobre ajustamiento de la red así como también reducir la carga computacional.

Para este tipo de red se debe definir previamente a su entrenamiento el ancho de la gaussiana más adecuado para lograr una buena generalización a partir del conjunto de vectores proporcionados. La selección de este valor será analizada durante la etapa de evaluación del desempeño de la solución computacional.

4. Verificación de identidad (fase de reconocimiento)

Para la verificación de una identidad debe existir un conjunto de imágenes de rostros que constituyan el espacio de rostros, además la red neuronal debe estar creada y entrenada para esta tarea.

El vector de características asociado a una nueva imagen es extraído y presentado como entrada a la red RBF, ésta analiza el vector y presenta sus resultados. Finalmente se selecciona la salida (resultado) con mayor valor y si supera un umbral se busca la identidad asociada a dicha salida, se compara esta identidad con la proporcionada junto con la imagen analizada y se valida si son iguales o no.

De esta manera concluye el proceso de verificación de una identidad.

Anexo 12

ESTUDIO DE CASOS DE USO FUNCIONALIDAD ADICIONAL

CASO DE USO :		COMPROBAR PERFIL PARA ACCESO (inclusión)
Actor:		Operador (O) o Administrador (A)
Propósito:	Controlar el acceso hacia las funciones de administración.	
Descripción: Comprueba que el usuario tenga el perfil adecuado para realizar operaciones de administración.		
Precondición:	Ninguna	
Secuencia Normal: (eventos)	Acción del Actor	Respuesta de la Solución Computacional
	1. Este caso de uso inicia cuando el Administrador solicita realizar operaciones de administración.	2. Comprueba que el usuario tenga el perfil de administración.
		3. Muestra las opciones de administración existentes.
		4. El caso de uso termina.
Poscondiciones:	Se ha permitido el acceso de un usuario válido a las funciones de administración existentes.	
Extensiones:	Paso	Acción
	3	a. Si el usuario no cuenta con el perfil adecuado. 1. S: Informa de esta situación al usuario. 2. S: El caso de uso termina.

Tabla 12.1. Descripción del caso de uso Comprobar Perfil para Acceso

CASO DE USO : GUARDAR IMAGEN INICIAL			
Actor: Operador (O)			
Propósito: Guardar una copia de la imagen original.			
Descripción: Permite guardar una copia de la imagen que se obtuvo ya sea para registrar un nuevo usuario o para verificar una identidad, independientemente de la fuente de la que provenga, incluso si se desea hacer una nueva copia de una imagen almacenada en disco.			
Precondición: Se ha obtenido una imagen desde alguna de las fuentes permitidas.			
Secuencia Normal: (eventos)	Acción del Actor		Respuesta de la Solución Computacional
	1. Este caso de uso inicia cuando el Operador solicita almacenar una copia de la imagen obtenida.		2. Solicita el nombre, formato y la ubicación donde se desea almacenar la copia de la imagen.
	3. Indica el nombre, formato y la ubicación para la copia de la imagen.		
	4. Solicita almacenar la copia de la imagen.		5. Valida los datos ingresados.
			6. Almacena una copia de la imagen con el nombre, formato y en la ubicación especificada.
		7. El caso de uso termina.	
Poscondiciones: Se ha almacenado una copia de la imagen obtenida (imagen inicial).			
Extensiones:	Paso		Acción
	3		a. Si el Operador solicita cancelar el proceso. 1. S: El caso de uso termina.
	5		a. Si los datos ingresados no son válidos. 1. S: Solicita al Operador que verifique los datos ingresados y le permite modificarlos. 2. S: El caso de uso continúa (paso 4).
	7		a. Si no se logró almacenar la copia. 1. S: Informa de esta situación al Operador. 2. S: El caso de uso termina.

Tabla 12.2. Descripción del caso de uso Guardar Imagen Inicial

CASO DE USO : GUARDAR IMAGEN ROSTRO LOCALIZADO		
Actor: Operador (O)		
Propósito:	Guardar la imagen que muestra el rostro localizado.	
Descripción: Permite guardar la imagen que resulta del proceso de localización del rostro, es decir con el área del rostro ubicada dentro de la imagen.		
Precondición:	Se ha localizado un rostro dentro de una imagen.	
Secuencia Normal: (eventos)	Acción del Actor	Respuesta de la Solución Computacional
	1. Este caso de uso inicia cuando el Operador solicita almacenar la imagen con el rostro localizado.	2. Solicita el nombre, formato y la ubicación donde se desea almacenar la imagen.
	3. Indica el nombre, formato y la ubicación para la imagen.	
	4. Solicita almacenar la imagen.	5. Valida los datos ingresados.
		6. Captura la imagen con el rostro localizado.
		7. Almacena la imagen con el nombre, formato y en la ubicación especificada.
		8. El caso de uso termina.
Poscondiciones:	Se ha almacenado la imagen que muestra el rostro localizado.	
Extensiones:	Paso	Acción
	3, 4	a. Si el Operador solicita cancelar el proceso. 1. S: El caso de uso termina.
	5	a. Si los datos ingresados no son válidos. 1. S: Solicita al Operador que verifique los datos ingresados y le permite modificarlos. 2. S: El caso de uso continúa (paso 4).
	8	a. Si no se logró almacenar la imagen. 1. S: Informa de esta situación al Operador. 2. S: El caso de uso termina.

Tabla 12.3. Descripción del caso de uso Guardar Imagen Rostro Localizado

CASO DE USO : ACTUALIZAR CLAVES DE USUARIO		
Actor: Administrador (A)		
Propósito:	Actualizar la clave de un usuario registrado.	
Descripción: Permite actualizar la clave de cualquier usuario registrado.		
Precondición:	El usuario debe estar autenticado con el perfil administrador, además deben existir usuarios registrados.	
Secuencia Normal: (eventos)	Acción del Actor	Respuesta de la Solución Computacional
	1. Este caso de uso inicia cuando el Administrador solicita actualizar la clave de un usuario.	2. Muestra los usuarios registrados.
	3. Indica el usuario cuya clave desea modificar.	4. Solicita establecer la nueva clave y confirmarla.
	5. Ingresa la nueva clave y la confirma.	
	6. Solicita actualizar la clave del usuario.	7. Valida los datos ingresados.
		8. Actualiza la clave del usuario.
		9. Informa al Administrador que el proceso de actualización ha finalizado correctamente.
	10. El caso de uso termina.	
Poscondiciones:	Se ha actualizado la clave de un usuario.	
Extensiones:	Paso	Acción
	3, 5, 6	a. Si el Administrador solicita cancelar el proceso. 1. S: El caso de uso termina.
	7	a. Si los datos ingresados no son válidos. 1. S: Solicita al Administrador que verifique los datos ingresados y le permite modificarlos. 2. S: El caso de uso continúa (paso 6).
	9	a. Si no se logró actualizar la información. 1. S: Informa de esta situación al Administrador y le permite volver a solicitar actualizar la clave del usuario. 2. S: El caso de uso continúa (paso 6).

Tabla 12.4. Descripción del caso de uso Actualizar Claves de Usuario

CASO DE USO : GENERAR REPORTE USUARIOS REGISTRADOS		
Actor: Administrador (A)		
Propósito:	Generar un reporte de los usuarios registrados.	
Descripción: Genera un reporte que permite conocer qué usuarios se encuentran actualmente registrados.		
Precondición:	El usuario debe estar autenticado con el perfil administrador, además deben existir usuarios registrados.	
Secuencia Normal: (eventos)	Acción del Actor	Respuesta de la Solución Computacional
	1. Este caso de uso inicia cuando el Administrador solicita generar el reporte de usuarios actualmente registrados.	2. Muestra una lista de los usuarios registrados.
		3. El caso de uso termina.
Poscondiciones:	Se muestra el reporte de usuarios actualmente registrados.	
Extensiones:	Paso	Acción
	2	a. Si ocurrió un error al generar el reporte. 1. S: Informa de esta situación al Administrador. 2. S: El caso de uso termina.

Tabla 12.5. Descripción del caso de uso Generar Reporte Usuarios Registrados

Anexo 13

IDENTIFICACIÓN DE CLASES DE ANÁLISIS DE LA SOLUCIÓN COMPUTACIONAL

13.1 Caso de uso Registrar Usuario

La Figura 13.1 muestra las clases de análisis que participan en la realización del caso de uso Registrar Usuario.

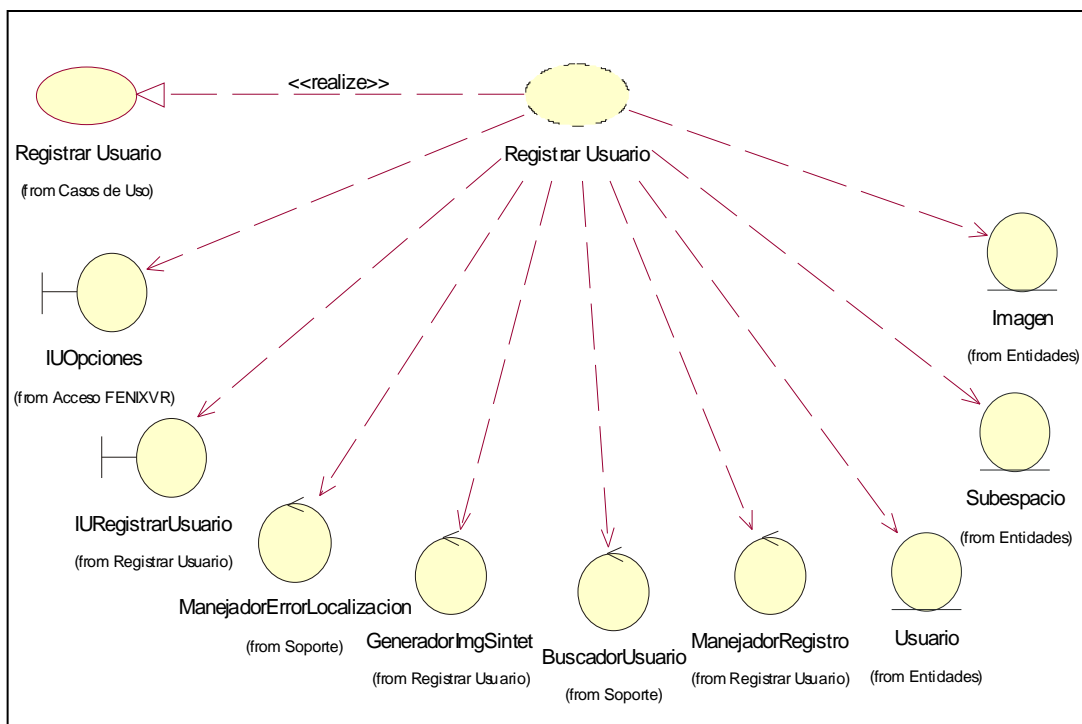


Figura 13.1. Realización del caso de uso *Registrar Usuario* - Análisis

A continuación se describen las responsabilidades asignadas a cada clase identificada:

CLASE	RESPONSABILIDAD
INTERFAZ	
IUOpciones	Esta clase tiene como responsabilidad presentar todas las opciones disponibles para el Operador.
IURegistrarUsuario	Esta clase tiene como responsabilidad permitir ingresar la información del nuevo usuario (datos e imágenes del rostro), presentar las tareas asociadas al proceso de registro que deben ser solicitadas por el Operador, delegar dichas tareas a otras clases, presentar las imágenes relacionadas al proceso de registro y validar los datos de entrada (formato) y el número de imágenes.
CONTROL	
ManejadorErrorLocalizacion	Esta clase tiene como responsabilidad gestionar los mensajes de error retornados por el proceso de localización de rostros si no se ha encontrado un rostro en la imagen analizada.
GeneradorImgSintet	Esta clase tiene como responsabilidad generar imágenes sintéticas a partir de una imagen inicial. Se entiende por imagen sintética a una variación de la imagen, es decir la imagen inicial con alguna transformación geométrica (traslación en alguna dirección).
BuscadorUsuario	Esta clase tiene como responsabilidad consultar si un usuario existe, empleando una clave para la búsqueda.
ManejadorRegistro	Esta clase tiene como responsabilidad controlar y coordinar el almacenamiento de toda la información del nuevo usuario.
ENTIDAD	
Usuario	Esta clase tiene como responsabilidad contener los datos de un usuario.
Imagen	Esta clase tiene como responsabilidad contener las imágenes del rostro de un usuario.
Subespacio	Esta clase tiene como responsabilidad contener las imágenes sintéticas del rostro de un usuario.

Tabla 13.1. Descripción de las clases de análisis identificadas para el caso de uso *Registrar Usuario*

13.2 Caso de uso Obtener Imagen

La Figura 13.2 muestra las clases de análisis que participan en la realización del caso de uso Obtener Imagen.

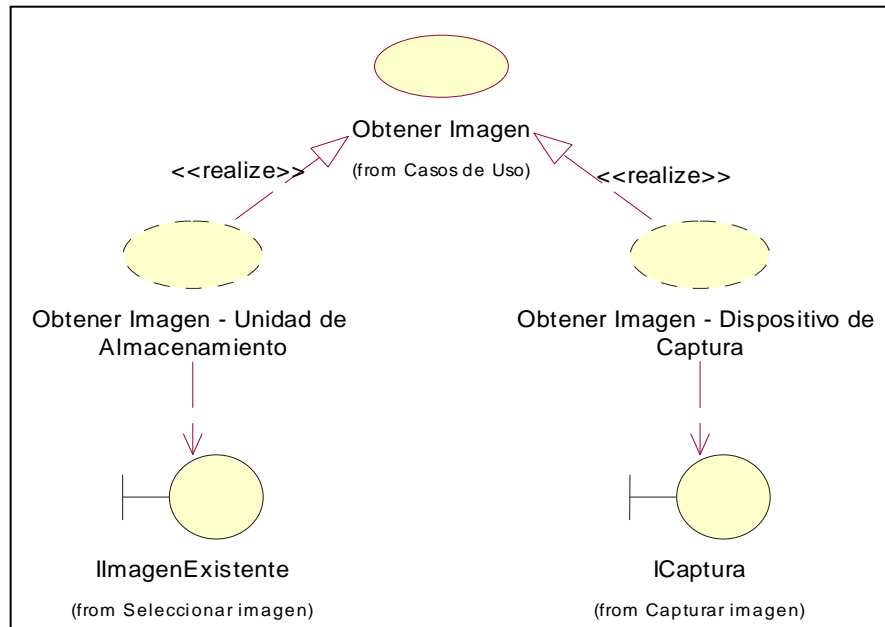


Figura 13.2. Realización del caso de uso *Obtener Imagen* - Análisis

A continuación se describen las responsabilidades asignadas a cada clase identificada:

CLASE	RESPONSABILIDAD
INTERFAZ	
IImagenExistente	Esta clase tiene como responsabilidad permitir seleccionar la ubicación y el nombre de la imagen que se desea utilizar, obtenerla desde la ubicación especificada y conocer su tamaño.
ICaptura	Esta clase tiene como responsabilidad establecer la conexión con el dispositivo de captura, adquirir una imagen a través de éste, conocer el tamaño de dicha imagen y finalizar la conexión con el dispositivo.

Tabla 13.2. Descripción de las clases de análisis identificadas para el caso de uso *Obtener Imagen*

13.3 Caso de uso Detectar Cámara

La Figura 13.3 muestra las clases de análisis que participan en la realización del caso de uso Detectar Cámara.

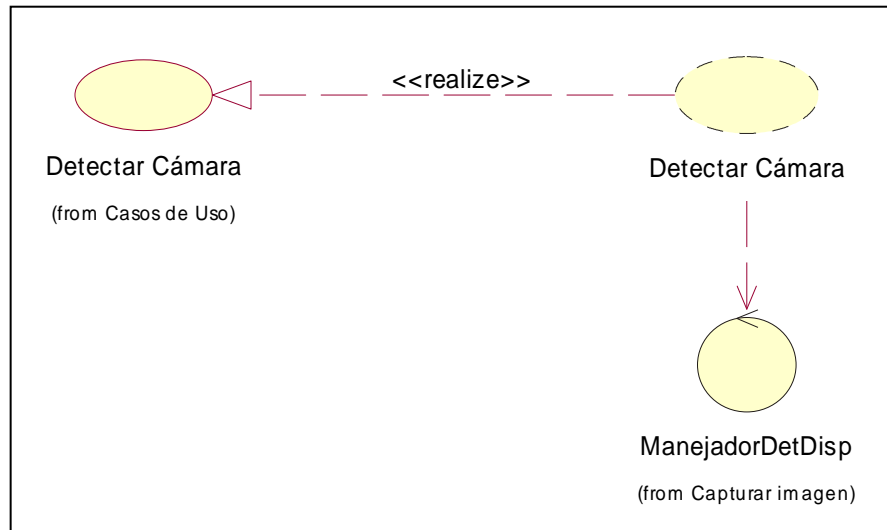


Figura 13.3. Realización del caso de uso *Detectar Cámara* - Análisis

A continuación se describen las responsabilidades asignadas a cada clase identificada:

CLASE	RESPONSABILIDAD
CONTROL	
ManejadorDetDisp	Esta clase tiene como responsabilidad detectar un dispositivo de captura específico que se encuentre accesible a través de una interfaz particular, es decir obtener el nombre e identificador de dicho dispositivo.

Tabla 13.3. Descripción de las clases de análisis identificadas para el caso de uso *Detectar Cámara*

13.4 Caso de uso Localizar Rostro

La Figura 13.4 muestra las clases de análisis que participan en la realización del caso de uso Localizar Rostro.

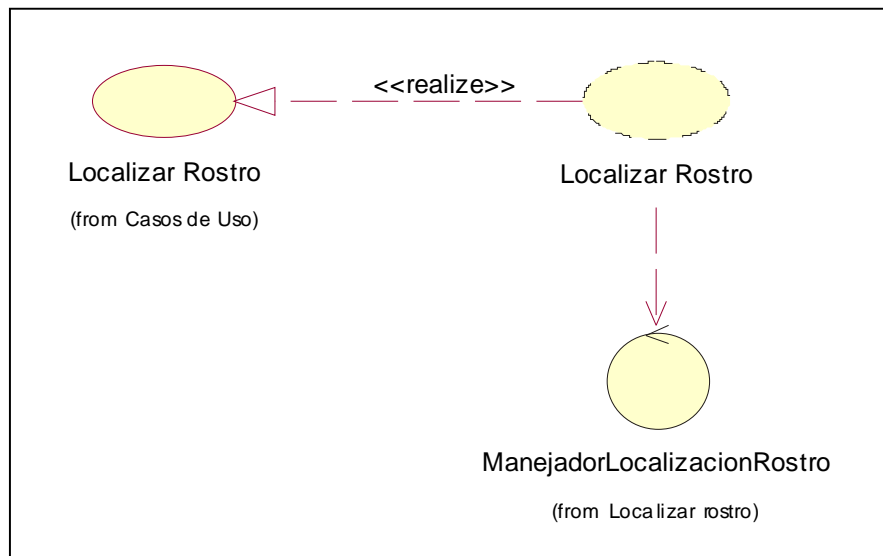


Figura 13.4. Realización del caso de uso *Localizar Rostro* - Análisis

A continuación se describen las responsabilidades asignadas a cada clase identificada:

CLASE	RESPONSABILIDAD
CONTROL	
ManejadorLocalizacionRostro	Esta clase tiene como responsabilidad controlar y coordinar el proceso de localización de un rostro en una imagen (preproceso, localización, extracción del rostro, normalización del rostro) y presentar la ubicación de dicho rostro en la imagen.

Tabla 13.4. Descripción de las clases de análisis identificadas para el caso de uso *Localizar Rostro*

13.5 Caso de uso Verificar Identidad

La Figura 13.5 muestra las clases de análisis que participan en la realización del caso de uso Verificar Identidad.

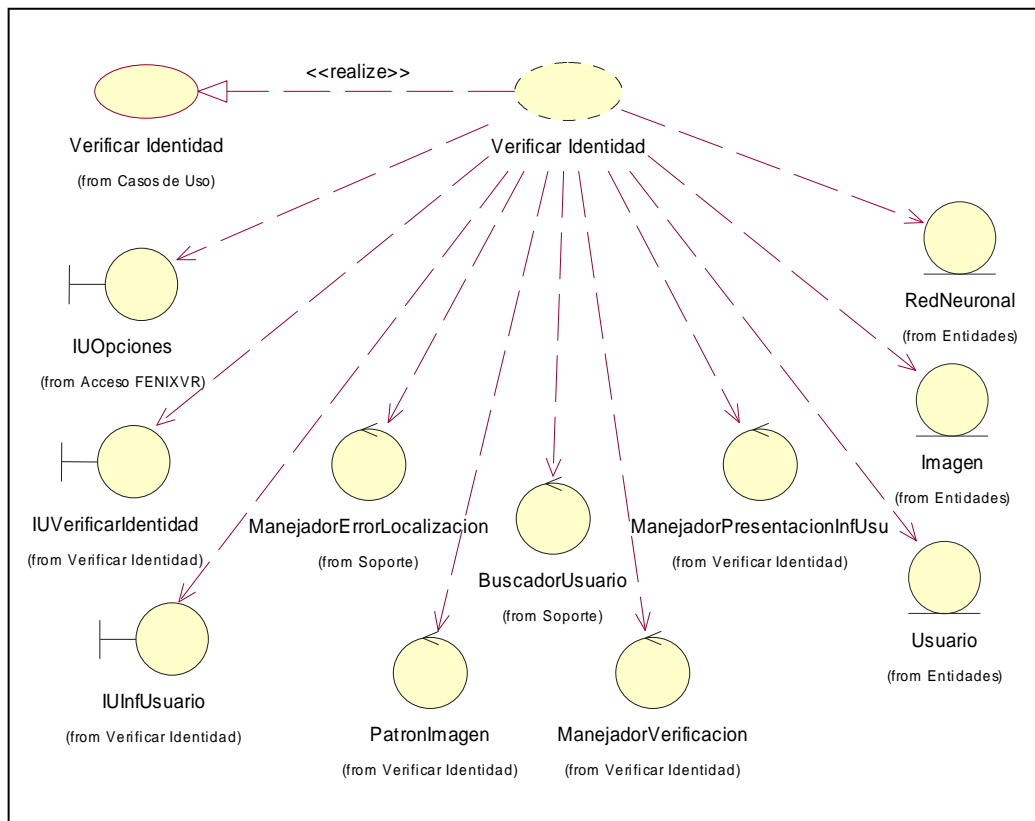


Figura 13.5. Realización del caso de uso *Verificar Identidad* - Análisis

A continuación se describen las responsabilidades asignadas a cada clase identificada:

CLASE	RESPONSABILIDAD
INTERFAZ	
IUOpciones	Esta clase tiene como responsabilidad presentar todas las opciones disponibles para el Operador.
IUVerificarIdentidad	Esta clase tiene como responsabilidad permitir ingresar la información necesaria para la verificación (clave e imagen del usuario a autenticar), presentar las tareas asociadas al proceso de verificación que deben ser

	solicitadas por el Operador, delegar dichas tareas a otras clases, presentar las imágenes relacionadas al proceso de verificación y validar la clave ingresada (formato).
IUInfUsuario	Esta clase tiene como responsabilidad presentar los datos (nombre, apellido, clave, género) y las imágenes almacenadas del usuario cuya identidad ha sido previamente verificada.
CONTROL	
ManejadorErrorLocalizacion	Esta clase tiene como responsabilidad gestionar los mensajes de error retornados por el proceso de localización de rostros si no se ha encontrado un rostro en la imagen analizada.
PatronImagen	Esta clase tiene como responsabilidad obtener las características más relevantes del rostro y realizar su reconstrucción a partir de dichas características.
BuscadorUsuario	Esta clase tiene como responsabilidad consultar si un usuario existe, empleando una clave para la búsqueda.
ManejadorVerificacion	Esta clase tiene como responsabilidad controlar y coordinar la tarea de verificación de identidad de un usuario a partir del patrón extraído de la imagen del rostro y de la clave ingresada.
ManejadorPresentacionInfUsu	Esta clase tiene como responsabilidad obtener toda la información almacenada del usuario cuya identidad ha sido previamente verificada.
ENTIDAD	
Usuario	Esta clase tiene como responsabilidad contener los datos de un usuario.
Imagen	Esta clase tiene como responsabilidad contener las imágenes del rostro de un usuario.
RedNeuronal	Esta clase tiene como responsabilidad contener a la red neuronal (neuronas, pesos, conexiones) empleada para realizar la tarea de verificación de identidad.

Tabla 13.5. Descripción de las clases de análisis identificadas para el caso de uso *Verificar Identidad*

13.6 Caso de uso Entrenar Red Neuronal

La Figura 13.6 muestra las clases de análisis que participan en la realización del caso de uso Entrenar Red Neuronal.

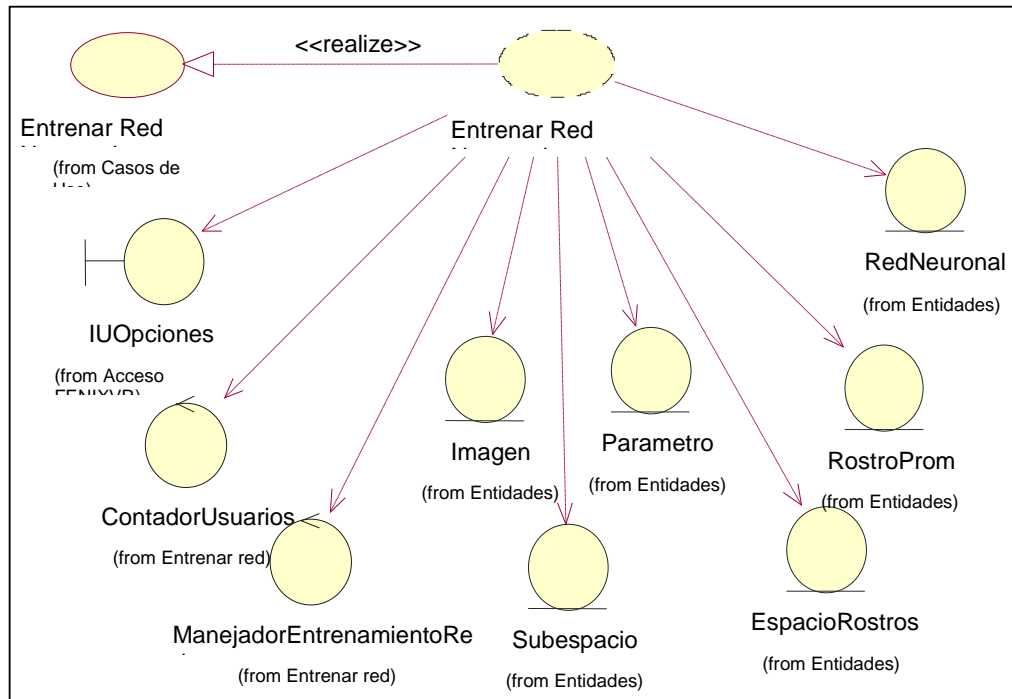


Figura 13.6. Realización del caso de uso *Entrenar Red Neuronal* - Análisis

A continuación se describen las responsabilidades asignadas a cada clase identificada:

CLASE	RESPONSABILIDAD
INTERFAZ	
IUOpciones	Esta clase tiene como responsabilidad presentar todas las opciones disponibles para el Operador.
CONTROL	
ContadorUsuarios	Esta clase tiene como responsabilidad consultar si existe un número adecuado de usuarios previamente registrados, para realizar el entrenamiento de la red neuronal. El número establecido es más de tres usuarios.
ManejadorEntrenamientoRed	Esta clase tiene como responsabilidad la creación del espacio de rostros y los vectores de características de los usuarios registrados, la creación y entrenamiento de

	la red neuronal y el almacenamiento de la red entrenada, el espacio de rostros y el rostro promedio.
ENTIDAD	
Imagen	Esta clase tiene como responsabilidad contener las imágenes del rostro de un usuario.
Subespacio	Esta clase tiene como responsabilidad contener las imágenes sintéticas del rostro de un usuario.
Parametro	Esta clase tiene como responsabilidad contener los valores de los parámetros que el usuario puede modificar para la creación y entrenamiento de la red neuronal.
EspacioRostros	Esta clase tiene como responsabilidad contener los eigenvectores o eigenfaces (vectores) que forman el espacio de rostros conocidos.
RostroProm	Esta clase tiene como responsabilidad contener al vector que representa al rostro promedio empleado para la proyección de imágenes en el espacio de rostros.
RedNeuronal	Esta clase tiene como responsabilidad contener a la red neuronal (neuronas, pesos, conexiones) creada y entrenada para realizar la tarea de verificación de identidad.

Tabla 13.6. Descripción de las clases de análisis identificadas para el caso de uso *Entrenar Red Neuronal*

13.7 Caso de uso Actualizar Parámetros de la Red Neuronal

La Figura 13.7 muestra las clases de análisis que participan en la realización del caso de uso Actualizar Parámetros de la Red Neuronal.

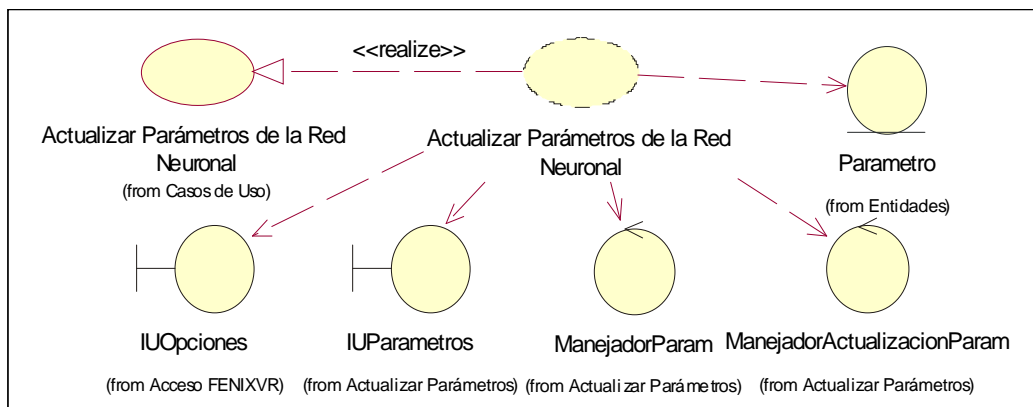


Figura 13.7. Realización *Actualizar Parámetros de la Red Neuronal* - Análisis

A continuación se describen las responsabilidades asignadas a cada clase identificada:

CLASE	RESPONSABILIDAD
INTERFAZ	
IUOpciones	Esta clase tiene como responsabilidad presentar todas las opciones disponibles para el Operador.
IUParámetros	Esta clase tiene como responsabilidad presentar los valores actuales de los parámetros que pueden ser modificados, permitir ingresar nuevos valores y validar su formato.
CONTROL	
ManejadorParam	Esta clase tiene como responsabilidad obtener los valores actuales de los parámetros.
ManejadorActualizacionParam	Esta clase tiene como responsabilidad controlar y coordinar la modificación de los parámetros.
ENTIDAD	
Parametro	Esta clase tiene como responsabilidad contener los valores de los parámetros que pueden ser modificados por un usuario.

Tabla 13.7. Descripción de las clases de análisis identificadas para el caso de uso *Actualizar Parámetros de la Red Neuronal*

13.8 Caso de uso Comprobar Perfil para Acceso

La Figura 13.8 muestra las clases de análisis que participan en la realización del caso de uso Comprobar Perfil para Acceso.

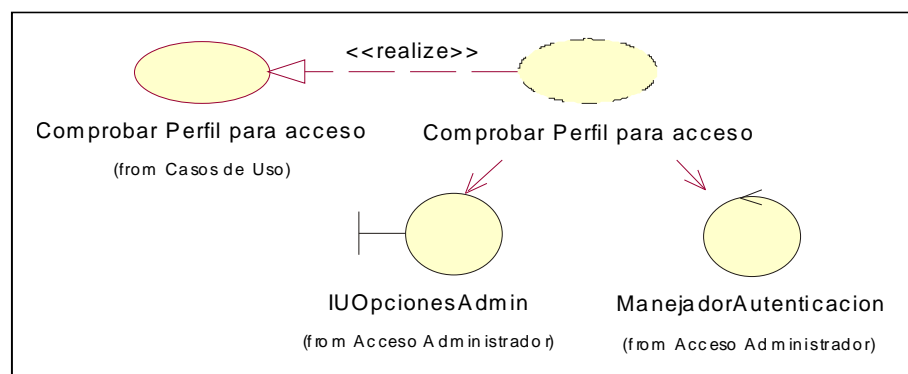


Figura 13.8. Realización del caso de uso *Comprobar Perfil para Acceso* – Análisis

A continuación se describen las responsabilidades asignadas a cada clase identificada:

CLASE	RESPONSABILIDAD
INTERFAZ	
IUOpcionesAdmin	Esta clase tiene como responsabilidad solicitar la autenticación del usuario que desea acceder a las tareas de administración y presentar todas las opciones disponibles para el Administrador.
CONTROL	
ManejadorAutenticacion	Esta clase tiene como responsabilidad comprobar si el usuario actual tiene perfil de administrador del equipo o dominio (de acuerdo a la autenticación Windows).

Tabla 13.8. Descripción de las clases de análisis identificadas para el caso de uso *Comprobar Perfil para Acceso*

13.9 Caso de uso Guardar Imagen Inicial

La Figura 13.9 muestra las clases de análisis que participan en la realización del caso de uso Guardar Imagen Inicial.

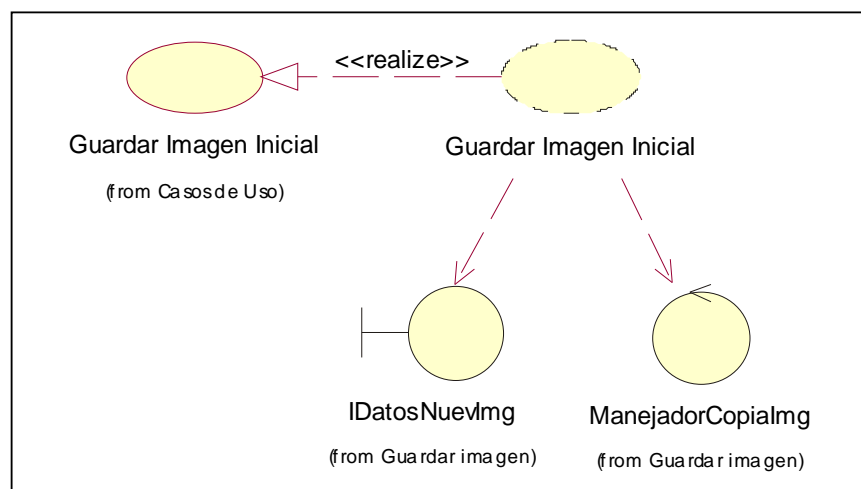


Figura 13.9. Realización del caso de uso *Guardar Imagen Inicial* - Análisis

A continuación se describen las responsabilidades asignadas a cada clase identificada:

CLASE	RESPONSABILIDAD
INTERFAZ	
IDatosNuevImg	Esta clase tiene como responsabilidad permitir ingresar o seleccionar la ubicación, el nombre y formato de la copia que se va a crear para una imagen, también valida los datos ingresados.
CONTROL	
ManejadorCopialmg	Esta clase tiene como responsabilidad crear un nuevo archivo gráfico a partir de uno ya existente empleando el nombre, formato y ubicación especificados por el usuario.

Tabla 13.9. Descripción de las clases de análisis identificadas para el caso de uso *Guardar Imagen Inicial*

13.10 Caso de uso *Guardar Imagen Rostro Localizado*

La Figura 13.10 muestra las clases de análisis que participan en la realización del caso de uso *Guardar Imagen Rostro Localizado*.

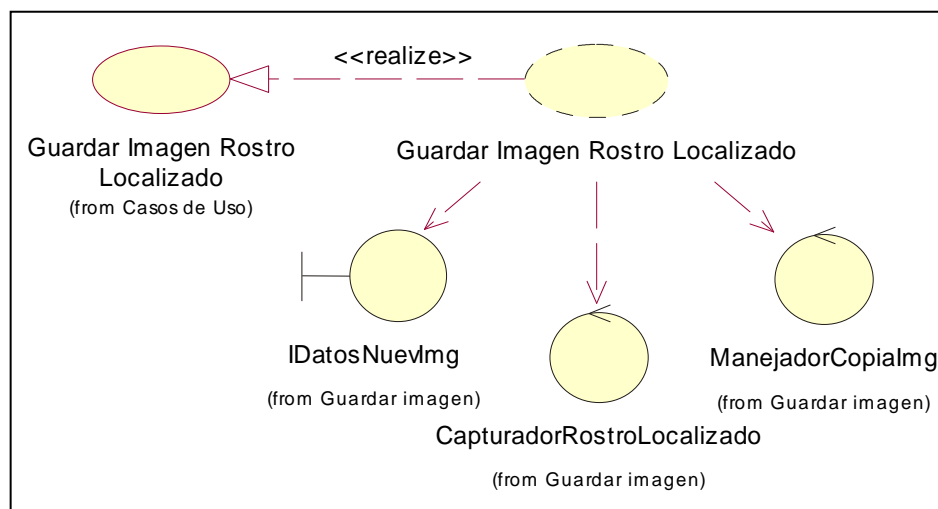


Figura 13.10. Realización del caso de uso *Guardar Imagen Rostro Localizado* – Análisis

A continuación se describen las responsabilidades asignadas a cada clase identificada:

CLASE	RESPONSABILIDAD
INTERFAZ	
IDatosNuevImg	Esta clase tiene como responsabilidad permitir ingresar o seleccionar la ubicación, el nombre y formato de la copia que se va a crear para una imagen, también valida los datos ingresados.
CONTROL	
CapturadorRostroLocalizado	Esta clase tiene como responsabilidad capturar la imagen que muestra el rostro localizado.
ManejadorCopiaImg	Esta clase tiene como responsabilidad crear un nuevo archivo gráfico a partir de uno ya existente empleando el nombre, formato y ubicación especificados por el usuario.

Tabla 13.10. Descripción de las clases de análisis identificadas para el caso de uso *Guardar Imagen Rostro Localizado*

13.11 Caso de uso Actualizar Claves de Usuario

La Figura 13.11 muestra las clases de análisis que participan en la realización del caso de uso Actualizar Claves de Usuario.

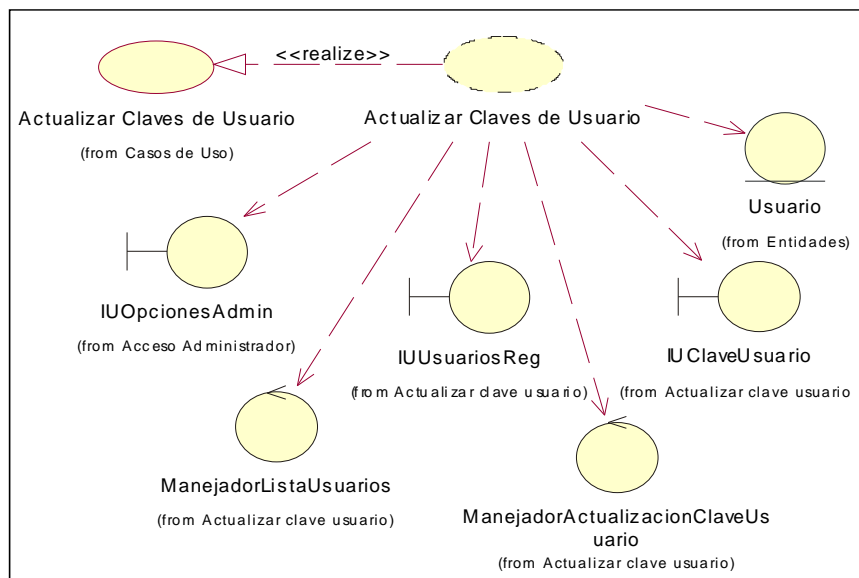


Figura 13.11. Realización del caso de uso *Actualizar Claves de Usuario* – Análisis

A continuación se describen las responsabilidades asignadas a cada clase identificada:

CLASE	RESPONSABILIDAD
INTERFAZ	
IUOpcionesAdmin	Esta clase tiene como responsabilidad solicitar la autenticación del usuario que desea acceder a las tareas de administración y presentar todas las opciones disponibles para el Administrador.
IUUsuariosReg	Esta clase tiene como responsabilidad presentar los usuarios registrados (nombre y apellido) y permitir seleccionar el usuario cuya clave se desea modificar.
IUClaveUsuario	Esta clase tiene como responsabilidad permitir ingresar la nueva clave del usuario y validar su formato.
CONTROL	
ManejadorListaUsuarios	Esta clase tiene como responsabilidad obtener el nombre y apellido de todos los usuarios registrados.
ManejadorActualizacionClaveUsuario	Esta clase tiene como responsabilidad controlar y coordinar la modificación o actualización de la clave de un usuario.
ENTIDAD	
Usuario	Esta clase tiene como responsabilidad contener los datos de un usuario.

Tabla 13.11. Descripción de las clases de análisis identificadas para el caso de uso *Actualizar Claves de Usuario*

13.12 Caso de uso Generar Reporte Usuarios Registrados

La Figura 13.12 muestra las clases de análisis que participan en la realización del caso de uso Generar Reporte Usuarios Registrados.

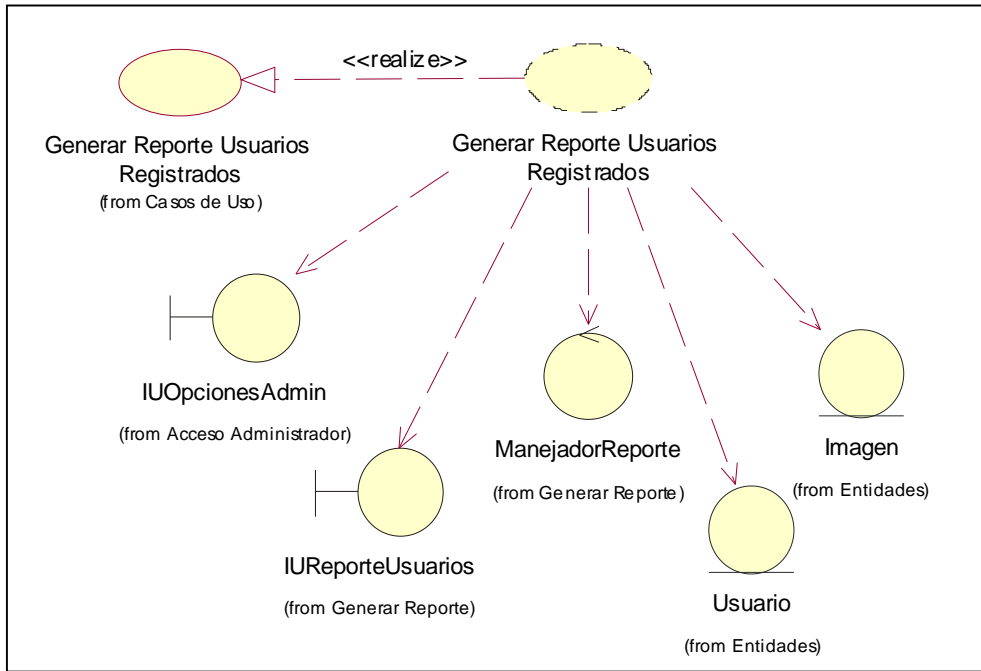


Figura 13.12. Realización del caso de uso *Generar Reporte Usuarios Registrados* - Análisis

A continuación se describen las responsabilidades asignadas a cada clase identificada:

CLASE	RESPONSABILIDAD
INTERFAZ	
IUOpcionesAdmin	Esta clase tiene como responsabilidad solicitar la autenticación del usuario que desea acceder a las tareas de administración y presentar todas las opciones disponibles para el Administrador.
IUReporteUsuarios	Esta clase tiene como responsabilidad presentar la información de los usuarios registrados (nombre, apellido, género, clave y una imagen del rostro por usuario).

CONTROL	
ManejadorReporte	Esta clase tiene como responsabilidad controlar y coordinar la generación del reporte de los usuarios registrados.
ENTIDAD	
Usuario	Esta clase tiene como responsabilidad contener los datos de un usuario.
Imagen	Esta clase tiene como responsabilidad contener las imágenes del rostro de un usuario.

Tabla 13.12. Descripción de las clases de análisis identificadas para el caso de uso *Generar Reporte Usuarios Registrados*.

Anexo 14

REALIZACIÓN DE CASOS DE USO - ANÁLISIS

14.1 Diagramas de Colaboración – Análisis: Caso de uso Registrar Usuario

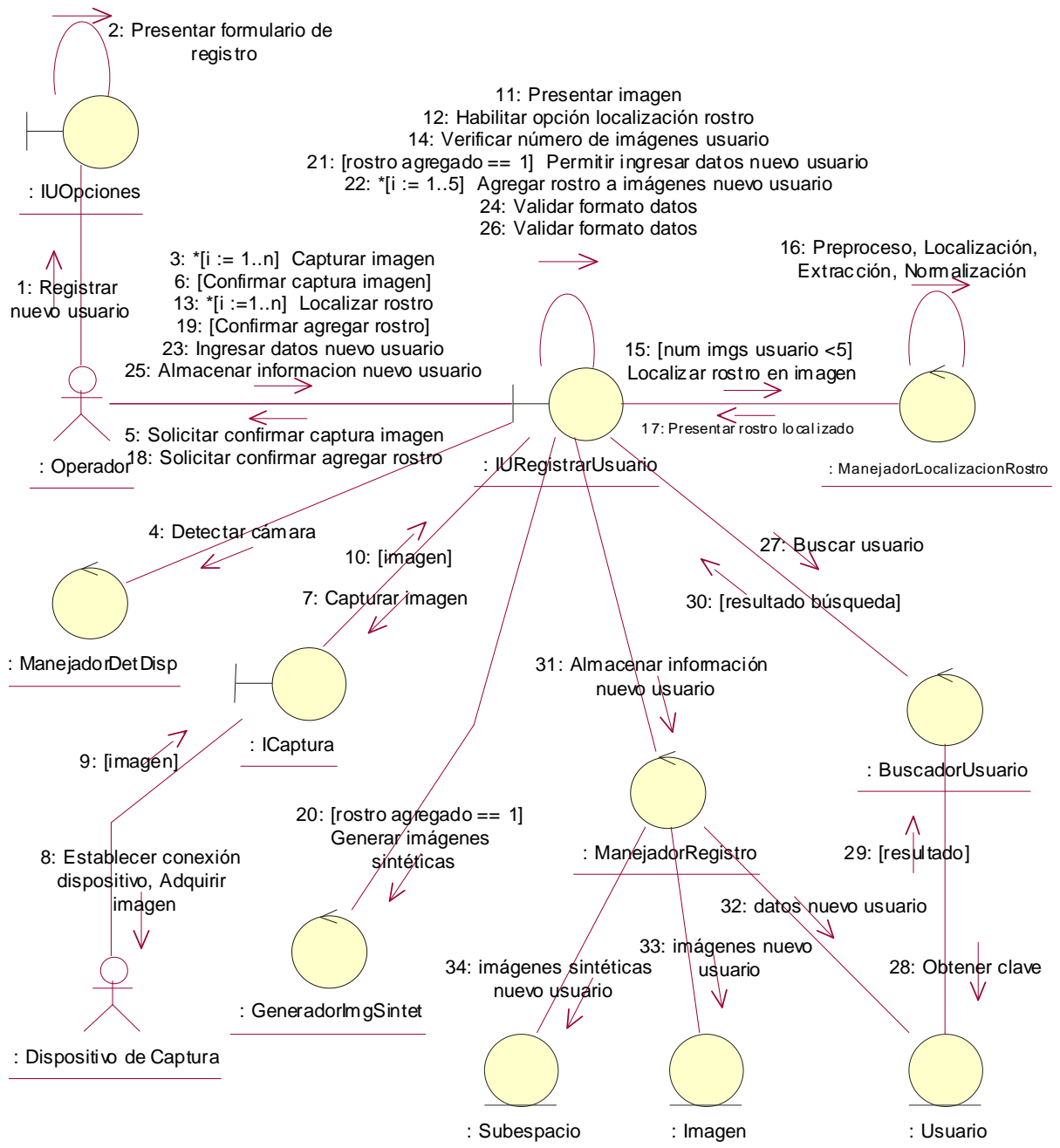


Figura 14.1. Diagrama de Colaboración - Análisis: Caso de uso *Registrar Usuario* (Flujo Básico – Capturar imagen)

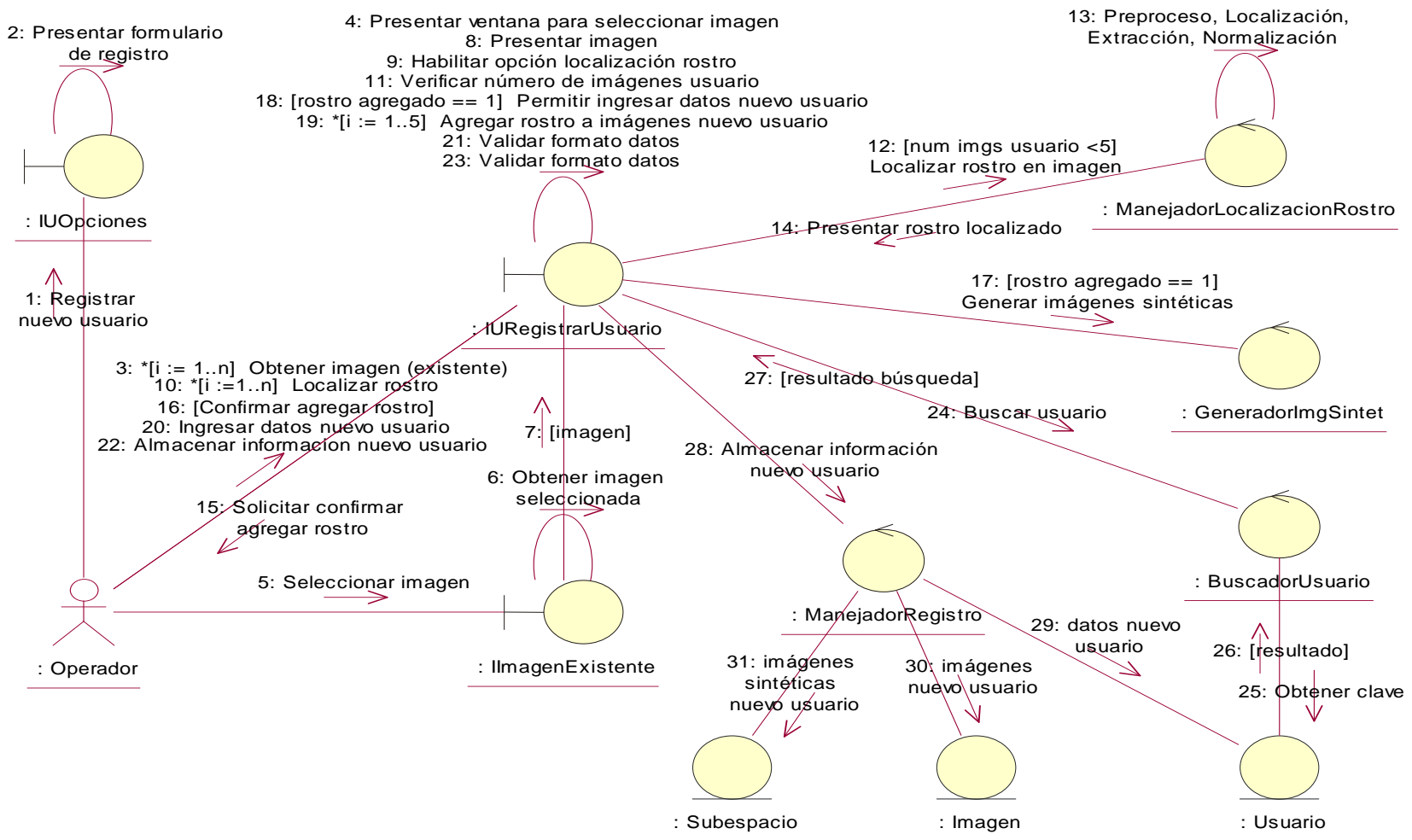


Figura 14.2. Diagrama de Colaboración - Análisis: Caso de uso *Registrar Usuario* (Flujo Básico - Imagen existente)

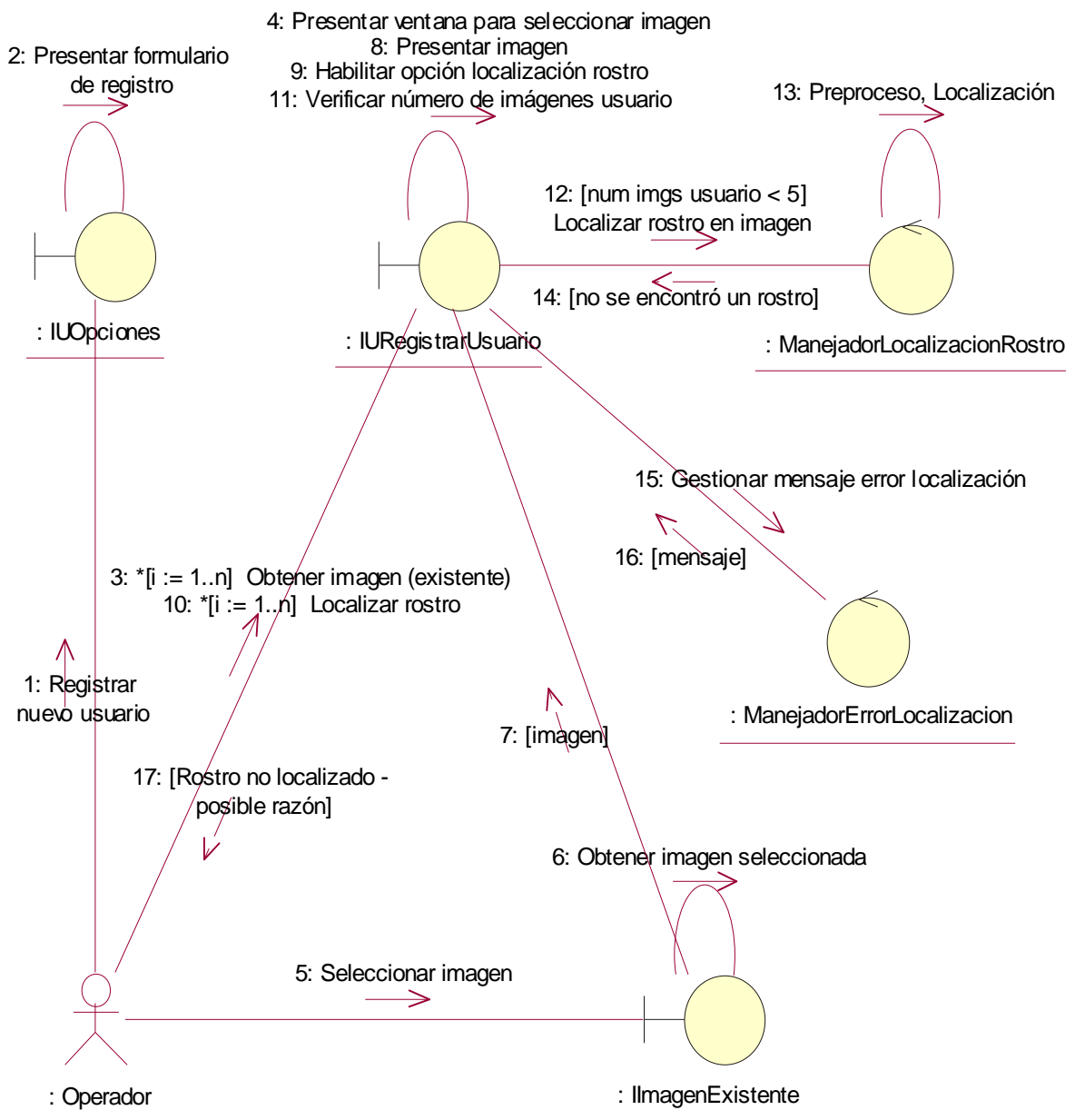


Figura 14.3. Diagrama de Colaboración - Análisis: Caso de uso *Registrar Usuario* (Extensión – No se localizó ningún rostro en la imagen)

14.2 Diagramas de Colaboración – Análisis: Caso de uso Verificar Identidad

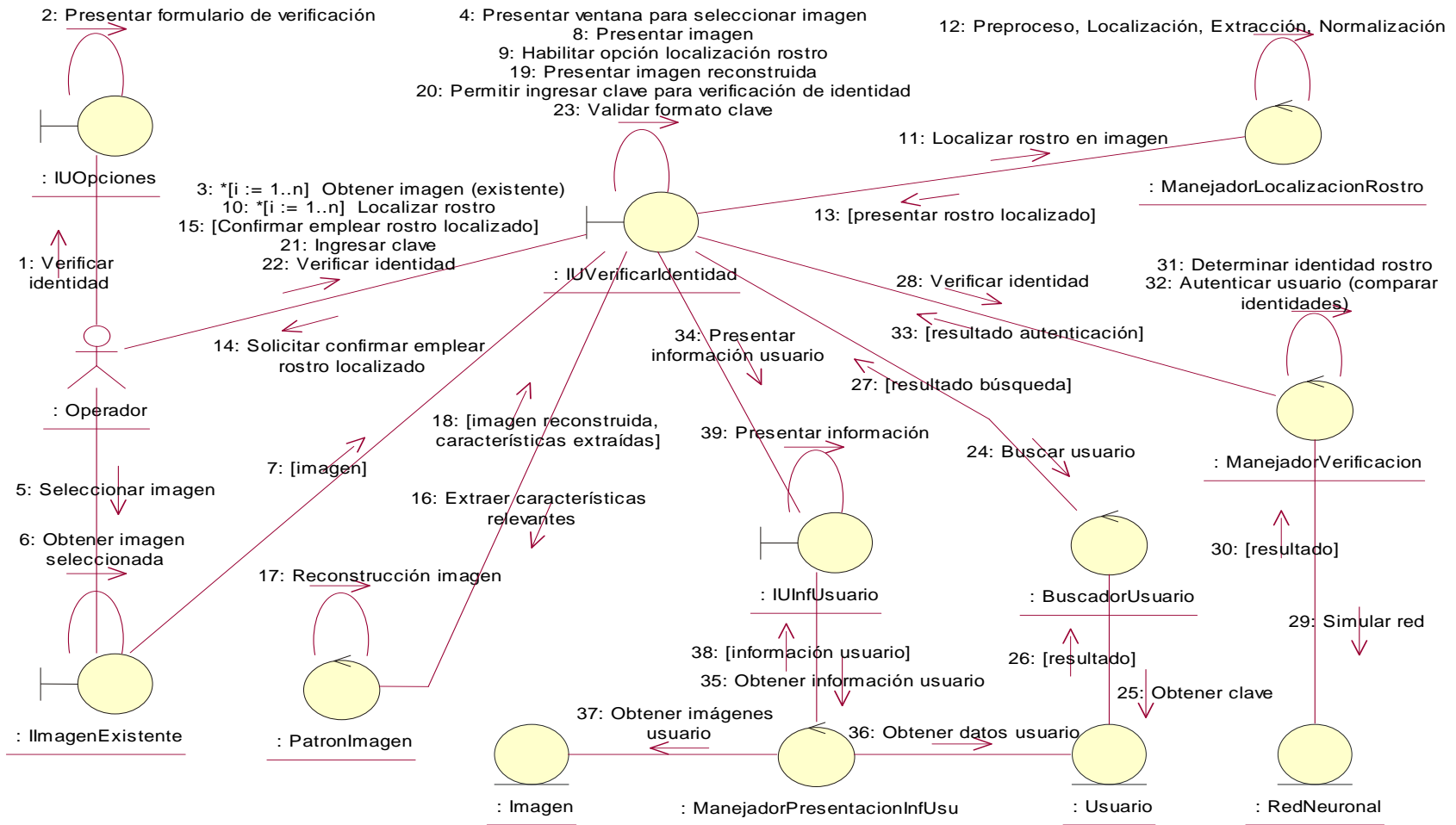


Figura 14.4. Diagrama de Colaboración - Análisis: Caso de uso Verificar Identidad (Flujo Básico - Imagen existente)

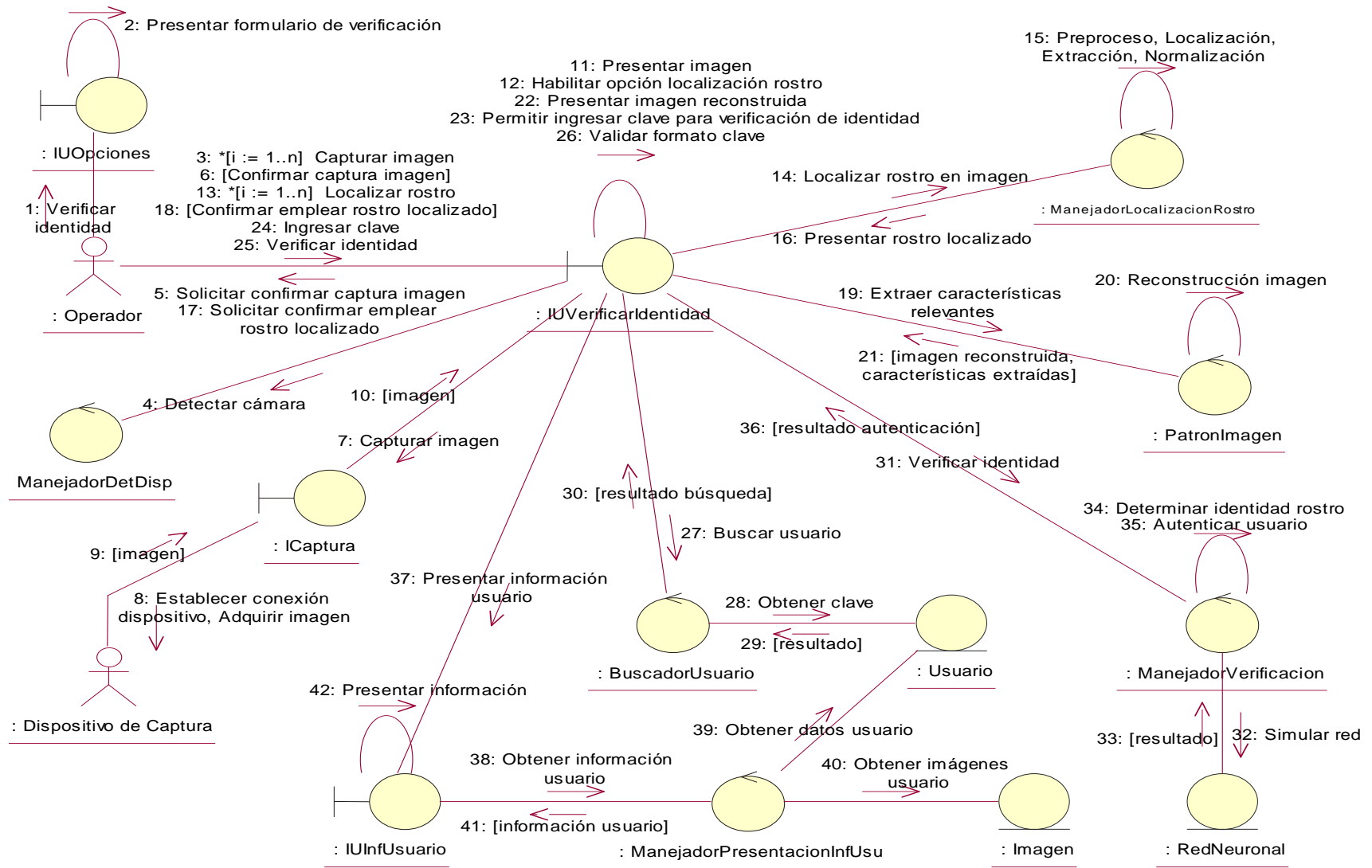


Figura 14.5. Diagrama de Colaboración - Análisis: Caso de uso *Verificar Identidad* (Flujo Básico – Capturar imagen)

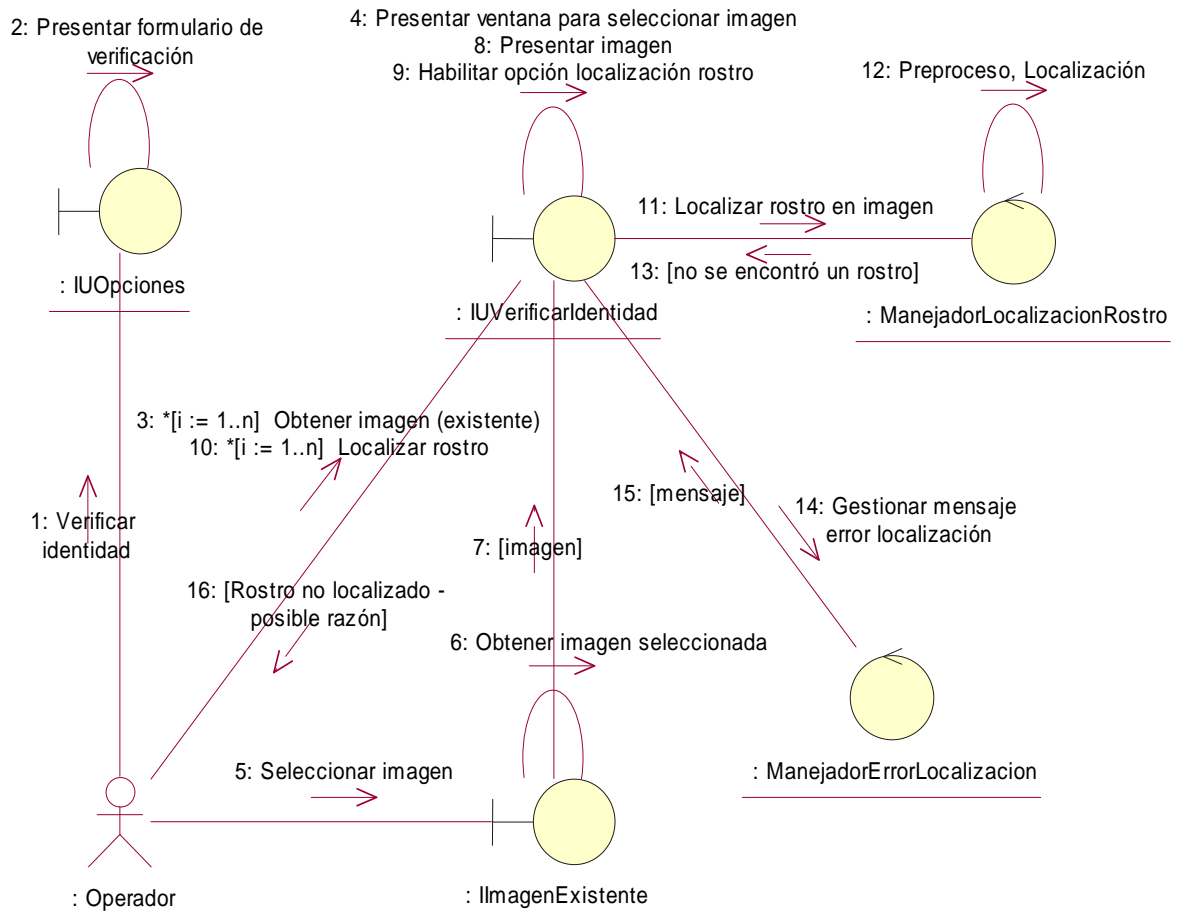


Figura 14.6. Diagrama de Colaboración - Análisis: Caso de uso *Verificar Identidad* (Extensión – No se localizó ningún rostro en la imagen)

14.3 Diagrama de Colaboración – Análisis: Caso de uso Obtener Imagen (*)

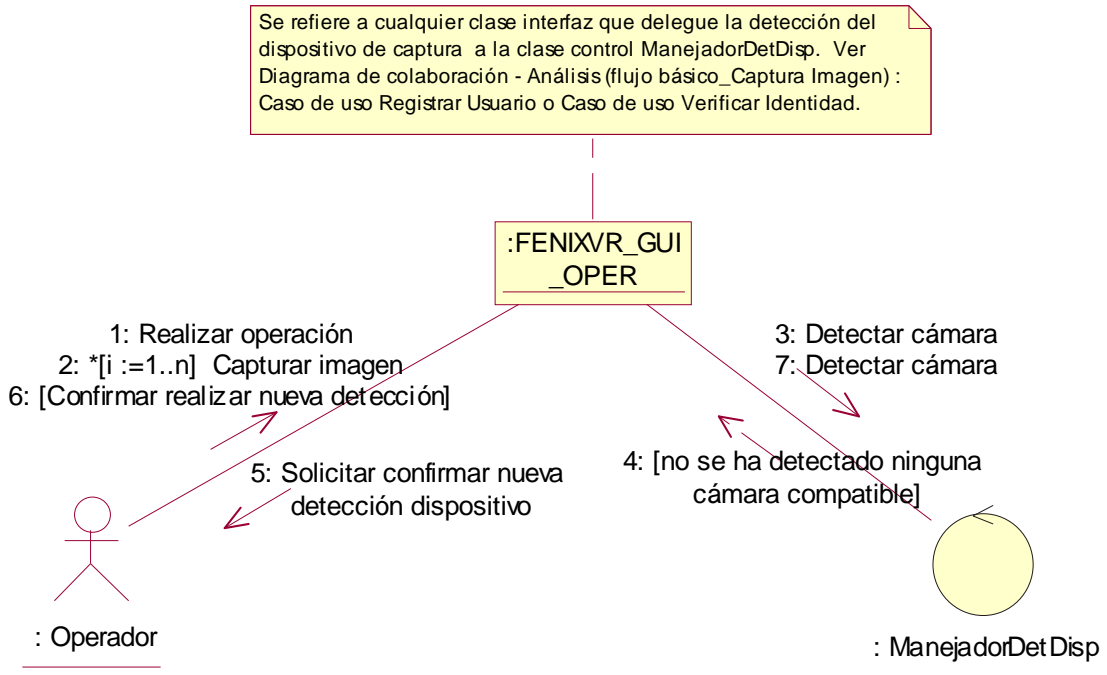


Figura 14.7. Diagrama de Colaboración - Análisis: Caso de uso *Obtener Imagen* – Realización: *Dispositivo de Captura* (Extensión – No se ha detectado un dispositivo)

(*) Aunque este caso de uso no puede ser considerado concreto en el ámbito de la solución computacional, se realiza el diagrama de Colaboración - Análisis para un escenario alternativo (*no se ha detectado un dispositivo*) debido a que el escenario descrito es resultado de un flujo alternativo de este caso de uno y no de otros casos de uso que lo emplean.

14.4 Diagrama de Colaboración – Análisis: Caso de uso Entrenar Red Neuronal

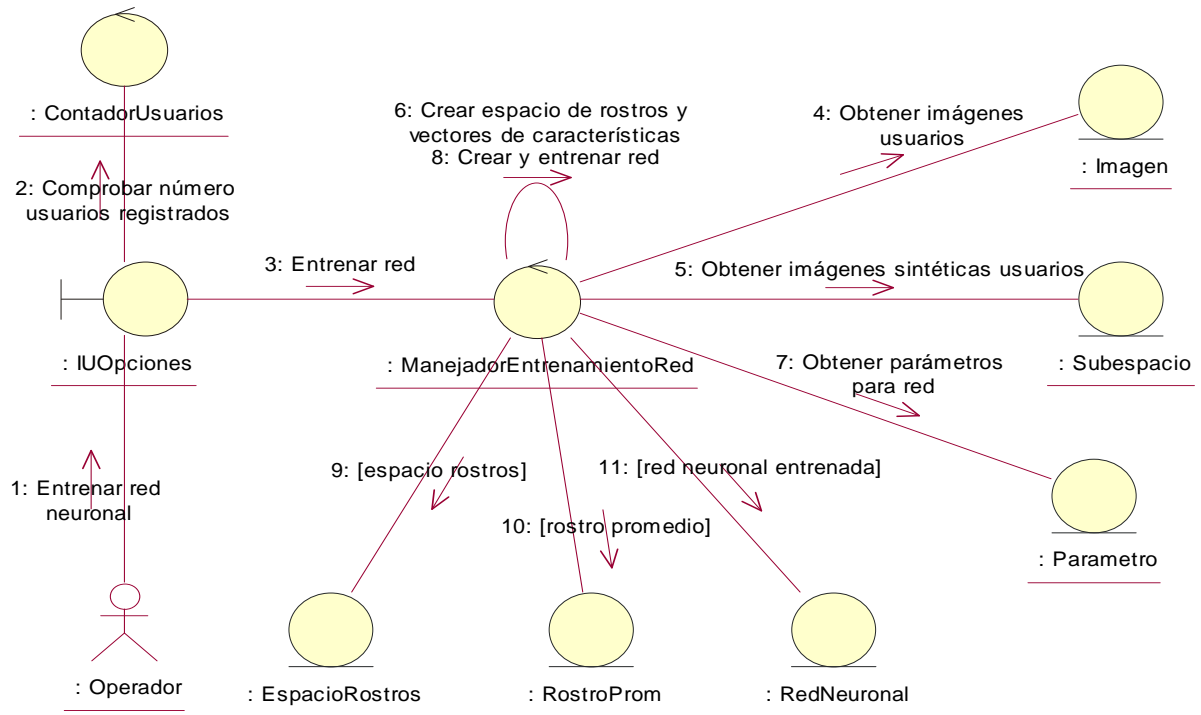


Figura 14.8. Diagrama de Colaboración - Análisis: Caso de uso *Entrenar Red Neuronal* (Flujo Básico)

14.5 Diagrama de Colaboración – Análisis: Caso de uso Actualizar Parámetros de la Red Neuronal

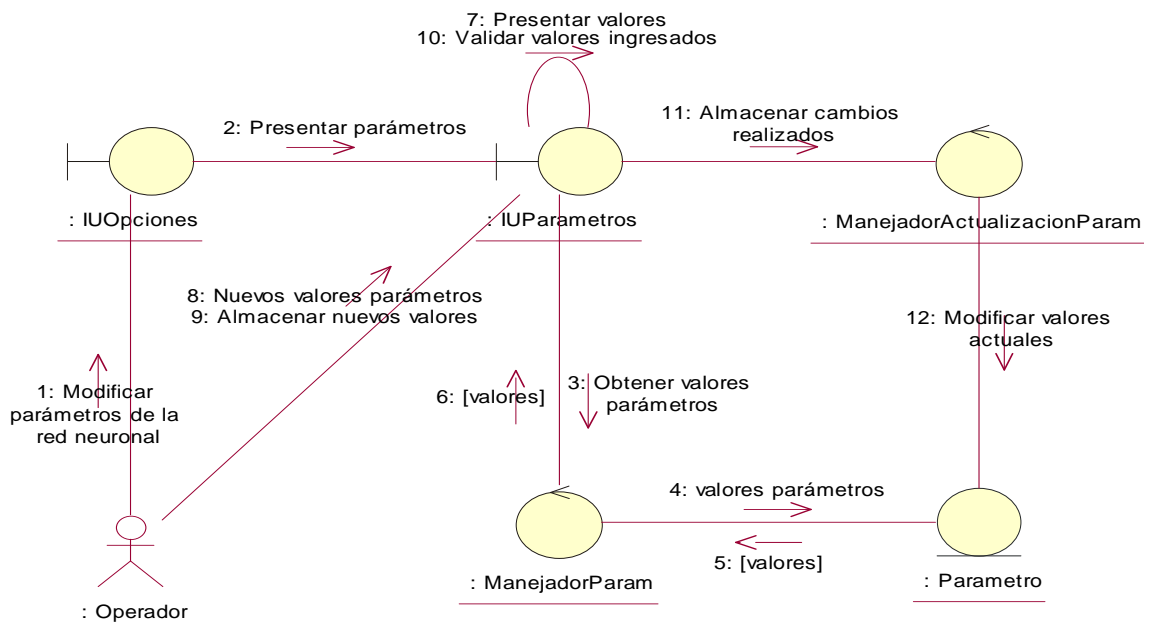


Figura 14.9. Diagrama de Colaboración - Análisis: Caso de uso *Actualizar Parámetros de la Red Neuronal* (Flujo Básico)

14.6 Diagramas de Colaboración – Análisis: Caso de uso Comprobar Perfil para acceso

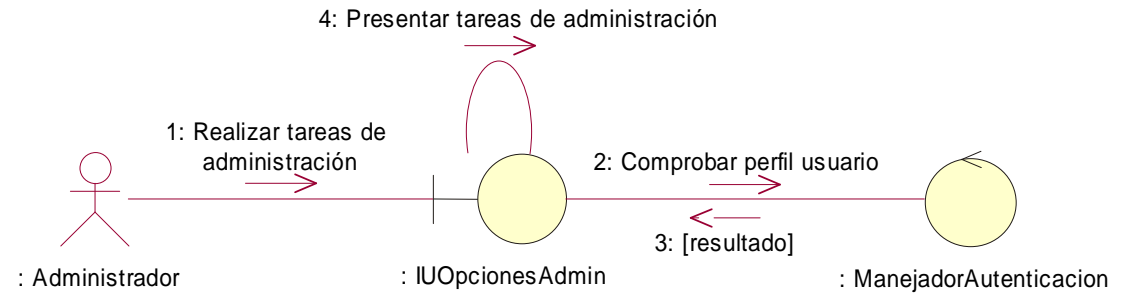


Figura 14.10. Diagrama de Colaboración - Análisis: Caso de uso *Comprobar Perfil para acceso* (Flujo Básico – Caso de uso iniciado por el Administrador)

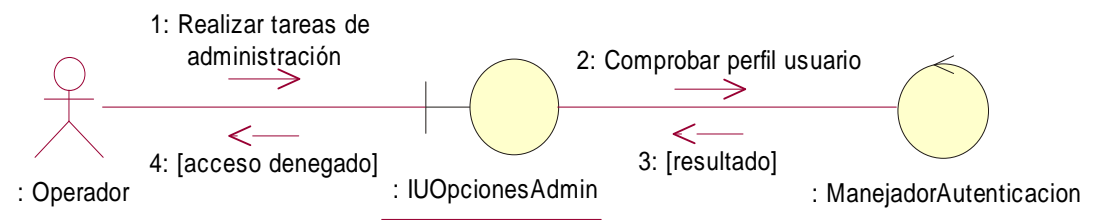


Figura 14.11. Diagrama de Colaboración - Análisis: Caso de uso *Comprobar Perfil para acceso* (Extensión – El usuario no cuenta con el perfil adecuado)

14.7 Diagrama de Colaboración – Análisis: Caso de uso Guardar Imagen Inicial

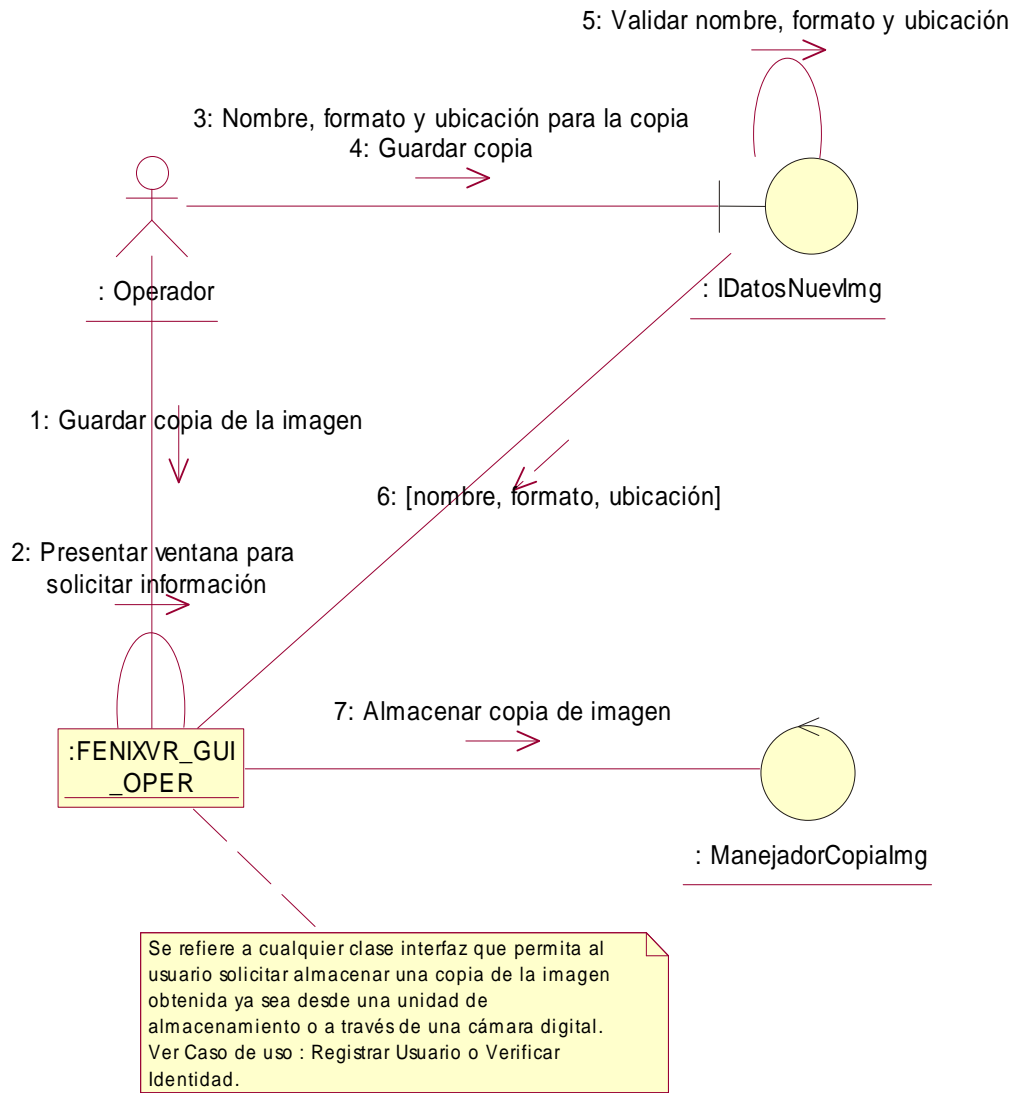


Figura 14.12. Diagrama de Colaboración - Análisis: Caso de uso *Guardar Imagen Inicial* (Flujo Básico)

14.8 Diagrama de Colaboración – Análisis: Caso de uso *Guardar Imagen Rostro Localizado*

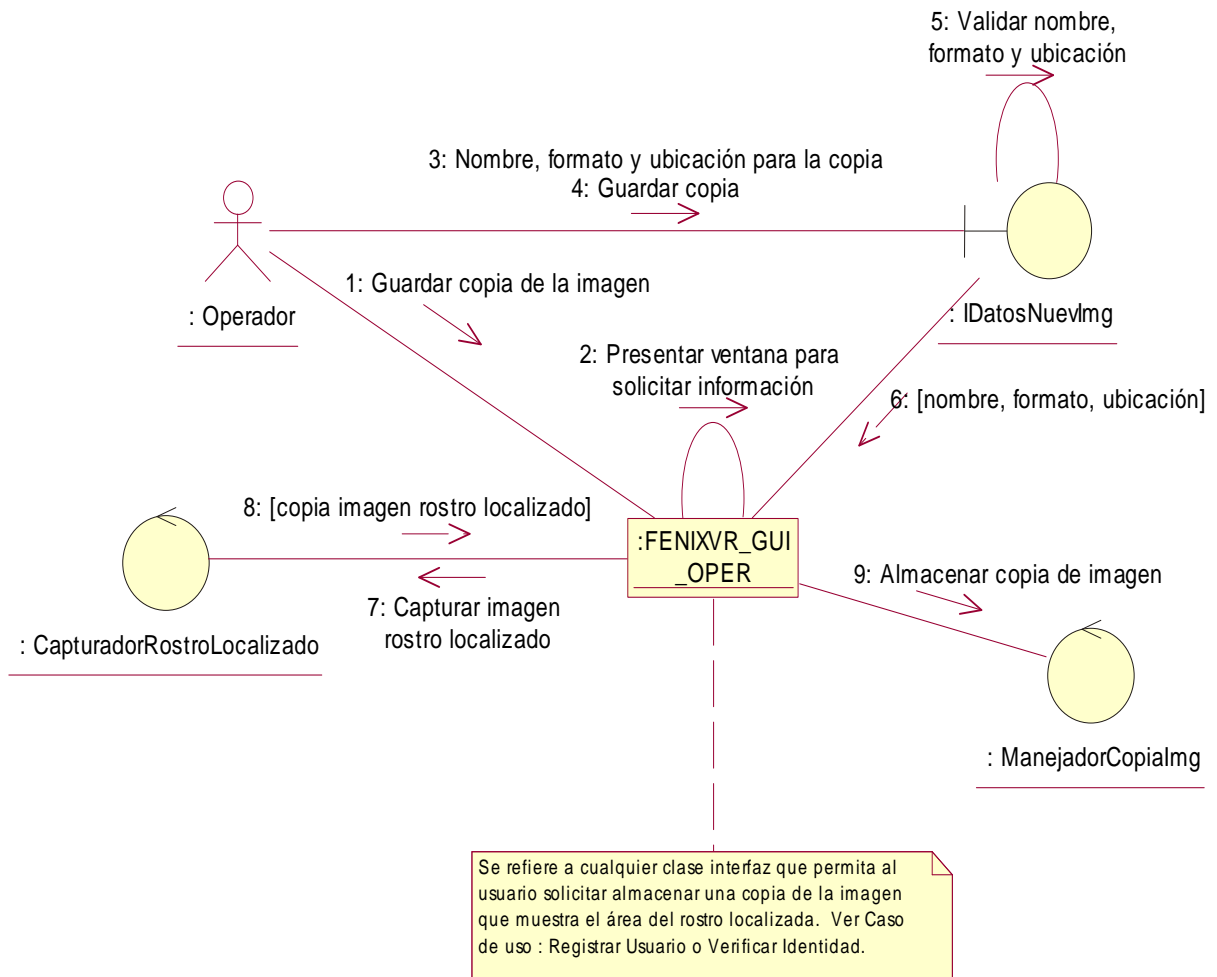


Figura 14.13. Diagrama de Colaboración - Análisis: Caso de uso *Guardar Imagen Rostro Localizado* (Flujo Básico)

14.9 Diagrama de Colaboración – Análisis: Caso de uso Actualizar Claves de Usuario

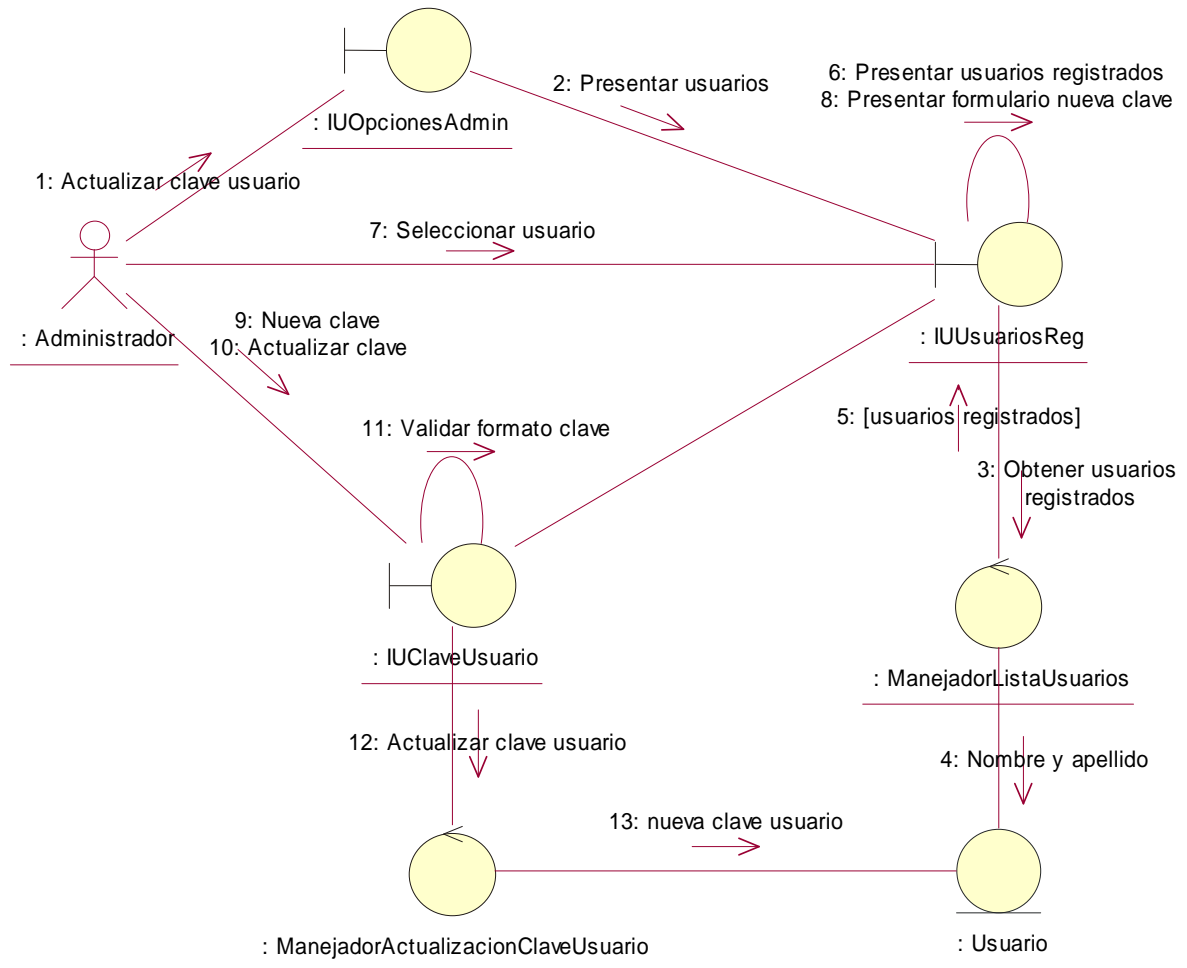


Figura 14.14. Diagrama de Colaboración - Análisis: Caso de uso *Actualizar Claves de Usuario* (Flujo Básico)

14.10 Diagrama de Colaboración – Análisis: Caso de uso *Generar Reporte Usuarios Registrados*

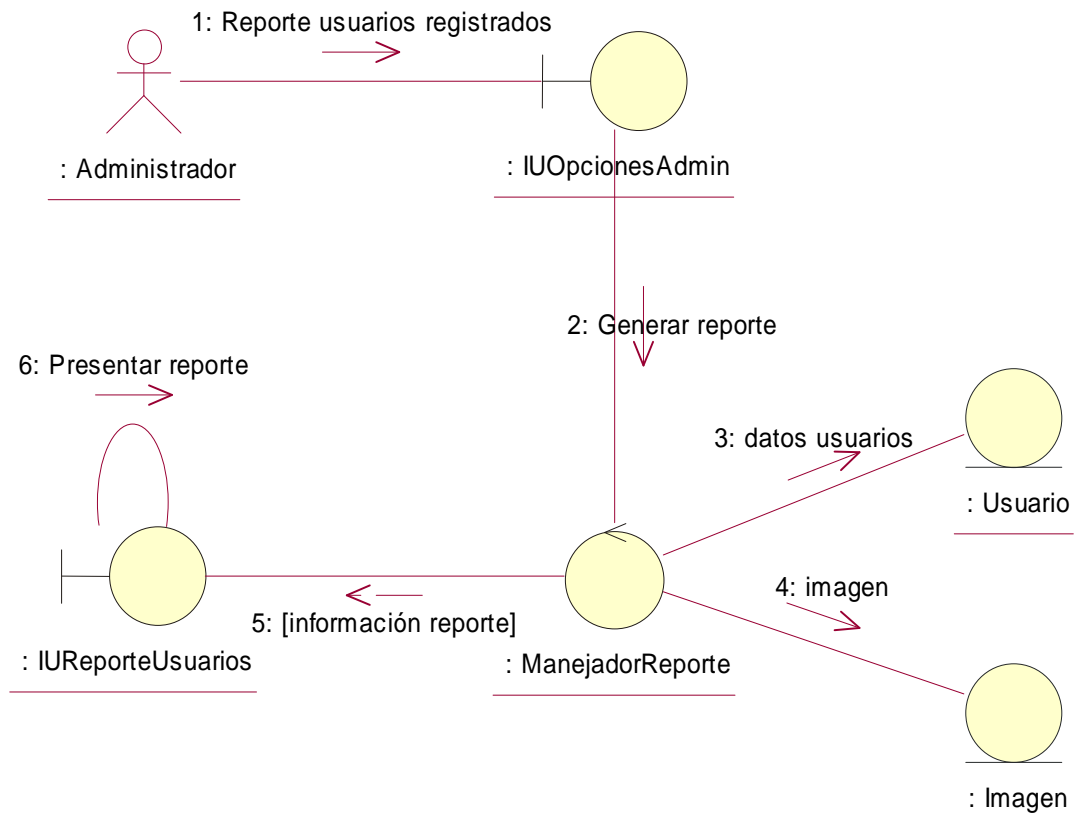


Figura 14.15. Diagrama de Colaboración - Análisis: Caso de uso *Generar Reporte Usuarios Registrados* (Flujo Básico)

Anexo 15

IDENTIFICACIÓN DE CLASES DE DISEÑO SOLUCIÓN COMPUTACIONAL

15.1 Identificación de Clases de Diseño – Interfaz

A continuación, en las Tablas 15.1 hasta 15.7, se detalla los atributos y operaciones de las clases de diseño de tipo interfaz.

CLASE : ImagenExistente		
Refinamiento clase de análisis : ImagenExistente		
ATRIBUTOS		
Nombre	Tipo	Descripción
+ img_orig_img	Matriz	Imagen obtenida desde una ubicación especificada. Imagen que contiene un rostro humano.
- ruta_img_txt	String	Ubicación y nombre de la imagen seleccionada.
- f_i	Integer	Número de filas de la imagen.
- c_i	Integer	Número de columnas de la imagen.
MÉTODOS		
Nombre	Descripción	
+ selec_img_fnc (obj : Object) : Object	Permite seleccionar la imagen con la que se desea trabajar para esto emplea el cuadro de diálogo estándar para recuperar archivos desplegado mediante <i>uigetfile</i> . Obtiene la ubicación y el nombre (ruta) de esa imagen.	
- obt_img_selec_fnc (obj : Object, ruta_img_txt : String) : Object	Obtiene una imagen empleando la ruta que recibe como parámetro de entrada.	

Tabla 15.1. Descripción de la clase de diseño *ImagenExistente*

CLASE : ICaptura		
Refinamiento clase de análisis : ICaptura		
ATRIBUTOS		
Nombre	Tipo	Descripción
+ img_orig_img	Matriz	Imagen obtenida a través de una cámara digital. Imagen original que contiene un rostro humano.
- f_i	Integer	Número de filas de la imagen.
- c_i	Integer	Número de columnas de la imagen.
MÉTODOS		
Nombre	Descripción	
+ conect_disp_fnc (disp_id_i : Integer) : vid_Object	Establece una conexión con un dispositivo de captura y retorna un objeto de enlace de video asociado con dicho dispositivo. Recibe como parámetro de entrada el identificador correspondiente al dispositivo de captura (disp_id_i).	
+ adq_img_fnc (obj : Object, vid : Object) : Object	Adquiere una imagen a través del dispositivo con el que se ha establecido una conexión y luego finaliza dicha conexión.	

Tabla 15.2. Descripción de la clase de diseño *ICaptura*

CLASE : FENIXVR_ADM		
Refinamiento clase de análisis : IUOpcionesAdmin		
ATRIBUTOS		
Nombre	Tipo	Descripción
---	---	---
MÉTODOS		
Nombre	Descripción	
+ autent_admin_fnc() : Boolean	Solicita a una instancia de la clase AutenticacionAdmin la autenticación del usuario que desea acceder a las tareas de administración.	
+ btn_act_clav_Callback() : void	Maneja el evento relacionado a la solicitud de actualizar la clave de un usuario registrado, se crea una instancia de la clase usuarios_registrados_GUI.	

+ btn_gen_rep_Callback() : void	Maneja el evento relacionado a la solicitud de generar un reporte de usuarios registrados, se crea una instancia de la clase reporte_GUI.
+ btn_salir_Callback() : void	Destruye el objeto.

Tabla 15.3. Descripción de la clase de diseño *FENIXVR_ADM*

CLASE : IDatosNuevImg		
Refinamiento clase de análisis : IDatosNuevImg		
ATRIBUTOS		
Nombre	Tipo	Descripción
- archivo_txt	String	Nombre de la copia que se va a crear para una imagen.
- ruta_txt	String	Ubicación donde se desea almacenar la copia que se va a crear
- idx_i	Integer	Índice que permite identificar el formato seleccionado para la copia que se va a crear de la imagen.
MÉTODOS		
Nombre	Descripción	
+ selec_ruta_img_fnc (obj : Object) : Object	Permite especificar el nombre, ubicación y formato de la copia que se va a crear para una imagen. Emplea el cuadro de diálogo estándar invocado con <i>uiputfile</i> .	

Tabla 15.4. Descripción de la clase de diseño *IDatosNuevImg*

CLASE : usuarios_registrados_GUI		
Refinamiento clase de análisis : IUUsuariosReg		
ATRIBUTOS		
Nombre	Tipo	Descripción
- lst_nomb_ape_usus	list	List box para los nombres y apellidos de los usuarios registrados.
MÉTODOS		
Nombre	Descripción	
+ most_lista_usu_fnc (dat_usus_curs : cursor) : void	Carga el list box con los nombres y apellidos de los usuarios registrados.	

+ lst_nomb_ape_usus_Callback() : Integer	Maneja el evento relacionado a la selección de un usuario en el list box, retorna el índice correspondiente al elemento seleccionado de la lista.
+ btn_salir_Callback() : void	Destruye el objeto.

Tabla 15.5. Descripción de la clase de diseño *usuarios_registrados_GUI*

CLASE : nueva_clave_usu_GUI		
Refinamiento clase de análisis : IUClaveUsuario		
ATRIBUTOS		
Nombre	Tipo	Descripción
- txt_nuev_clav	String	Cuadro de texto para la nueva clave.
- txt_confirmar_clav	String	Cuadro de texto para la confirmación de la nueva clave.
MÉTODOS		
Nombre	Descripción	
+ btn_acept_Callback() : void	Maneja el evento relacionado a la solicitud de actualizar la clave del usuario. Solicita a una instancia de la clase <i>ManejadorActualizacionClaveUsuario</i> la actualización de la clave.	
+ btn_cancel_Callback() : void	Destruye el objeto.	
- valid_nuev_clav_usu_fnc (nuev_clav_i : Integer, confirm_nuev_clav_i : Integer) : Integer	Valida el formato de la clave ingresada.	

Tabla 15.6. Descripción de la clase de diseño *nueva_clave_usu_GUI*

CLASE : reporte_GUI		
Refinamiento clase de análisis : IUReporteUsuarios		
ATRIBUTOS		
Nombre	Tipo	Descripción
- txt_nomb	String	Nombres de los usuarios registrados.
- txt_apel	String	Apellidos de los usuarios registrados.
- txt_clav	String	Claves de los usuarios registrados.

- ax_img_usu	Matriz	Imagen del rostro de cada usuario registrado.
MÉTODOS		
Nombre	Descripción	
+ btn_salir_Callback()	Destruye el objeto.	

Tabla 15.7. Descripción de la clase de diseño *reporte_GUI*

15.2 Identificación de Clases de Diseño – Control

A continuación, en las Tablas 15.8 hasta 15.23, se detallan los atributos y operaciones de las clases de diseño de tipo control.

CLASE : ManejadorErrorLocalizacion		
Refinamiento clase de análisis : ManejadorErrorLocalizacion		
ATRIBUTOS		
Nombre	Tipo	Descripción
- band_rostro_i	Integer	Valor para definir si se produjo o no un error durante la localización de un rostro en una imagen.
- msj_err_txt	String	Mensaje que se presentará al usuario indicando la razón por la que no se localizó un rostro en la imagen.
MÉTODOS		
Nombre	Descripción	
+ gest_msg_fnc (obj : Object, band_err_i : Integer) : Object	Gestiona el valor de error retornado por el subsistema Localizar rostro. Retorna una instancia que contiene el mensaje de error asociado a dicho valor.	

Tabla 15.8. Descripción de la clase de diseño *ManejadorErrorLocalizacion*

CLASE : BuscadorUsuario		
Refinamiento clase de análisis : BuscadorUsuario		
ATRIBUTOS		
Nombre	Tipo	Descripción
- clav_busq_txt	String	Clave de usuario.
MÉTODOS		
Nombre	Descripción	
+ validar_clave_fnc (obj : Object, obj_DB : AccesoBD) : Integer	Realiza la búsqueda de un usuario previamente registrado con una clave específica y retorna un valor asociado al resultado de la búsqueda.	

Tabla 15.9. Descripción de la clase de diseño *BuscadorUsuario*

CLASE : GeneradorImgSintet		
Refinamiento clase de análisis : GeneradorImgSintet		
ATRIBUTOS		
Nombre	Tipo	Descripción
- r1_img	Matriz	Imagen que se crea a partir de la imagen extraída automáticamente por el subsistema Localizar rostro, es una imagen sintética generada empleando dicha imagen.
- r2_img	Matriz	Similar a r1_img.
- r3_img	Matriz	Similar a r1_img.
- r4_img	Matriz	Similar a r1_img.
- r5_img	Matriz	Similar a r1_img.
- r6_img	Matriz	Similar a r1_img.
MÉTODOS		
Nombre	Descripción	
+ calc_img_sint_fnc (obj : Object, img_gris_img : Integer, ancho_i : Integer, alto_i : Integer, fil_img_i : Integer, col_img_i : Integer) : Object	Crea seis imágenes sintéticas utilizando las coordenadas y la imagen proporcionadas como parámetros de entrada.	

Tabla 15.10. Descripción de la clase de diseño *GeneradorImgSintet*

CLASE : Preproceso		
Refinamiento clase de análisis : ManejadorLocalizacionRostro		
ATRIBUTOS		
Nombre	Tipo	Descripción
---	---	---
MÉTODOS		
Nombre	Descripción	
+ elim_ruido_fnc (img_inic_img : mx) : mx	Elimina posible ruido en la imagen original mediante la aplicación del filtro de paso bajo de Wiener.	
+ comp_ilum_fnc (img_rest_img : mx) : mx	Compensa la iluminación de la imagen o mejora su contraste.	

Tabla 15.11. Descripción de la clase de diseño *Preproceso*

CLASE : SegmentacionColor		
Refinamiento clase de análisis : ManejadorLocalizacionRostro		
ATRIBUTOS		
Nombre	Tipo	Descripción
- normR_mx	Matriz	Proporción de rojo en la imagen.
- normG_mx	Matriz	Proporción de verde en la imagen.
- img_mascara_mx	Matriz	Máscara del área del Rostro. Matriz Binaria en la que el valor uno indica posibles áreas de piel.
- rostros_i	Integer	Valor que permite establecer si en la imagen existen o no áreas del color de la piel humana.
MÉTODOS		
Nombre	Descripción	
+ conv_rgb2nrgb_fnc (obj : Object, img_norm_img : mx) : Object	Convierte los valores RGB de los píxeles de la imagen que recibe como parámetro de entrada, al espacio rg- normalizado.	
+ seg_img_fnc (obj : Object, img_gris_img : mx) : Object	Realiza el proceso de segmentación de una imagen basándose en rangos de valores para el color de la piel.	

Tabla 15.12. Descripción de la clase de diseño *SegmentacionColor*

CLASE : Hough		
Refinamiento clase de análisis : ManejadorLocalizacionRostro		
ATRIBUTOS		
Nombre	Tipo	Descripción
- voto_ord_vc	Vector	Vector que contiene el puntaje que cada elipse acumuló en el proceso de votación empleado por la transformada de Hough. Sólo se consideran las elipses con mayor votación.
- posx_vc	Vector	Vector que contiene la posición en el eje x (fila) en la que está ubicado el centro de cada elipse.
- posy_vc	Vector	Vector que contiene la posición en el eje y (columna) en la que está ubicado el centro de cada elipse.
- sejMay_vc	Vector	Vector que contiene el valor del semi-eje mayor de cada elipse.
- sejMen_vc	Vector	Vector que contiene el valor del semi-eje menor de cada elipse.
MÉTODOS		
Nombre	Descripción	
+ hough_fnc (obj : Object, img_piel_mx : mx, img_msc_mx : mx) : Object	Detecta mediante la transformada de Hough todas las posibles elipses que existen en la imagen que recibe como parámetro de entrada.	
+ fus_elip_fnc (obj : Object, fil_i : Integer, col_i : Integer, FIL_i : Integer, COL_i : Integer) : Object	Fusiona las elipses de tamaño similar y centros adyacentes detectadas previamente mediante la transformada de Hough.	

Tabla 15.13. Descripción de la clase de diseño *Hough*

CLASE : DeteccionOjos		
Refinamiento clase de análisis : ManejadorLocalizacionRostro		
ATRIBUTOS		
Nombre	Tipo	Descripción
- ojos_etiq_mx	Matriz	Imagen etiquetada que contiene sólo los candidatos a áreas de los ojos.
- num_parj_i	Integer	Número de parejas posibles a ser áreas de los ojos.

- obj1_vc	Vector	Contiene uno de los elementos que forman las parejas de posibles ojos.
- obj2_vc	Vector	Contiene uno de los elementos que forman las parejas de posibles ojos.
- ang_mx	Matriz	Contiene los ángulos formados por cada pareja de posibles ojos en relación a la horizontal.
MÉTODOS		
Nombre	Descripción	
+ detect_ojos_fnc (obj : Object, candidato_img : mx, mascara_mx : mx, YCbCr_img : mx) : Object	Realiza la detección de posibles candidatos a ojos en la imagen proporcionada como parámetro de entrada y establece la ubicación de éstos.	
- loc_areas_ojos_fnc (img_rostro_img : mx) : mx	Localiza los posibles candidatos a áreas de los ojos mediante la umbralización de la imagen de entrada en los espacios de color HSV y rg-normalizado.	
- elb_mapa_ojos_fnc (YCbCr_img : mx, mascara_mx : mx) : mx	Elabora un mapa de candidatos a áreas de los ojos basándose en el espacio de color YCbCr.	
- anl_geom_sfnc (obj : Object, relac_i : Integer, num_objs_i : Integer, col_i : Integer) : Object	Analiza la disposición geométrica de los candidatos a áreas de los ojos y elimina a aquellos candidatos que no cumplen ciertas características.	

Tabla 15.14. Descripción de la clase de diseño *DeteccionOjos*

CLASE : DeteccionBoca		
Refinamiento clase de análisis : ManejadorLocalizacionRostro		
ATRIBUTOS		
Nombre	Tipo	Descripción
- boca_etiq_mx	Matriz	Imagen etiquetada que contiene sólo los candidatos a área de la boca.
- num_objs_i	Integer	Número de candidatos.
MÉTODOS		
Nombre	Descripción	
+ detect_boca_fnc (obj : Object, YCbCr_img : mx, mascara_mx : mx) : Object ct	Realiza la detección de los posibles candidatos a área de la boca empleando una imagen en el espacio YCbCr que es proporcionada como parámetro de entrada.	

Tabla 15.15. Descripción de la clase de diseño *DeteccionBoca*

CLASE : CaracteristicasFaciales		
Refinamiento clase de análisis : ManejadorLocalizacionRostro		
ATRIBUTOS		
Nombre	Tipo	Descripción
- candidato_st	struct	Estructura que contiene los vértices y la puntuación de los candidatos a rostro.
- num_can_i	Integer	Número de candidatos a rostro que cumplen las condiciones establecidas.
MÉTODOS		
Nombre	Descripción	
+ anl_caract_fnc (obj : Object, prop_ojo1_st : st, prop_ojo2_st : st, prop_boca_st : st, area_mascara_i : Integer) : Object	Busca los tríos formados por una pareja de candidatos a ojos y un candidato a boca que cumplan ciertas condiciones para ser considerados como candidatos a rostros.	
- eval_ang_fnc (triangulo_mx : mx) : vc	Calcula los ángulos formados por las intersecciones de los segmentos trazados a través de los puntos que recibe como parámetro de entrada.	
- eval_area_fnc (triangulo_mx : mx) : Double	Calcula el área del triángulo formado por los puntos que recibe como parámetro de entrada.	
- eval_cand_fnc (triangulo_mx : mx, area_mascara_i : Integer, area_1_db : Double, area_2_db : Double) : Double	Evalúa a los candidatos a rostros y les asigna una puntuación basándose en ciertos aspectos que permiten establecer los mejores candidatos a ser el área del rostro.	

Tabla 15.16. Descripción de la clase de diseño *CaracteristicasFaciales*

CLASE : ExtraccionNormalizacion		
Refinamiento clase de análisis : ManejadorLocalizacionRostro		
ATRIBUTOS		
Nombre	Tipo	Descripción
- img_gris_img : mx	Matriz	Imagen original normalizada en escala de grises y rotada.
- angulo_rot_db	Double	Ángulo que se debe rotar la imagen para normalizar la posición de los ojos.

- X_min_i	Integer	Primera fila en la que está ubicada el área del rostro en relación con la imagen original.
- Y_min_i	Integer	Primera columna en la que está ubicada el área del rostro en relación con la imagen original.
- rostro_img	Matriz	Imagen que contiene sólo el rostro localizado.
- cen1_db	Double	Coordenadas del centroide del área del ojo (centroide aproximado)
- cen2_db	Double	Coordenadas del centroide del área del ojo (centroide aproximado)
- cen3_db	Double	Coordenadas del centroide del área de la boca (centroide aproximado)
- csi_i	Integer	Columna superior izquierda del recuadro que contiene al rostro localizado.
- fsi_i	Integer	Fila superior izquierda del recuadro que contiene al rostro localizado.
- ancho_i	Integer	Ancho del recuadro que contiene al rostro localizado.
- alto_i	Integer	Alto del recuadro que contiene al rostro localizado.
MÉTODOS		
Nombre		Descripción
+ recortar_rostro_fnc (obj : Object, ojos_etiq_img : mx, boca_etiq_img : mx, prop_ojos_st : st, prop_boca_st : st) : Object		Calcula las coordenadas de los centros aproximados para las áreas de los ojos y boca previamente detectadas luego de rotar la imagen original de acuerdo al ángulo obtenido. Finalmente recorta el área del rostro basándose en distancias antropométricas del mismo.
+ imadjust (rostro_img : mx) : mx		Realiza un estiramiento del histograma de una imagen.
+ anl_eigen_fnc (rostro_img : mx, EigenFaces_mx : mx, RostroProm_vc : vc, obj_ImgRec : ImagenReconstruida) : Double		Confirma si la imagen obtenida luego del proceso de localización es un rostro o no, para esto calcula la distancia euclidiana entre el rostro localizado y su proyección en el espacio de rostros.

Tabla 15.17. Descripción de la clase de diseño *ExtraccionNormalizacion*

CLASE : AutenticacionAdmin		
Refinamiento clase de análisis : <i>ManejadorAutenticacion</i>		
ATRIBUTOS		
Nombre	Tipo	Descripción
---	---	---
MÉTODOS		
Nombre	Descripción	
+ IsAdmin() : Boolean	<p>Determina si el usuario actual tiene perfil de administrador del equipo o dominio (de acuerdo a la autenticación Windows). El valor que retorna indica si el usuario es o no administrador: 1 (verdadero) = Si es administrador 0 (falso) = Caso contrario.</p>	

Tabla 15.18. Descripción de la clase de diseño *AutenticacionAdmin*

CLASE : ManejadorCopialmg		
Refinamiento clase de análisis : <i>ManejadorCopiaImg</i>		
ATRIBUTOS		
Nombre	Tipo	Descripción
---	---	---
MÉTODOS		
Nombre	Descripción	
+ guard_copia_img_fnc (rutaG_img_txt : String, img_origen_img : mx, formato_txt : String) : void	<p>Crea un nuevo archivo gráfico a partir de la imagen enviada como parámetro de entrada, empleando la ruta y el formato proporcionados también como parámetros de entrada.</p>	

Tabla 15.19. Descripción de la clase de diseño *ManejadorCopiaImg*

CLASE : CapturadorRostroLocalizado		
Refinamiento clase de análisis : CapturadorRostroLocalizado		
ATRIBUTOS		
Nombre	Tipo	Descripción
- captura_frm	Matriz	Imagen en la que se muestra el rostro localizado.
MÉTODOS		
Nombre	Descripción	
+ capt_rostro_loc_fnc (obj : Object, handles : hdl) : Object	Captura desde la GUI la imagen que muestra el rostro localizado.	

Tabla 15.20. Descripción de la clase de diseño *CapturadorRostroLocalizado*

CLASE : ManejadorListaUsuarios		
Refinamiento clase de análisis : ManejadorListaUsuarios		
ATRIBUTOS		
Nombre	Tipo	Descripción
- num_usus_i	Integer	Número de usuarios registrados.
- dat_usus_curs	cursor	Cursor que contiene los nombres y apellidos de todos los usuarios registrados.
MÉTODOS		
Nombre	Descripción	
+ listar_usus_reg_fnc (obj : Object, obj_BD : AccesoBD) : Object	Obtiene el nombre y apellido de cada usuario registrado.	

Tabla 15.21. Descripción de la clase de diseño *ManejadorListaUsuarios*

CLASE : ManejadorActualizacionClaveUsuario		
Refinamiento clase de análisis : ManejadorActualizacionClaveUsuario		
ATRIBUTOS		
Nombre	Tipo	Descripción
- idx_i	Integer	Índice que identifica al usuario seleccionado en el list box.
- nomb_txt	String	Nombre del usuario seleccionado.

- apel_txt	String	Apellido del usuario seleccionado.
- ant_clav_txt	String	Clave anterior del usuario seleccionado (clave que se desea actualizar).
MÉTODOS		
Nombre	Descripción	
+ act_clav_usu_reg_fnc (obj : Object, nuev_clav_txt : String, obj_BD : AccesoBD) : Integer	Determina la clave del usuario seleccionado y solicita su actualización.	

Tabla 15.22. Descripción de la clase de diseño *ManejadorListaUsuarios*

CLASE : ManejadorReporte		
Refinamiento clase de análisis : ManejadorReporte		
ATRIBUTOS		
Nombre	Tipo	Descripción
- dat_usus_curs	cursor	Cursor que contiene los datos de todos los usuarios registrados (nombre, apellido, género y clave).
- img_usus_curs	cursor	Cursor que contiene una imagen del rostro por cada usuario registrado.
MÉTODOS		
Nombre	Descripción	
+ gen_rep_usus_fnc (obj : Object, obj_BD : AccesoBD, obj_ImgRec : ImagenReconstruida) : Object	Genera el reporte de los usuarios registrados.	

Tabla 15.23. Descripción de la clase de diseño *ManejadorReporte*

REALIZACIÓN DE CASOS DE USO - DISEÑO

16.1 Diagrama de Secuencia – Diseño: Caso de uso Obtener Imagen (*)

- Extensión – No se ha detectado un dispositivo

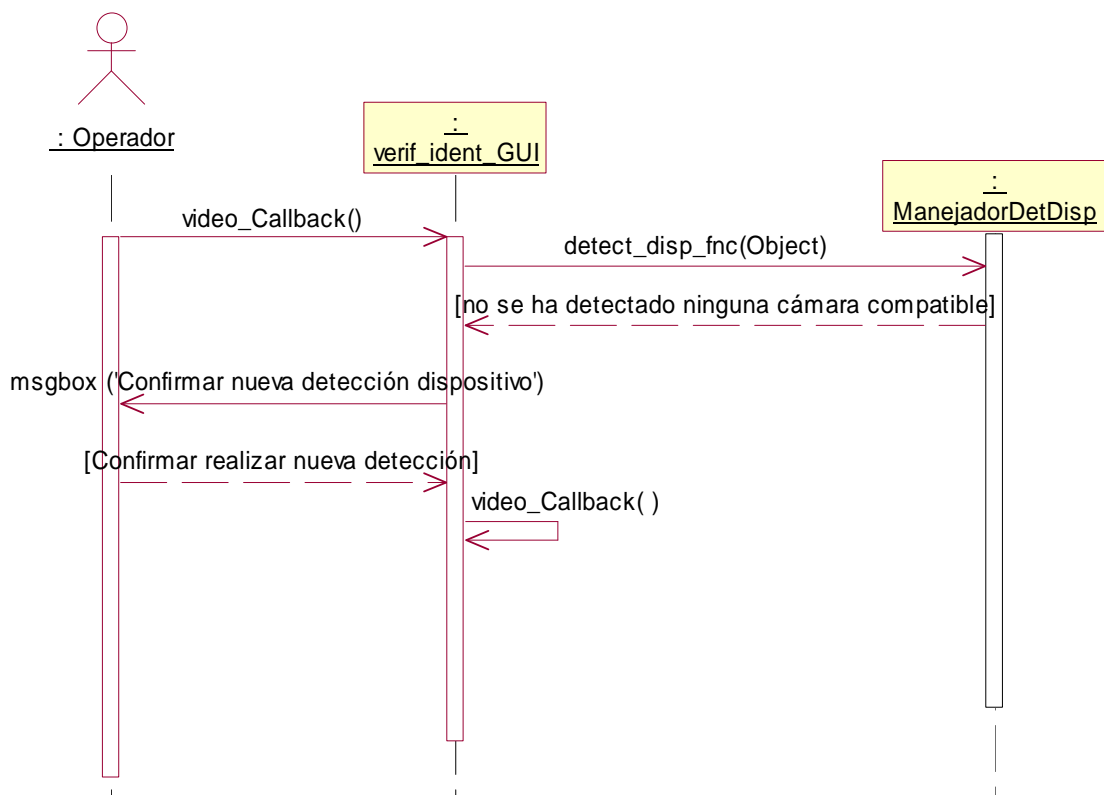


Figura 16.1. Diagrama de Secuencia - Diseño: Caso de uso *Obtener Imagen* – Realización: *Dispositivo de Captura* (Extensión – No se ha detectado un dispositivo)

Descripción del flujo de sucesos-diseño:

La secuencia se inicia cuando el Operador selecciona dentro de `verif_ident_GUI` la opción correspondiente al escenario que se describe (*Obtener Imagen - Video*), esto genera un evento que es manejado por `video_Callback()` que emplea la operación `detect_disp_fnc()` proporcionada

por ManejadorDetDisp para detectar si existe un dispositivo de captura que se encuentre accesible a través de una interfaz genérica de Windows para dispositivos de video (dispositivos que utilicen controladores WDM o VFW). Si no se ha detectado un dispositivo que cumpla estas condiciones se solicita al Operador confirmar si desea ejecutar una nueva detección.

Si el Operador desea que se ejecute una nueva detección, `verif_ident_GUI` genera automáticamente una llamada a `video_Callback()` y el proceso se repite tal como se describe en el párrafo anterior.

(*) Aunque este caso de uso no puede ser considerado concreto en el ámbito de la solución computacional se realiza el diagrama de secuencia y la descripción del flujo de sucesos para un escenario alternativo (*no se ha detectado un dispositivo*), debido a que dicho escenario es resultado de un flujo alternativo de este caso de uso y no de otros casos de uso que lo emplean. Se toma como contexto el caso de uso *Verificar Identidad*.

16.2 Diagrama de Secuencia – Diseño: Caso de uso Localizar Rostro (*)

○ Flujo Básico

Véase Figura 16.2.

Descripción del flujo de sucesos-diseño:

La secuencia se inicia cuando el Operador solicita a un objeto de la clase `verif_ident_GUI` localizar un rostro en la imagen obtenida (desde un dispositivo de almacenamiento o capturada), para esto el Operador presiona el botón *Analizar* que genera un evento que es atendido por `verif_ident_GUI` a través de `btn_analizar_Callback()` que se encarga de solicitar los servicios del subsistema *Localizar Rostro* a través de la interfaz `ManejadorLocalizacionRostro` (`detect_rostro_fnc()`).

A su vez esta interfaz comunica a un objeto de la clase `LocalizacionRostro` la solicitud enviada desde el exterior, invocando la operación `detect_rostro_fnc()` de `LocalizacionRostro`. Esta operación a su vez utiliza la operación `elim_ruido_fnc()` que elimina en lo posible cualquier ruido

presente en la imagen y la operación `comp_ilum_fnc()` que mejora o corrige la iluminación de la imagen, estas dos operaciones son proporcionadas por Preproceso. Una vez que la imagen ha sido preprocesada `detect_rostro_fnc()` invoca la operación `conv_rgb2hrgb_fnc()` para obtener los planos de la imagen en el espacio de color rg-normalizado y luego utiliza la operación `seg_img_fnc()` para localizar posibles áreas del color de la piel basándose en los planos del espacio rg-normalizado, ambas operaciones pertenecen a SegmentacionColor. La imagen segmentada es convertida por `detect_rostro_fnc()` en una imagen en escala de grises de menor tamaño para reducir el costo computacional que involucra la aplicación de la transformada de Hough en una imagen, esta transformada es ejecutada por la operación `hough_fnc()` ofrecida por Hough. Luego de esto se emplea la operación `fus_elip_fnc()` que es proporcionada también por Hough para eliminar posibles elipses que no cuentan con la votación suficiente para ser consideradas como potenciales candidatas a áreas del rostro.

Para cada elemento de Hough se emplea la operación `detect_ojos_fnc()` proporcionada por DeteccionOjos. Esta operación por su parte utiliza las operaciones `loc_areas_ojos_fnc()` que localiza posibles candidatos a ojos empleando los espacios de color HSV y rg-normalizado, la operación `elb_mapa_ojos_fnc()` que se basa en el espacio de color YCbCr para localizar candidatos a ojos, y la operación `anl_geom_sfnc()` que analiza la disposición geométrica de los candidatos a áreas de los ojos que resultan de las dos operaciones anteriores y elimina a aquellos que no cumplen ciertas características. Finalizado esto, se invoca la operación `detect_boca_fnc()` ofrecida por DeteccionBoca, con el objetivo de obtener los posibles candidatos a área de la boca empleando el espacio de color YCbCr.

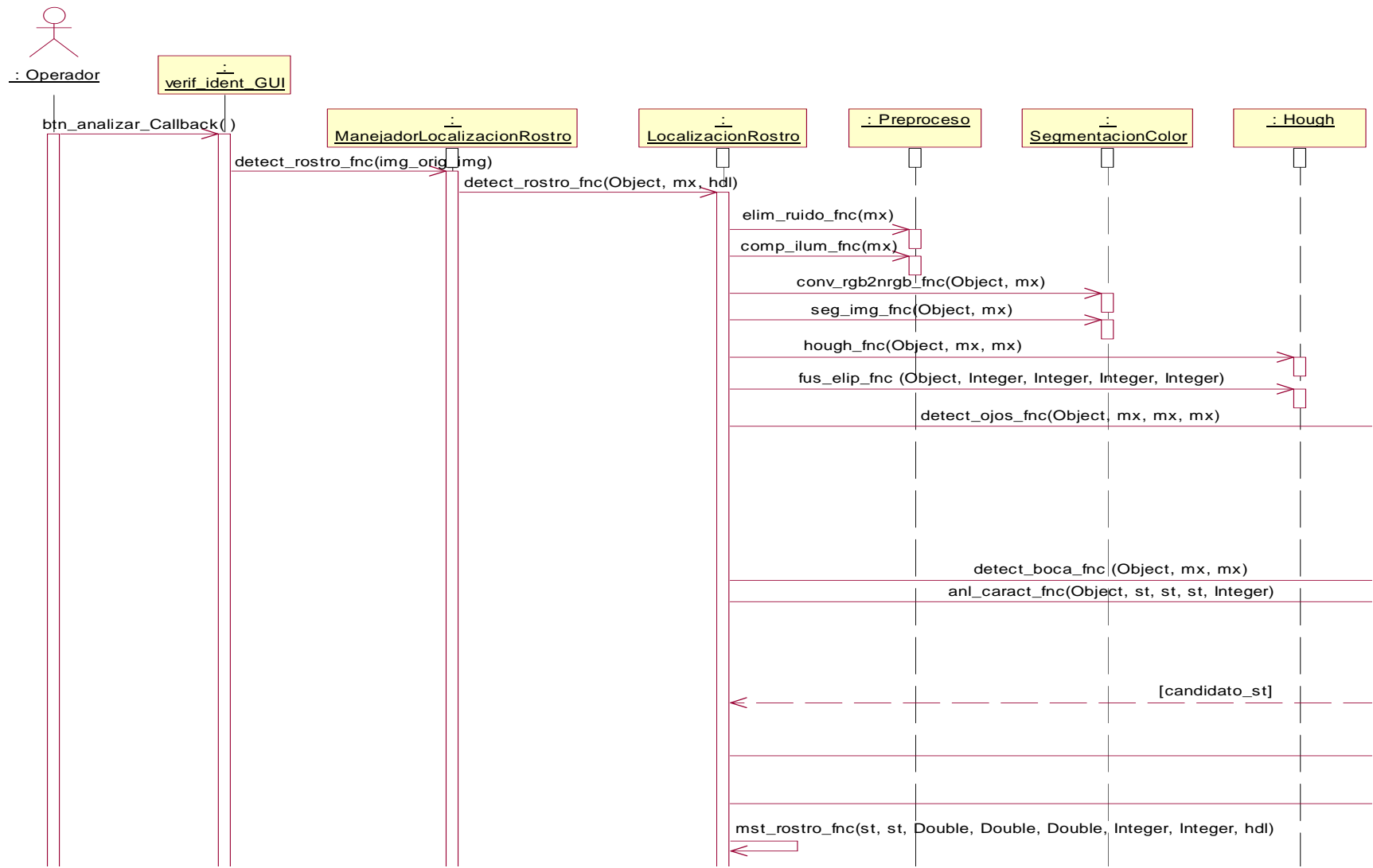
Luego de esto LocalizaciónRostro obtiene de DeteccionOjos y de DeteccionBoca el número de candidatos encontrados y las imágenes etiquetadas que contienen a dichos candidatos, e invoca la operación

anl_caract_fnc() ofrecida por CaracteristicasFaciales, que busca los tríos formados por una pareja de candidatos a ojos y un candidato a boca que cumplan ciertas condiciones para ser considerados como candidatos a rostros. Para esto anl_caract_fnc() utiliza las operaciones eval_ang_fnc() y eval_area_fnc() de CaracteristicasFaciales que permiten determinar si los tríos formados constituyen triángulos semejantes al creado por el área de los ojos y la boca en un rostro humano, luego emplea la operación eval_cand_fnc() proporcionada también por CaracteristicasFaciales y que permite evaluar los candidatos a rostros asignándole a cada uno de éstos una puntuación basada en ciertos aspectos que determinan los mejores candidatos para ser el área del rostro.

LocalizaciónRostro obtiene de CaracteristicasFaciales la estructura que contiene los vértices y la puntuación de los candidatos a rostro, toma la información del candidato con mayor puntuación e invoca la operación recortar_rostro_fnc() de ExtraccionNormalizacion para que calcule las coordenadas aproximadas de las áreas de los ojos y boca luego de rotar la imagen, recorte el área del rostro basándose en distancias antropométricas y realice el estiramiento del histograma de esta nueva imagen (área del rostro) utilizando para esto la operación imadjust().

Esta nueva imagen es proporcionada como parámetro de entrada a la operación anl_eigen_fnc() ofrecida por ExtraccionNormalizacion, la cual calcula la distancia euclidiana entre el rostro localizado (nueva imagen) y su proyección en el espacio de rostros y retorna esta distancia. Si la distancia es menor a un valor umbral, LocalizaciónRostro utiliza la operación mst_rostro_fnc() para mostrar el área del rostro en la imagen inicial y luego desplegarla en la ventana cuyo manejador ha sido enviado como parámetro de entrada a detect_rostro_fnc().

(*) Aunque este caso de uso no puede ser considerado concreto en el ámbito de la solución computacional se realiza el diagrama de secuencia y la descripción del flujo de sucesos del escenario básico tomando como contexto el caso de uso *Verificar Identidad* con el objetivo de entender el funcionamiento del subsistema que lo realiza.



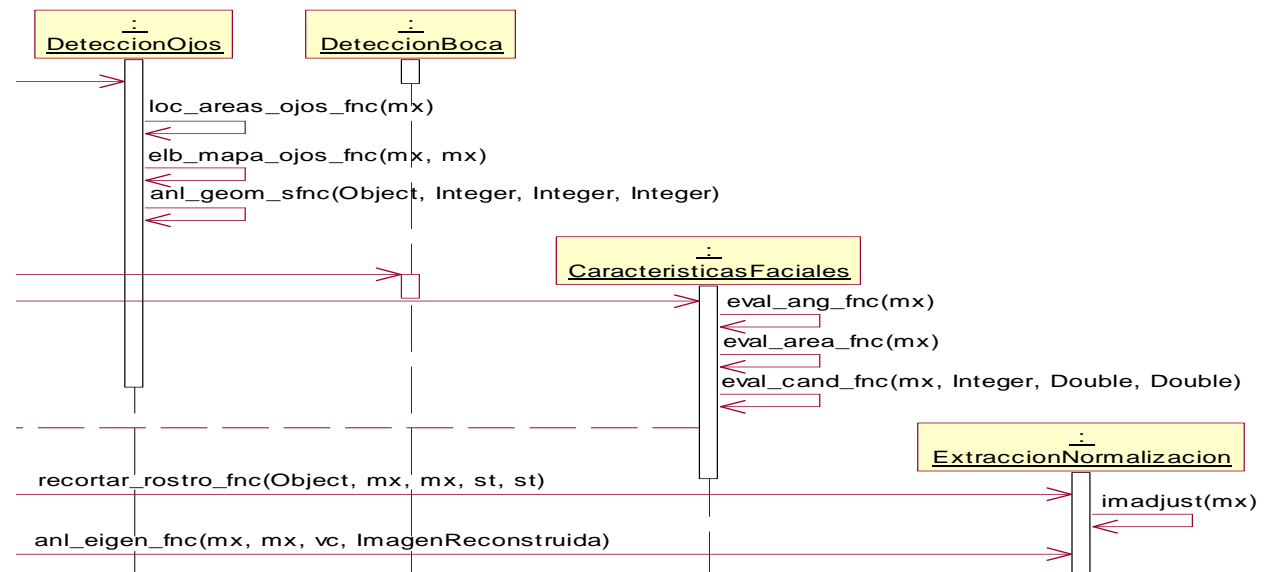


Figura 16.2. Diagrama de Secuencia - Diseño: Caso de uso *Localizar Rostro* (Flujo básico)

16.3 Diagramas de Secuencia – Diseño: Caso de uso Comprobar Perfil para acceso

- Flujo Básico – Caso de uso iniciado por el Administrador

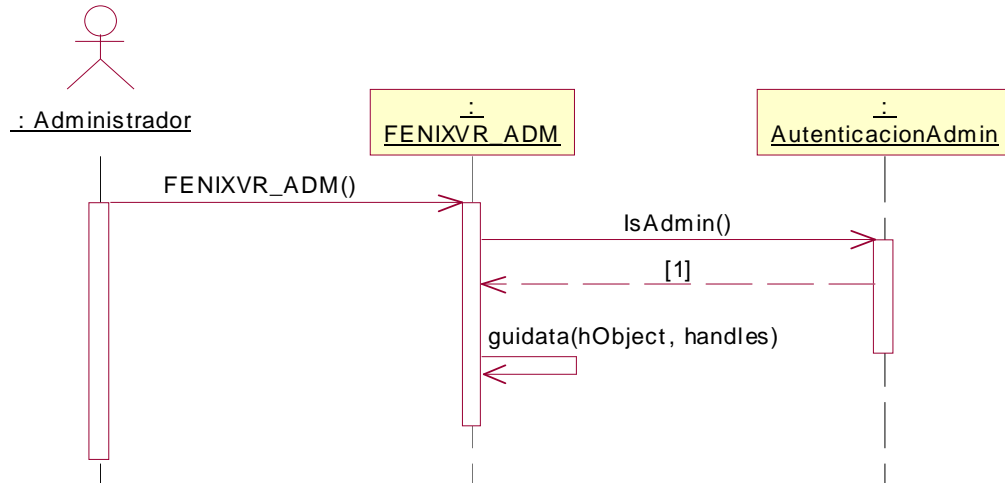


Figura 16.3. Diagrama de Secuencia - Diseño: Caso de uso *Comprobar Perfil para acceso* (Flujo Básico – Caso de uso iniciado por el Administrador)

Descripción del flujo de sucesos-diseño:

La secuencia se inicia cuando el Administrador, al solicitar realizar tareas de administración, activa el objeto de la clase FENIXVR_ADM, que antes de desplegar la interfaz con las opciones de administración solicita a un objeto de la clase AutenticacionAdmin a través de la operación IsAdmin() determinar si el usuario en cuya sesión se está ejecutando la solución computacional tiene perfil de administrador del equipo o dominio, de acuerdo a la autenticación Windows.

Como el escenario que se describe es exitoso, el Administrador tiene el perfil requerido, por lo tanto FENIXVR_ADM presenta la ventana con las tareas de administración disponibles.

- Extensión – El usuario no cuenta con el perfil adecuado

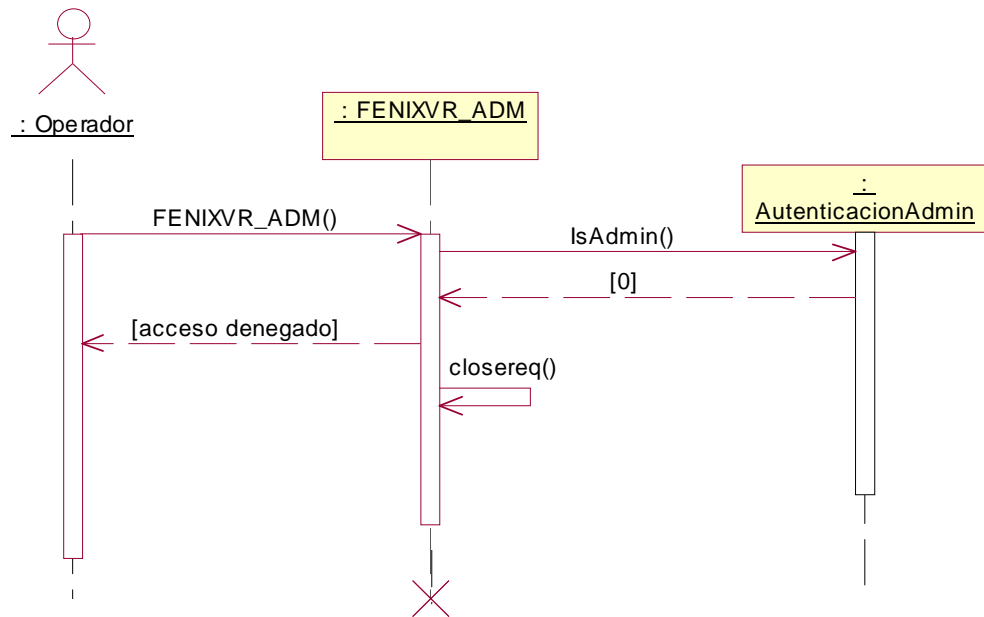


Figura 16.4. Diagrama de Secuencia - Diseño: Caso de uso *Comprobar Perfil para acceso* (Extensión – El usuario no cuenta con el perfil adecuado)

Descripción del flujo de sucesos-diseño:

La secuencia se inicia cuando el Operador, al solicitar realizar tareas de administración, activa el objeto de la clase FENIXVR_ADM, que antes de desplegar la interfaz con las opciones de administración solicita a un objeto de la clase AutenticacionAdmin a través de la operación IsAdmin() determinar si el usuario en cuya sesión se está ejecutando la solución computacional tiene perfil de administrador del equipo o dominio, de acuerdo a la autenticación Windows.

Como el escenario que se describe es una extensión (flujo alternativo), el Operador no tiene el perfil requerido, por lo tanto FENIXVR_ADM presenta un mensaje al Operador informándole que no puede acceder a las operaciones de administración y cierra la ventana que contiene estas opciones antes que se haga visible, utilizando la operación closereq() que es proporcionada por la herramienta de desarrollo.

16.4 Diagrama de Secuencia – Diseño: Caso de uso Guardar Imagen Inicial

○ Flujo Básico

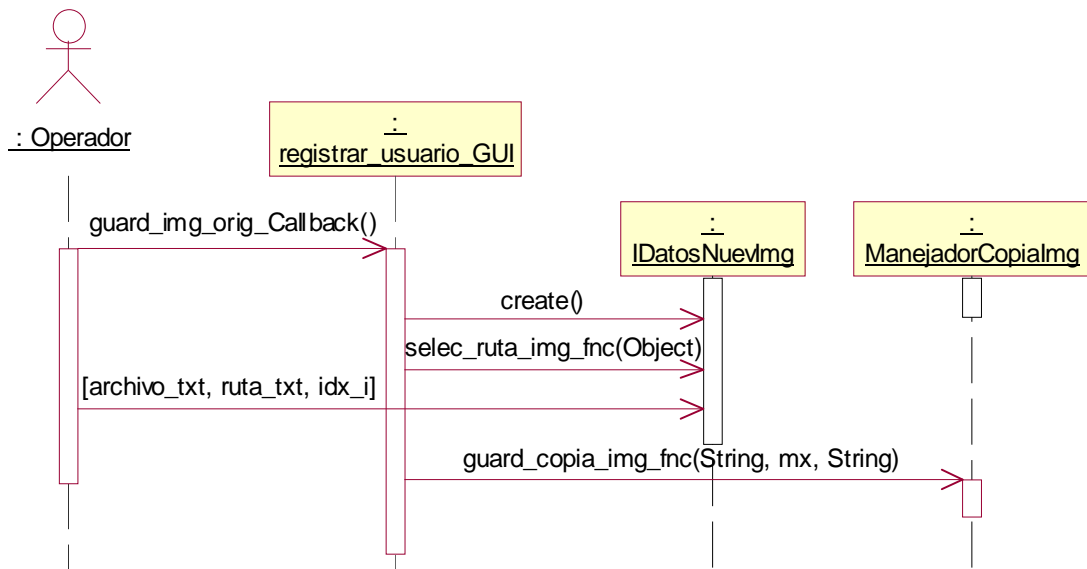


Figura 16.5. Diagrama de Secuencia - Diseño: Caso de uso *Guardar Imagen Inicial* (Flujo Básico)

Descripción del flujo de sucesos-diseño:

(*) El escenario que se describe a continuación corresponde a la solicitud de guardar la imagen que se emplea para registrar un nuevo usuario, el flujo de eventos es el mismo para el caso que se desee almacenar una imagen empleada para verificar una identidad.

La secuencia se inicia cuando el Operador selecciona dentro de registrar_usuario_GUI la opción correspondiente al escenario que se describe (*Guardar Imagen Original*), esto genera un evento que es manejado por guard_img_orig_Callback() que se encarga de solicitar un objeto de la clase IDatosNuevImg e invocar la operación selec_ruta_img_fnc() de este objeto, con el fin de desplegar un cuadro de diálogo estándar que permita al Operador especificar el nombre, ubicación y formato de la copia que se va a crear para la imagen.

El Operador ingresa el nombre, ubicación y formato para la copia de la imagen y presiona el botón *Guardar* del cuadro de diálogo desplegado, esto invoca a la operación `guard_copia_img_fnc()` ofrecida por `ManejadorCopialmg`, que se encarga de crear un nuevo archivo gráfico a partir de la imagen original.

16.5 Diagrama de Secuencia – Diseño: Caso de uso *Guardar Imagen Rostro Localizado*

- *Flujo Básico*

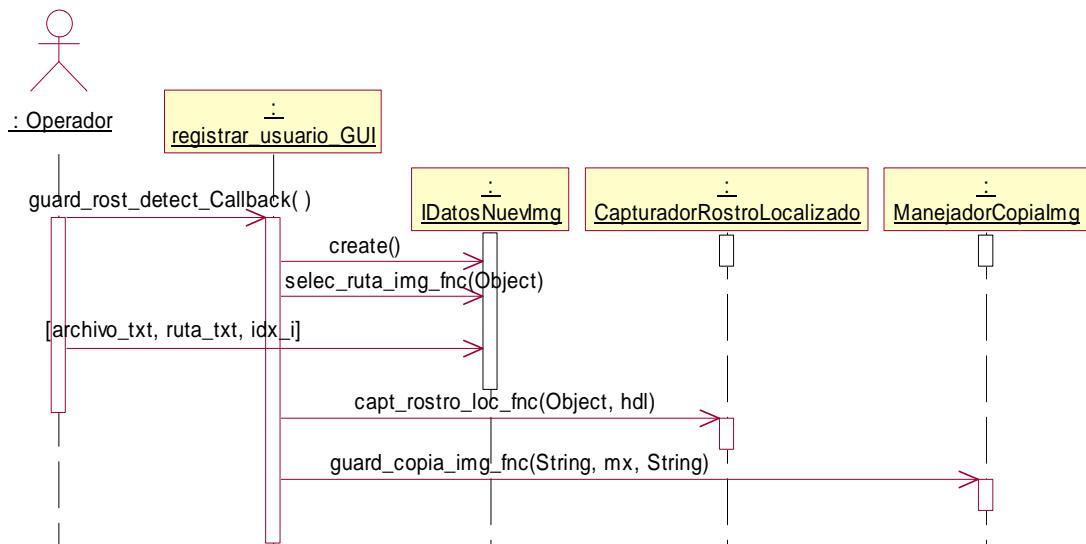


Figura 16.6. Diagrama de Secuencia - Diseño: Caso de uso *Guardar Imagen Rostro Localizado* (Flujo Básico)

Descripción del flujo de sucesos-diseño:

(*) El escenario que se describe a continuación corresponde a la solicitud de guardar la imagen que contiene el rostro detectado cuando se está registrando a un nuevo usuario, el flujo de eventos es el mismo para el caso que se desee almacenar una imagen empleada para verificar una identidad.

La secuencia se inicia cuando el Operador selecciona dentro de `registrar_usuario_GUI` la opción correspondiente al escenario que se

describe (*Guardar Rostro Detectado*), esto genera un evento que es manejado por `guard_rost_detect_Callback()` que se encarga de solicitar un objeto de la clase `IdatosNuevImg` e invocar la operación `selec_ruta_img_fnc()` de este objeto, con el fin de desplegar un cuadro de diálogo estándar que permita al Operador especificar el nombre, ubicación y formato de la copia que se va a crear para la imagen.

El Operador ingresa el nombre, ubicación y formato para la copia de la imagen y presiona el botón *Guardar* del cuadro de diálogo desplegado, esto invoca a la operación `capt_rostro_loc_fnc()` ofrecida por `CapturadorRostroLocalizado`, con el fin de capturar desde la ventana de registro la imagen que muestra el rostro localizado.

Luego de esto se utiliza la operación `guard_copia_img_fnc()` proporcionada por `ManejadorCopialmg`, que se encarga de crear un nuevo archivo gráfico a partir de la imagen que recibe como parámetro, en este caso, la imagen capturada.

16.6 Diagrama de Secuencia – Diseño: Caso de uso Actualizar Claves de Usuario

- *Flujo Básico*

Véase Figura 16.7.

Descripción del flujo de sucesos-diseño:

La secuencia se inicia cuando el Administrador solicita a un objeto de la clase `FENIXVR_ADM` actualizar la clave de un usuario al presionar el botón *Actualizar Clave*. Esto genera un evento que es atendido por `FENIXVR_ADM` a través de `btn_act_clav_Callback()` que activa y despliega la ventana de usuarios registrados (`usuarios_registrados_GUI`), para que así el Administrador pueda seleccionar el usuario cuya clave desea actualizar.

Antes que la ventana `usuarios_registrados_GUI` sea visible, solicita un objeto de la clase `AccesoBD` e invoca la operación `listar_usus_reg_fnc()` ofrecida por `ManejadorListaUsuarios` para obtener la lista de los usuarios registrados. Esta operación a su vez utiliza la operación `consultar_usuarios_sp()` de `AccesoBD` que consulta en la tabla `Usuario` el nombre y apellido de todos los usuarios registrados.

Finalmente `usuarios_registrados_GUI` obtiene de `ManejadorListaUsuarios` el cursor que contiene los nombres y apellidos de los usuarios para presentarlos en pantalla mediante la operación `most_lista_usu_fnc()`.

Una vez que `usuarios_registrados_GUI` es visible, el Administrador selecciona del list box el usuario cuya clave desea actualizar. Esto genera un evento que es atendido por `usuarios_registrados_GUI` a través de `lst_nomb_ape_usus_Callback()` que activa y despliega la ventana para establecer la nueva clave del usuario (`nueva_clave_usu_GUI`).

Desplegada la ventana `nueva_clave_usu_GUI`, el Administrador ingresa la nueva clave y la confirma, luego solicita realizar la actualización al presionar el botón *Aceptar* que genera un evento que es atendido por `nueva_clave_usu_GUI` a través de `btn_acept_Callback()`, que a su vez utiliza las operaciones `valid_nuev_clav_usu_fnc()` de `nueva_clave_usu_GUI` para validar el formato de la clave ingresada y la operación `act_clav_usu_reg_fnc()` que es proporcionada por `ManejadorActualizacionClaveUsuario` para actualizar la clave del usuario. Esta operación a su vez utiliza a `actualizar_clav_sp()` que es proporcionada por `AccesoBD` para que acceda a la tabla `Usuario` y modifique el campo *clave* que corresponda al usuario seleccionado.

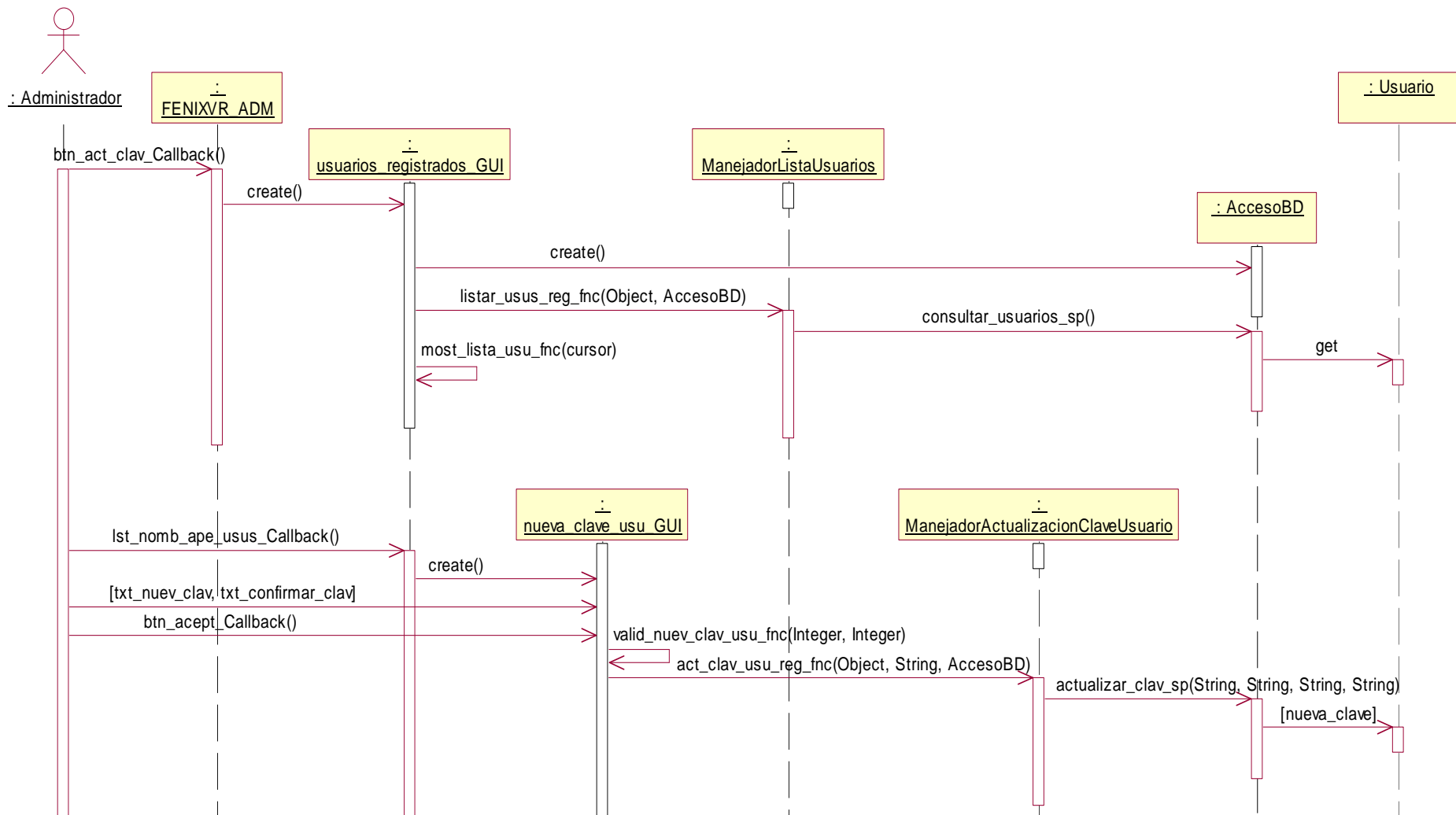


Figura 16.7. Diagrama de Secuencia - Diseño: Caso de uso *Actualizar Claves de Usuario* (Flujo Básico)

16.7 Diagrama de Secuencia – Diseño: Caso de uso Generar Reporte Usuarios Registrados

- *Flujo Básico*

Véase Figura 16.8.

Descripción del flujo de sucesos-diseño:

La secuencia se inicia cuando el Administrador solicita a un objeto de la clase FENIXVR_ADM presentar el reporte de usuarios registrados al presionar el botón *Ver Reporte*. Esto genera un evento que es atendido por FENIXVR_ADM a través de `btn_gen_rep_Callback()` que solicita un objeto de la clase AccesoBD y un objeto de la clase ImagenReconstruida, e invoca la operación `gen_rep_usus_fnc()` que es proporcionada por ManejadorReporte para obtener los datos necesarios para generar el reporte solicitado.

Esta operación a su vez utiliza la operación `reporte_sp()` de AccesoBD, que obtiene el nombre, apellido, clave, género y una imagen por cada usuario registrado, también emplea la operación `reconst_img_fnc()` de ImagenReconstruida para transformar los objetos BLOB almacenados, que corresponden a las imágenes de los usuarios, en estructuras bidimensionales. Finalmente despliega la ventana de reporte que contiene los datos de los usuarios registrados (`reporte_GUI`).

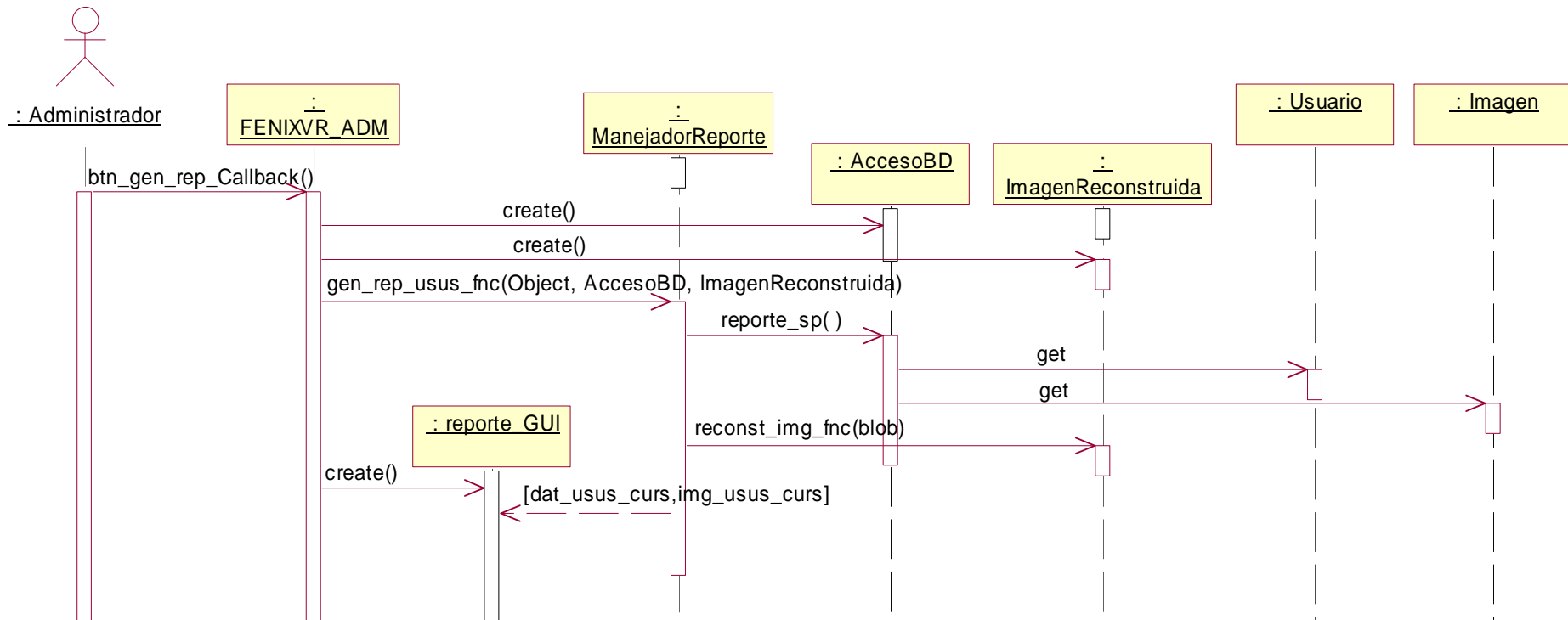


Figura 16.8. Diagrama de Secuencia - Diseño: Caso de uso *Generar Reporte Usuarios Registrados* (Flujo Básico)

Anexo 17

DESCRIPCIÓN DE LAS CARACTERÍSTICAS GENERALES DE LAS GUI

A continuación se realiza una breve descripción de las características de cada interfaz gráfica (GUI) que forma parte de la solución computacional. Las pantallas se muestran en el manual de usuario.

- *Presentación*

Nombre:	splash_GUI
Descripción:	Es la ventana de presentación de FENIXVR.

Tabla 17.1. Descripción de la pantalla de presentación

- *Menú Principal – Operador*

Nombre:	FENIXVR	
Descripción:	Menú principal con las opciones disponibles para cualquier usuario de la solución computacional.	
BOTONES		
Nombre	Activo	Con clic
<i>Registrar Usuario</i>	Siempre	btn_reg_usu_Callback() activa y despliega la ventana de registro de usuarios.
<i>Entrenar Red Neuronal</i>	Siempre	btn_ent_red_Callback() verifica que existan usuarios registrados e invoca la operación ent_red_fnc().
<i>Verificar Identidad</i>	Siempre	btn_verif_ident_Callback() activa y despliega la ventana de verificación de identidad.
<i>Preferencias</i>	Siempre	btn_preferencias_Callback() activa y despliega la ventana de preferencias.
<i>Salir</i>	Siempre	Descarga GUI / finaliza ejecución de la solución computacional.

Tabla 17.2. Descripción del menú principal (opciones Operador)

- *Registrar Usuario*

Nombre:	registrar_usuario_GUI	
Descripción:	En esta ventana el Operador ingresa los datos e imágenes del nuevo usuario para registrarlo.	
BOTONES		
Nombre	Activo	Con clic
<i>Analizar</i>	Luego que el Operador ha obtenido una imagen (desde cualquier fuente) y ha sido desplegada en pantalla.	btn_analizar_Callback()
<i>Registrar Usuario</i>	Luego que se agrega la primera imagen del rostro del nuevo usuario al conjunto de imágenes para el registro de éste.	btn_reg_usu_Callback()
<i>Salir</i>	Siempre	Descarga esta ventana y retorna al menú principal.

Tabla 17.3. Descripción de la pantalla Registrar Usuario

- *Entrenar Red Neuronal*

Esta opción no posee una interfaz gráfica propia. Una vez que el Operador ha seleccionado *Entrenar Red Neuronal* en el menú principal una ventana emergente confirma si se desea iniciar el proceso de entrenamiento. Si el Operador presiona el botón *Si* en dicha ventana, automáticamente se ejecutan las tareas inmersas en el entrenamiento de la red neuronal.

- *Verificar Identidad*

Nombre:	verif_ident_GUI	
Descripción:	En esta ventana el Operador ingresa una imagen que contiene el rostro de un usuario y una clave, y solicita verificar la identidad de dicho usuario.	
BOTONES		
Nombre	Activo	Con clic
<i>Analizar</i>	Luego que el Operador ha obtenido una imagen	btn_analizar_Callback()

	y ha sido desplegada.	
<i>Verificar identidad</i>	Luego que el Operador confirma que desea usar la imagen del rostro para verificar la identidad.	btn_verif_Callback()
<i>Salir</i>	Siempre	Descarga esta ventana y retorna al menú principal.

Tabla 17.4. Descripción de la pantalla Verificar Identidad

- *Información usuario autenticado*

Nombre:	most_img_usu_GUI	
Descripción:	Muestra todos los datos e imágenes almacenadas del usuario cuya identidad ha sido verificada.	
BOTONES		
Nombre	Activo	Con clic
<i>Salir</i>	Siempre	Descarga esta ventana y retorna a verif_ident_GUI

Tabla 17.5. Descripción de la pantalla Información Usuario

- *Preferencias*

Nombre:	preferencias_GUI	
Descripción:	En esta ventana se pueden modificar dos de los parámetros usados para la creación y entrenamiento de una red neuronal, también se puede especificar la ubicación de los archivos que se requieren para el proceso de detección de un rostro.	
BOTONES		
Nombre	Activo	Con clic
<i>Buscar</i>	Siempre	Despliega un cuadro de diálogo estándar para buscar el directorio en que se encuentran los archivos usados para la detección de rostros.
<i>Salir</i>	Siempre	btn_salir_Callback() y retorna al menú principal.

Tabla 17.6. Descripción de la pantalla Preferencias

- *Menú Principal – Administrador*

Nombre:	FENIXVR_ADM	
Descripción:	Ventana principal que contiene las tareas de administración existentes.	
BOTONES		
Nombre	Activo	Con clic
<i>Actualizar Clave</i>	Siempre	btn_act_clav_Callback() que activa y despliega la ventana de usuarios registrados.
<i>Ver Reporte</i>	Siempre	btn_gen_rep_Callback()
<i>Salir</i>	Siempre	Descarga GUI / finaliza ejecución de la solución computacional.

Tabla 17.7. Descripción de la pantalla Menú de Administración

- *Usuarios Registrados*

Nombre:	usuarios_registrados_GUI	
Descripción:	En esta pantalla el Administrador selecciona de una lista donde se encuentran todos los usuarios registrados, el usuario cuya clave desea actualizar.	
BOTONES		
Nombre	Activo	Con clic
<i>Salir</i>	Siempre	Descarga GUI / retorna a FENIXVR_ADM.

Tabla 17.8. Descripción de la pantalla Usuarios Registrados

- *Cambiar clave*

Nombre:	nueva_clave_usu_GUI	
Descripción:	En esta ventana el Administrador puede cambiar la clave del usuario seleccionado en usuarios_registrados_GUI, ingresando una clave nueva y confirmándola.	

BOTONES		
Nombre	Activo	Con clic
<i>Aceptar</i>	Siempre	BTN_ACEPT_CALLBACK()
<i>Cancelar</i>	Siempre	Descarga GUI / retorna a usuarios_registrados_GUI sin actualizar la clave del usuario.

Tabla 17.9. Descripción de la pantalla Establecer nueva clave

- *Reporte*

La Figura 17.1 muestra el diseño del formato para el reporte de usuarios registrados.

REPORTE DE USUARIOS REGISTRADOS				
IMAGEN	NOMBRE	APELLIDO	CLAVE	GÉNERO
				
				

Figura 17.1. Formato del reporte Usuarios Registrados

Anexo 18

DETALLE DE LOS SUBSISTEMAS DE DISEÑO

A continuación se detallan los subsistemas identificados durante la etapa de diseño de la solución computacional.

- *Subsistema GUIs Acceso*
 - a. *Clases*
 - FENIXVR
 - FENIXVR_ADM

- *Subsistema AutenticacionAdmin*
 - a. *Clases*
 - AutenticacionAdmin

 - b. *Interfaces*
 - Perfil (IsAdmin ())

- *Subsistema Registrar Usuario*
 - a. *Clases*
 - registrar_usuario_GUI
 - ManejadorRegistro
 - GeneradorImgSintet

- *Subsistema Verificar Identidad*
 - a. *Clases*
 - verif_ident_GUI
 - most_img_usu_GUI
 - ManejadorVerificacion

- PatronImagen
- ManejadorPresentacionInfUsu

- *Subsistema Entrenar Red*
 - a. *Clases*
 - ManejadorEntrenamientoRed

- *Subsistema Actualizar Parámetros*
 - a. *Clases*
 - preferencias_GUI
 - ManejadorParametros

- *Subsistema Actualizar clave usuario*
 - a. *Clases*
 - usuarios_registrados_GUI
 - nueva_clave_usu_GUI
 - ManejadorListaUsuarios
 - ManejadorActualizacionClaveUsuario

- *Subsistema Generar Reporte*
 - a. *Clases*
 - reporte_GUI
 - ManejadorReporte

- *Subsistema Obtener Imagen*
 - a. *Clases*
 - ICaptura
 - ManejadorDetDisp
 - IImagenExistente

 - b. *Interfaces*
 - Captura (detect_disp_fnc (), conect_disp_fnc(), adq_img_fnc())
 - Seleccion (selec_img_fnc())

- *Subsistema Localizar Rostro*
 - a. *Clases*
 - LocalizacionRostro
 - Preproceso
 - SegmentacionColor
 - Hough
 - DeteccionOjos
 - DeteccionBoca
 - CaracteristicasFaciales
 - ExtraccionNormalizacion
 - b. *Interfaces*
 - ManejadorLocalizacionRostro (detect_rostro_fnc())
- *Subsistema Guardar Imagen*
 - a. *Clases*
 - IDatosNuevImg
 - ManejadorCopialmg
 - CapturadorRostroLocalizado
 - b. *Interfaces*
 - Guardar (select_ruta_img_fnc(), guard_copia_img_fnc (),
capt_rostro_loc_fnc ())
- *Soporte (paquete global)*
 - a. *Clases*
 - AccesoBD
 - ManejadorErrorLocalizacion
 - ImagenReconstruida

Anexo 19

DISEÑO LÓGICO Y FÍSICO DE DATOS

Las Figuras 19.1 y 19.2 muestran el diseño de la base de datos empleada dentro de la solución computacional.

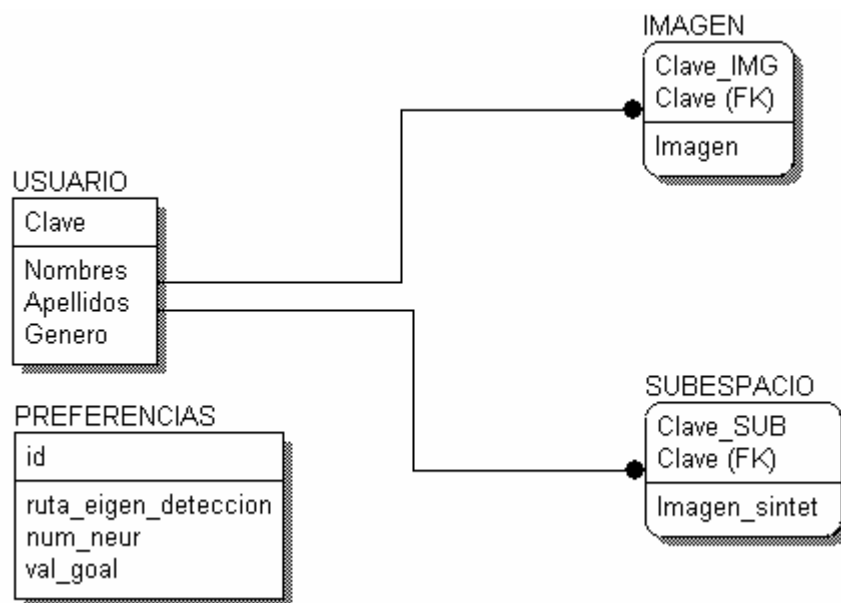


Figura 19.1. Diseño Lógico

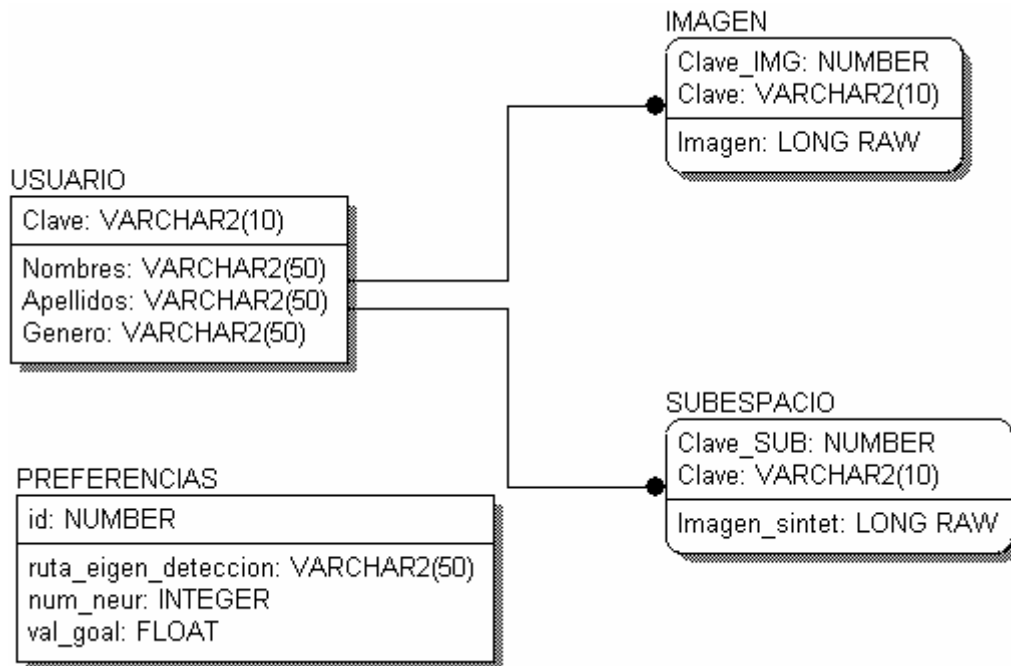


Figura 19.2. Diseño Físico

Cada una de las tablas del diagrama entidad-relación (Figura 19.2) se detalla en función de sus atributos y tipos de datos físicos en el siguiente cuadro:

DESCRIPCIÓN DE ATRIBUTOS				
Entidad	Atributo	Tipo de Dato	Requerido	Descripción
USUARIO	*Clave	Texto (10)	NN	Clave del usuario
USUARIO	Nombres	Texto (50)	NN	Nombres del usuario
USUARIO	Apellidos	Texto (50)	NN	Apellidos del usuario
USUARIO	Genero	Texto (50)	NN	Género
IMAGEN	*Clave_IMG	Autonumérico	NN	Clave principal de la entidad IMAGEN
IMAGEN	Imagen	Objeto OLE	NN	Imagen del rostro

IMAGEN	Clave_USU	Texto (10)	NN	Clave foránea (USUARIO)
SUBESPACIO	*Clave_SUB	Autonumérico	NN	Clave principal de la entidad SUBESPACIO
SUBESPACIO	Imagen_sintet	Objeto OLE	NN	Imagen sintética
SUBESPACIO	Clave_USU	Texto (10)	NN	Clave foránea (USUARIO)
PREFERENCIAS	*id	Numérico (entero)	NN	Clave principal de la entidad PREFERENCIAS
PREFERENCIAS	ruta_eigen_de teccion	Texto (50)	N	Ubicación de los archivos requeridos para la detección de rostros
PREFERENCIAS	num_neur	Numérico (entero)	NN	Número de neuronas añadidas por desplazamiento. Valor por defecto 2.
PREFERENCIAS	val_goal	Double	NN	Error mínimo cuadrático estándar esperado. Valor por defecto 0.8

Tabla 19.1. Descripción de las tablas del diagrama entidad-relación

Anexo 20

IDENTIFICACIÓN DE COMPONENTES IMPLEMENTACIÓN

En las Figuras 20.1 hasta 20.38 se muestra la implementación de los componentes a través de las clases del modelo de diseño.

20.1 SUBSISTEMA DE IMPLEMENTACIÓN - GUI ACCESO

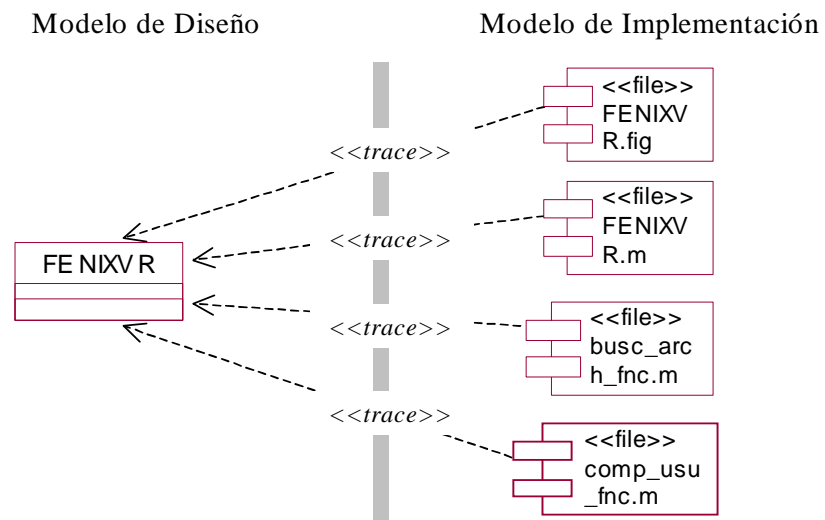


Figura 20.1. Trazas de la clase de diseño FENIXVR

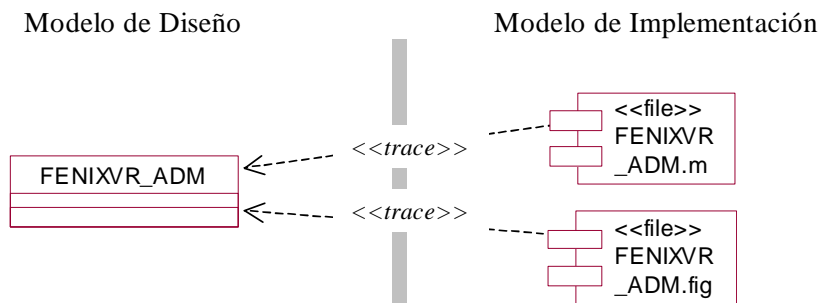


Figura 20.2. Trazas de la clase de diseño FENIXVR_ADM

20.2 SUBSISTEMA DE IMPLEMENTACIÓN – AUTENTICACIONADMIN

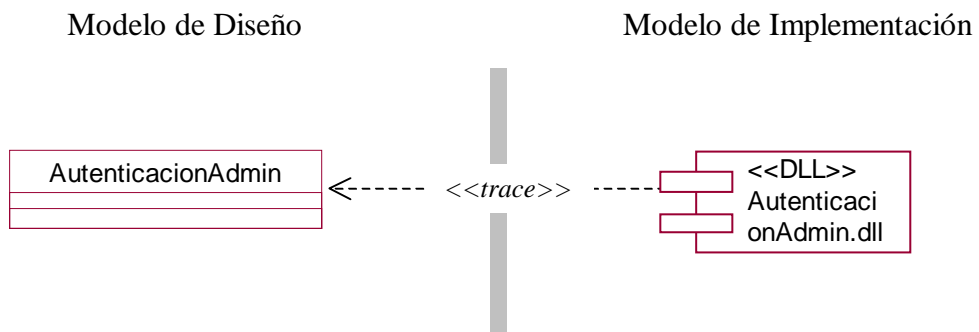


Figura 20.3. Trazas de la clase de diseño AutenticacionAdmin

20.3 SUBSISTEMA DE IMPLEMENTACIÓN - REGISTRAR USUARIO

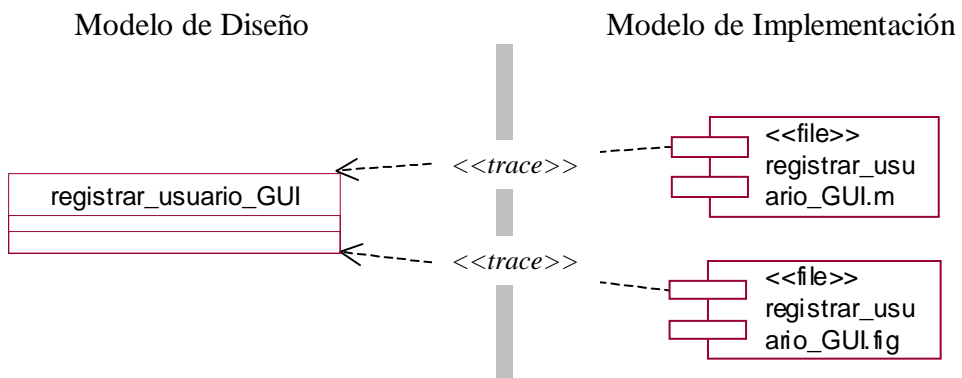


Figura 20.4. Trazas de la clase de diseño registrar_usuario_GUI

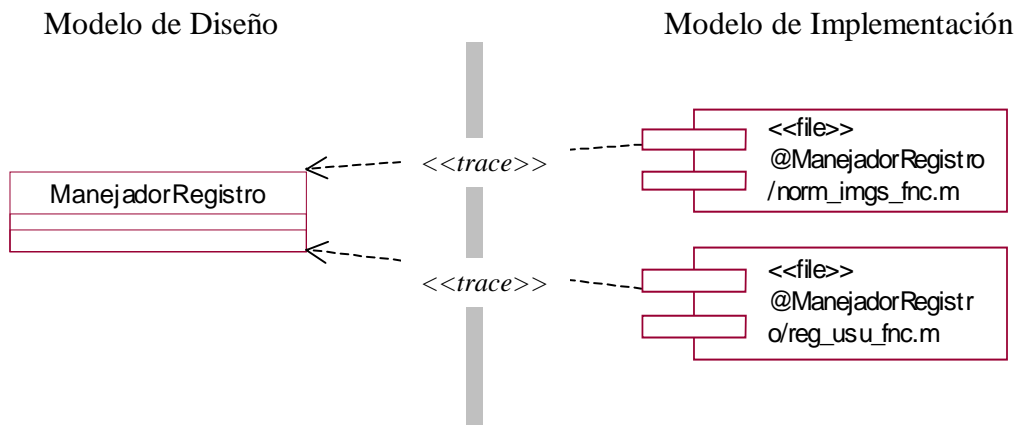


Figura 20.5. Trazas de la clase de diseño ManejadorRegistro

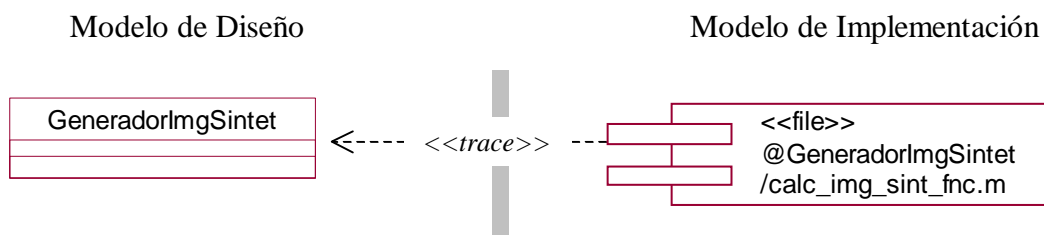


Figura 20.6. Trazas de la clase de diseño GeneradorImgSintet

20.4 SUBSISTEMA DE IMPLEMENTACIÓN – VERIFICAR IDENTIDAD

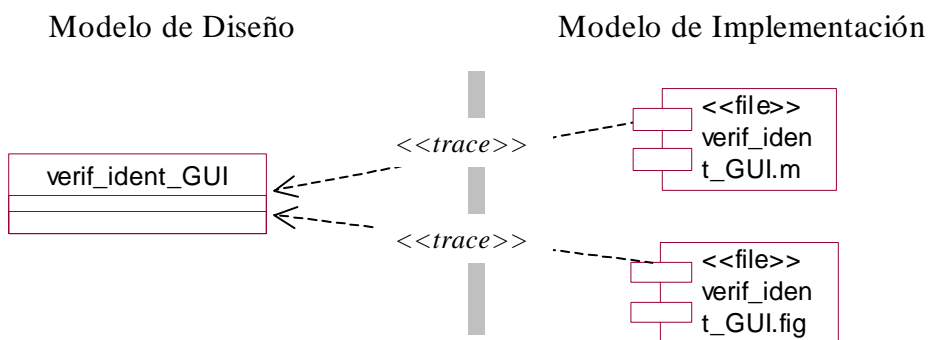


Figura 20.7. Trazas de la clase de diseño verif_ident_GUI

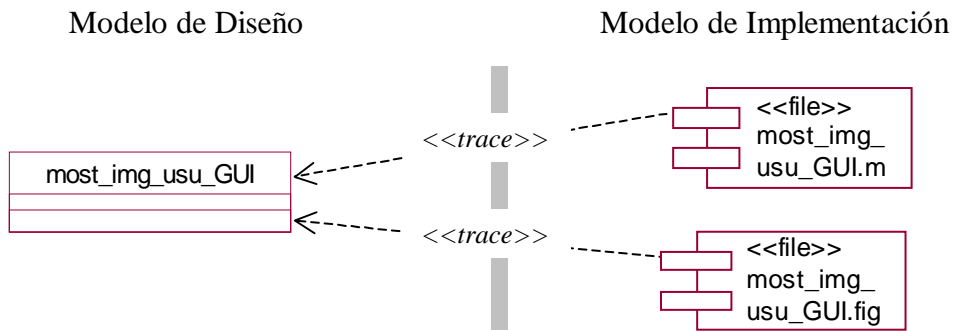


Figura 20.8. Trazas de la clase de diseño `most_img_usu_GUI`

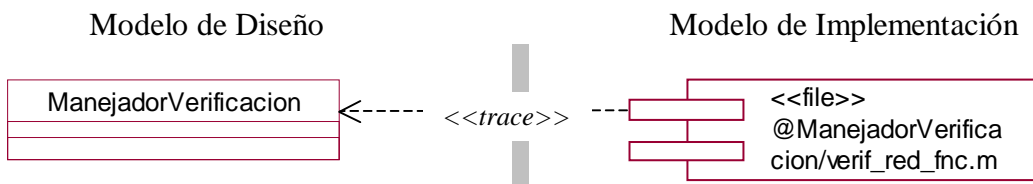


Figura 20.9. Trazas de la clase de diseño `ManejadorVerificacion`

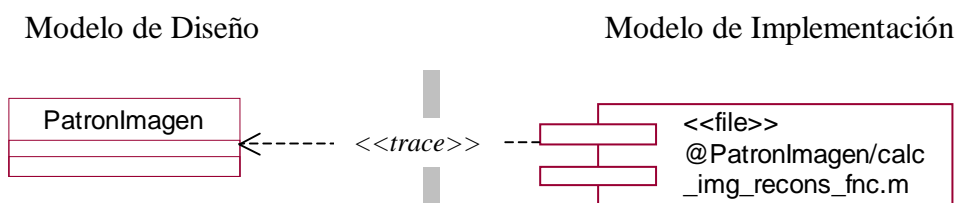


Figura 20.10. Trazas de la clase de diseño `PatronImagen`

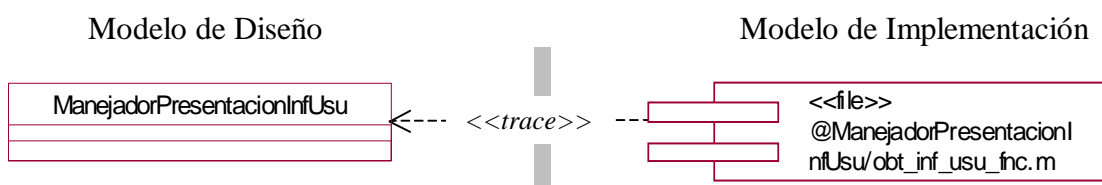


Figura 20.11. Trazas de la clase de diseño `ManejadorPresentacionInfUsu`

20.5 SUBSISTEMA DE IMPLEMENTACIÓN – ENTRENAR RED

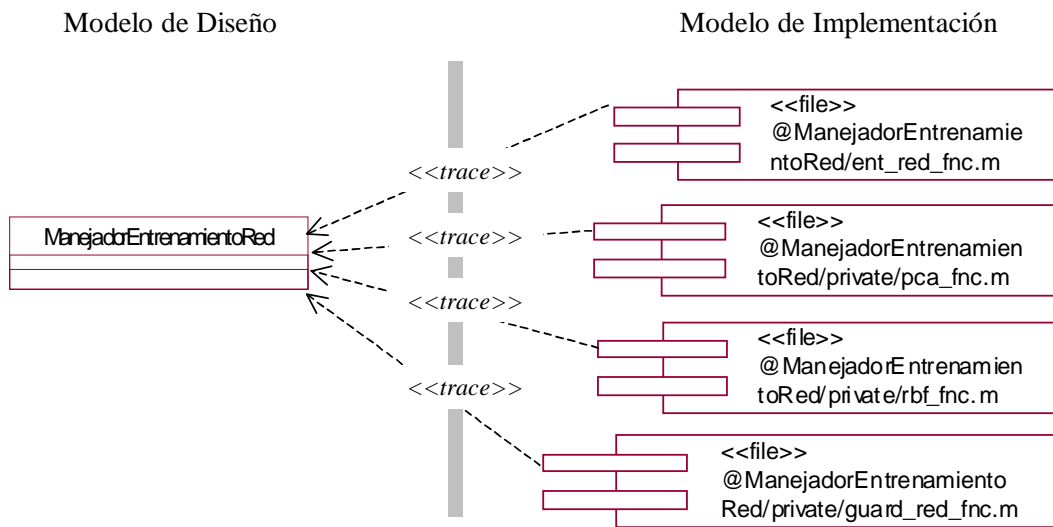


Figura 20.12. Trazas de la clase de diseño ManejadorEntrenamientoRed

20.6 SUBSISTEMA DE IMPLEMENTACIÓN – ACTUALIZAR PARÁMETROS

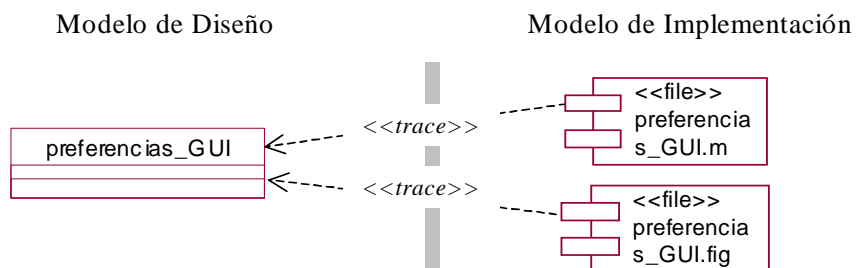


Figura 20.13. Trazas de la clase de diseño preferencias_GUI

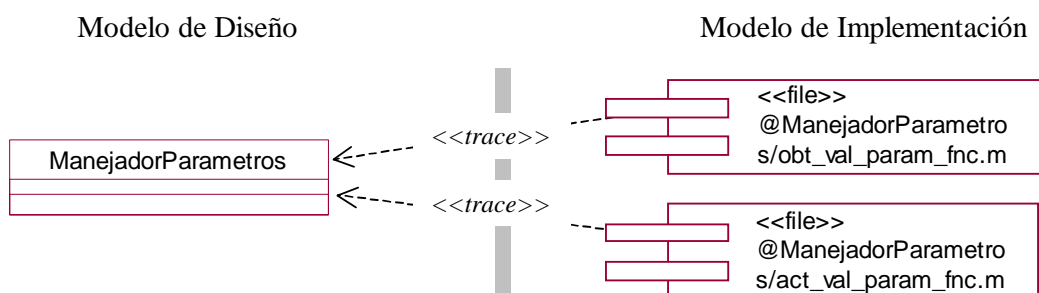


Figura 20.14. Trazas de la clase de diseño ManejadorParametros

20.7 SUBSISTEMA DE IMPLEMENTACIÓN – ACTUALIZAR CLAVE USUARIO

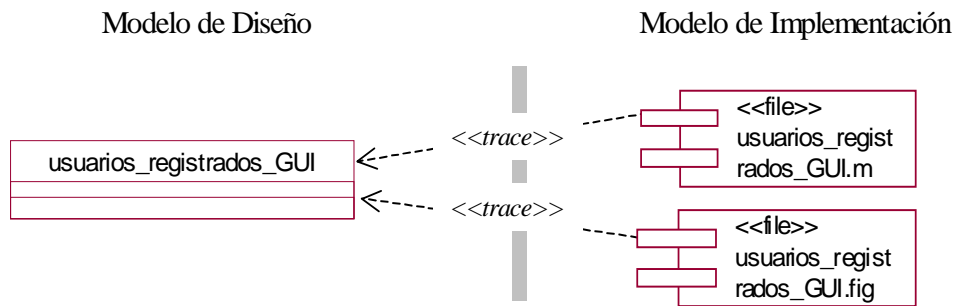


Figura 20.15. Trazas de la clase de diseño `usuarios_registrados_GUI`

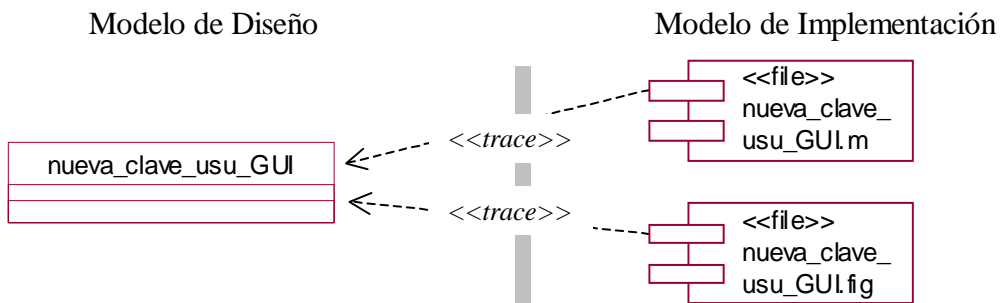


Figura 20.16. Trazas de la clase de diseño `nueva_clave_usu_GUI`

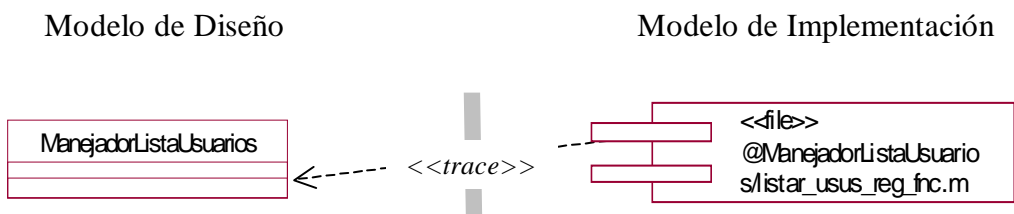


Figura 20.17. Trazas de la clase de diseño `ManejadorListaUsuarios`

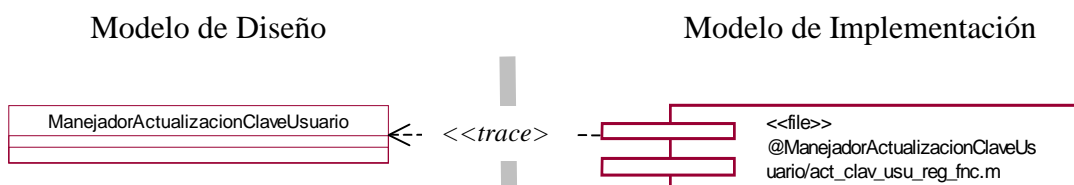


Figura 20.18. Trazas de la clase de diseño `ManejadorActualizacionClaveUsuario`

20.8 SUBSISTEMA DE IMPLEMENTACIÓN – GENERAR REPORTE

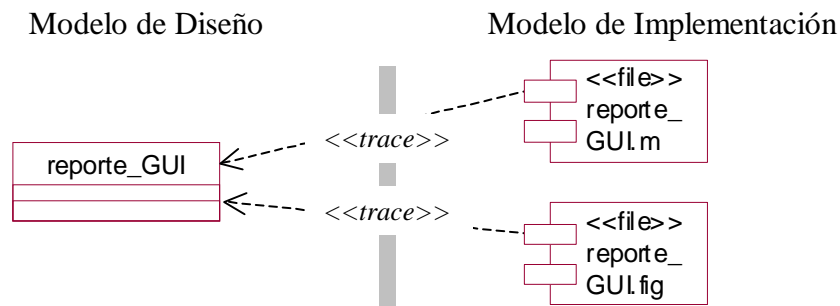


Figura 20.19. Trazas de la clase de diseño `reporte_GUI`

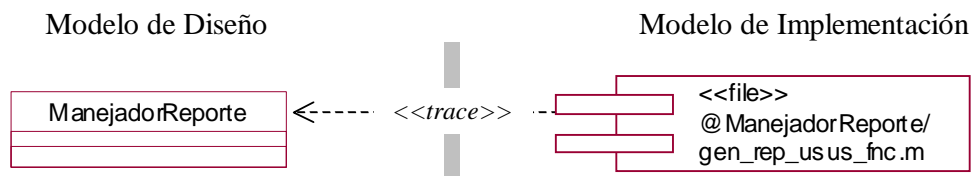


Figura 20.20. Trazas de la clase de diseño `ManejadorReporte`

20.9 SUBSISTEMA DE IMPLEMENTACIÓN – OBTENER IMAGEN

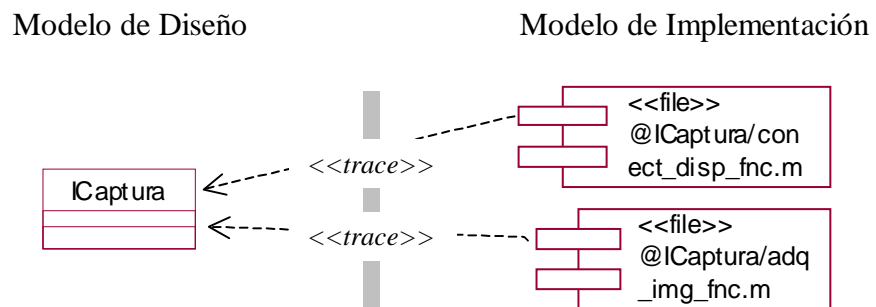


Figura 20.21. Trazas de la clase de diseño `ICaptura`

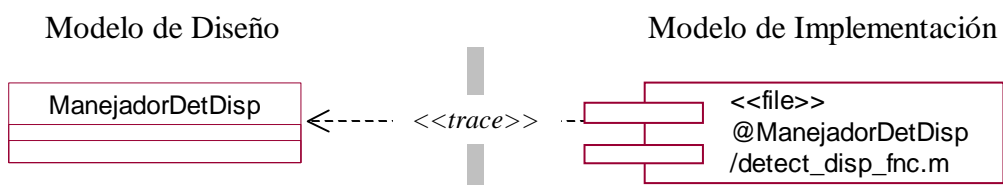


Figura 20.22. Trazas de la clase de diseño `ManejadorDetDisp`

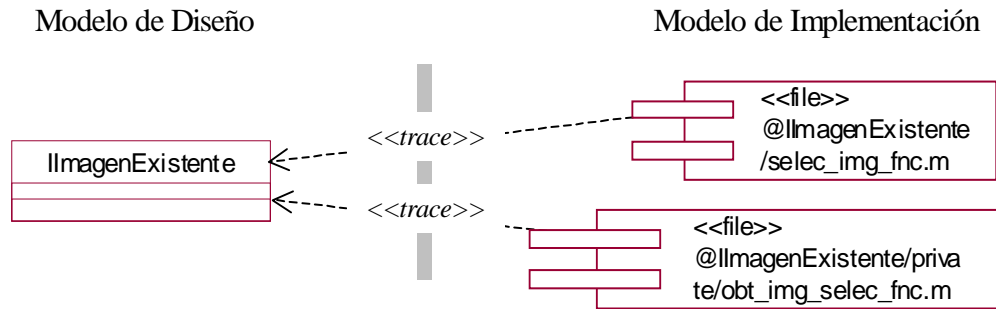


Figura 20.23. Trazas de la clase de diseño `ImagenExistente`

20.10 SUBSISTEMA DE IMPLEMENTACIÓN – LOCALIZAR ROSTRO

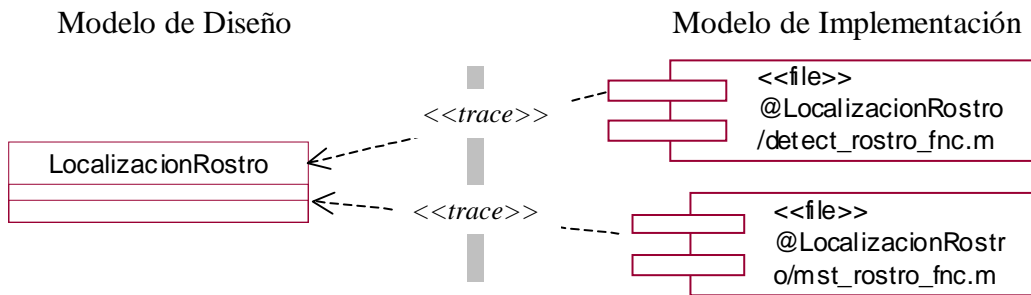


Figura 20.24. Trazas de la clase de diseño `LocalizacionRostro`

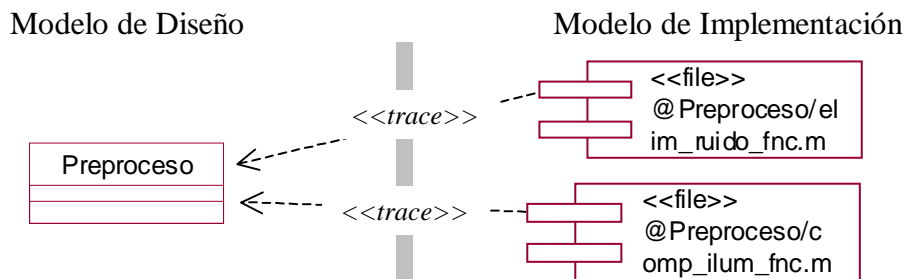


Figura 20.25. Trazas de la clase de diseño `Preproceso`

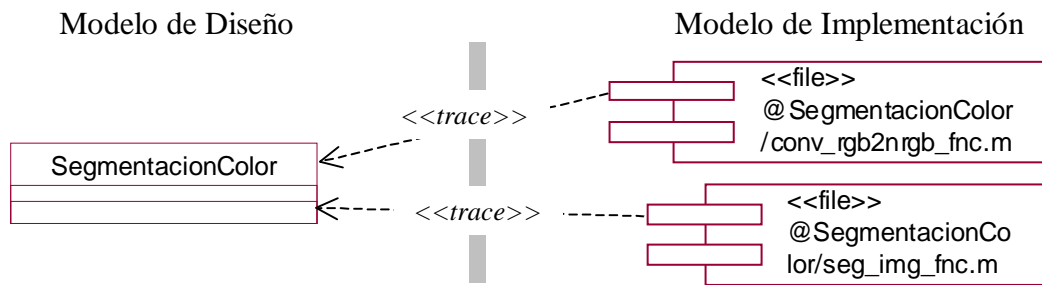


Figura 20.26. Trazas de la clase de diseño SegmentacionColor

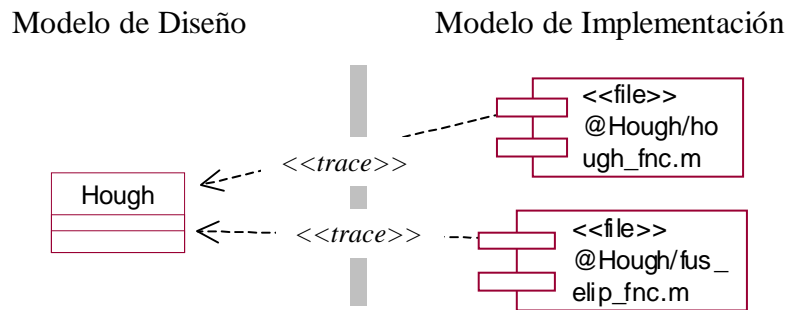


Figura 20.27. Trazas de la clase de diseño Hough

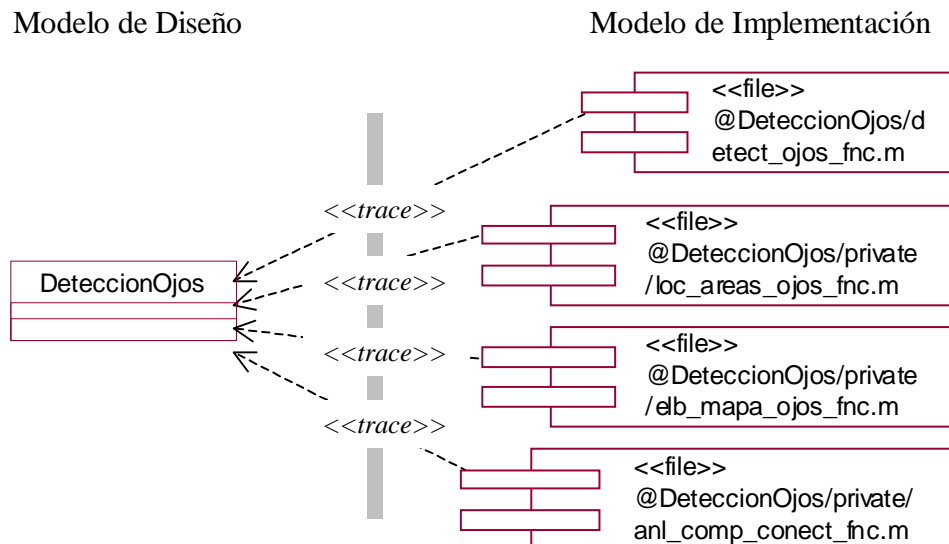


Figura 20.28. Trazas de la clase de diseño DeteccionOjos

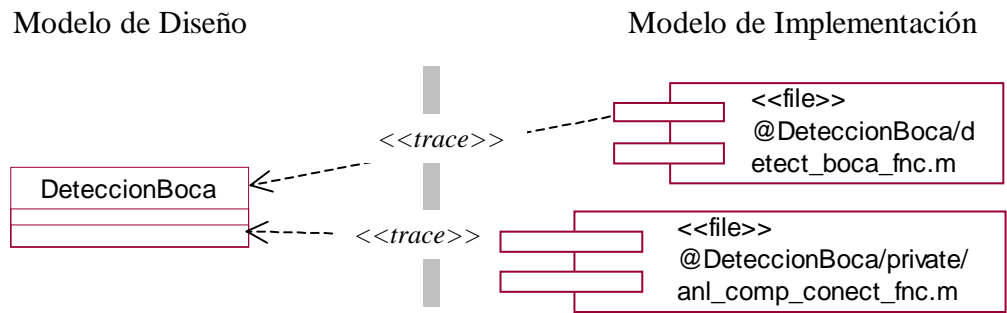


Figura 20.29. Trazas de la clase de diseño DeteccionBoca

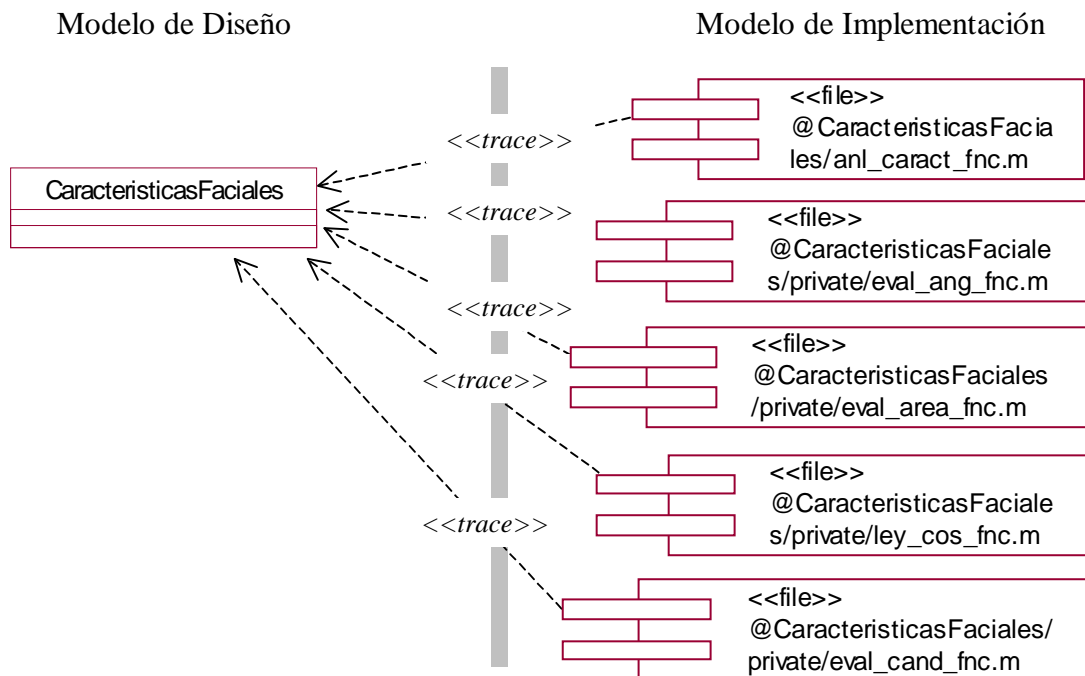


Figura 20.30. Trazas de la clase de diseño CaracteristicasFaciales

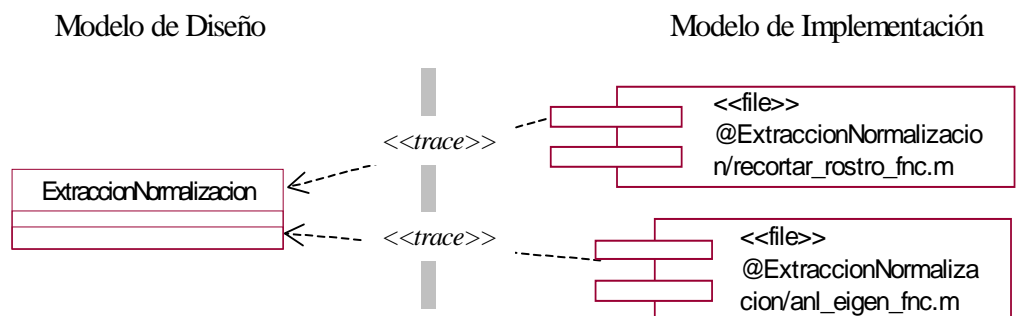


Figura 20.31. Trazas de la clase de diseño ExtraccionNormalizacion

20.11 SUBSISTEMA DE IMPLEMENTACIÓN – GUARDAR IMAGEN

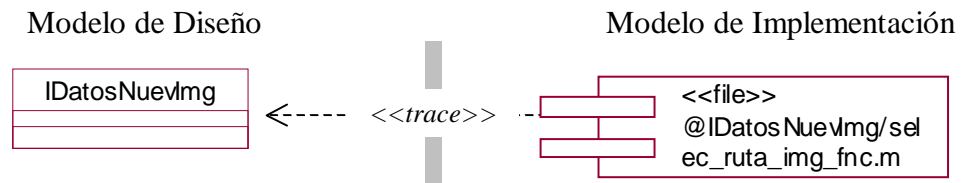


Figura 20.32. Trazas de la clase de diseño `IDatosNuevImg`

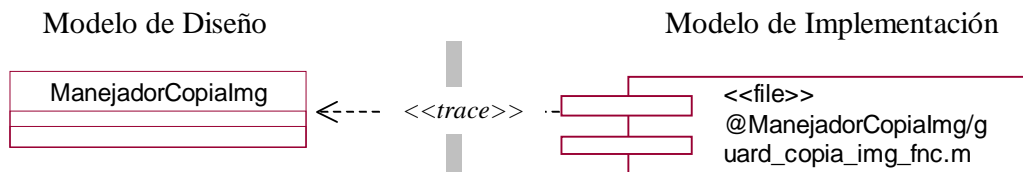


Figura 20.33. Trazas de la clase de diseño `ManejadorCopialmg`

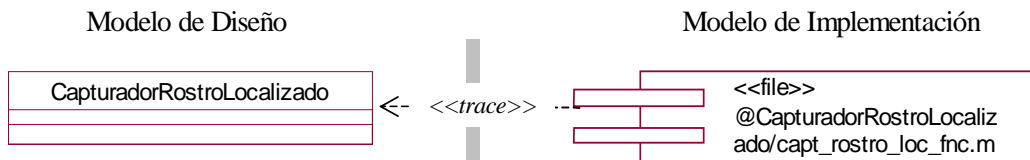


Figura 20.34. Trazas de la clase de diseño `CapturadorRostroLocalizado`

20.12 PAQUETE GLOBAL DE IMPLEMENTACIÓN – SOPORTE

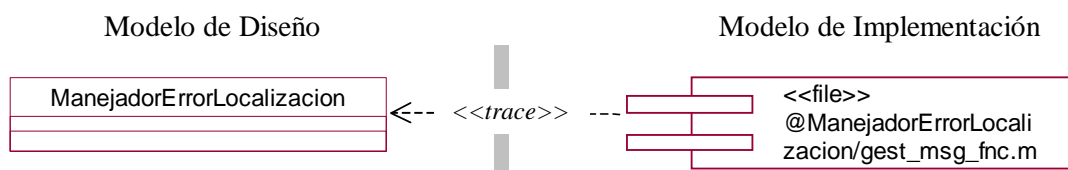


Figura 20.35. Trazas de la clase de diseño `ManejadorErrorLocalizacion`

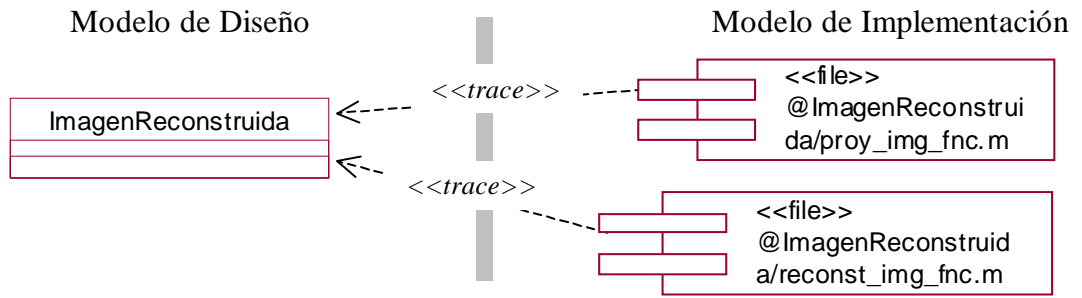


Figura 20.36. Trazas de la clase de diseño ImagenReconstruida

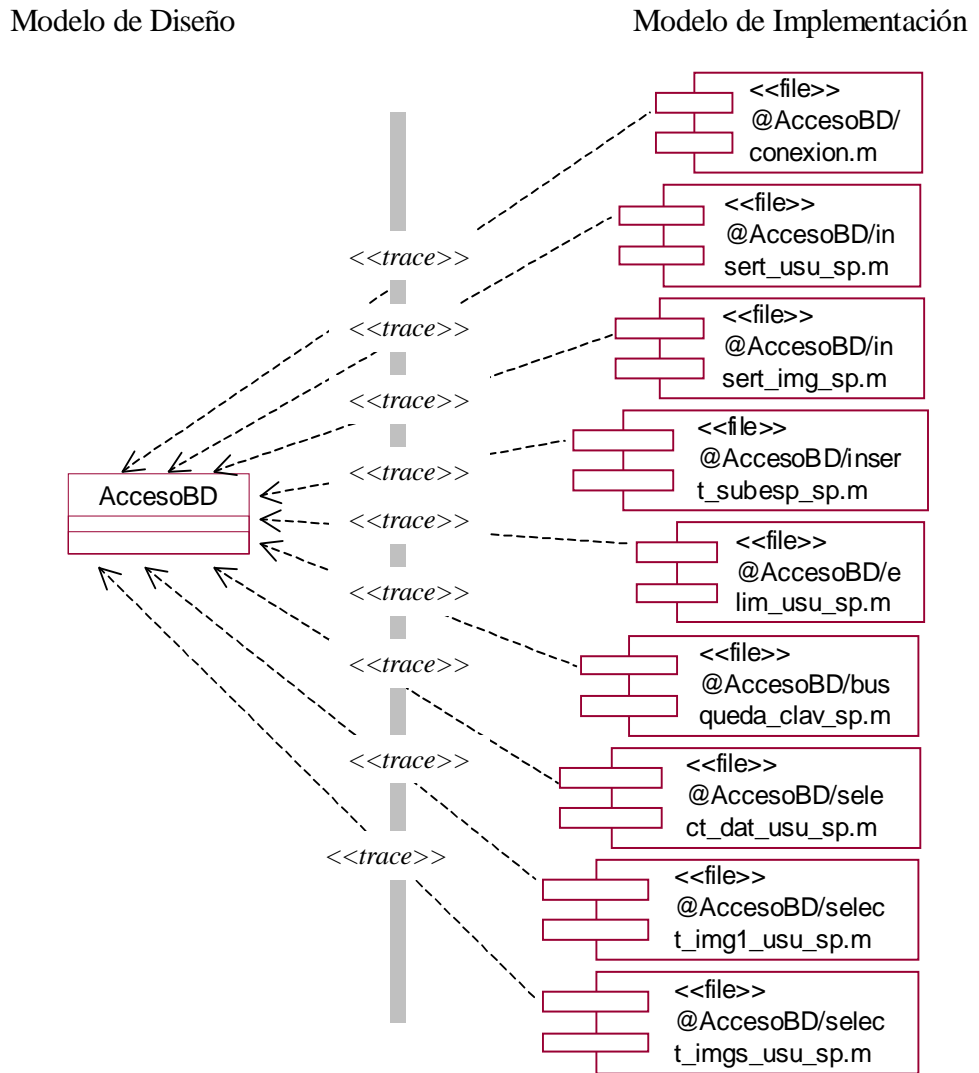


Figura 20.37. Trazas de la clase de diseño AccesoBD (parte I)

Modelo de Diseño

Modelo de Implementación

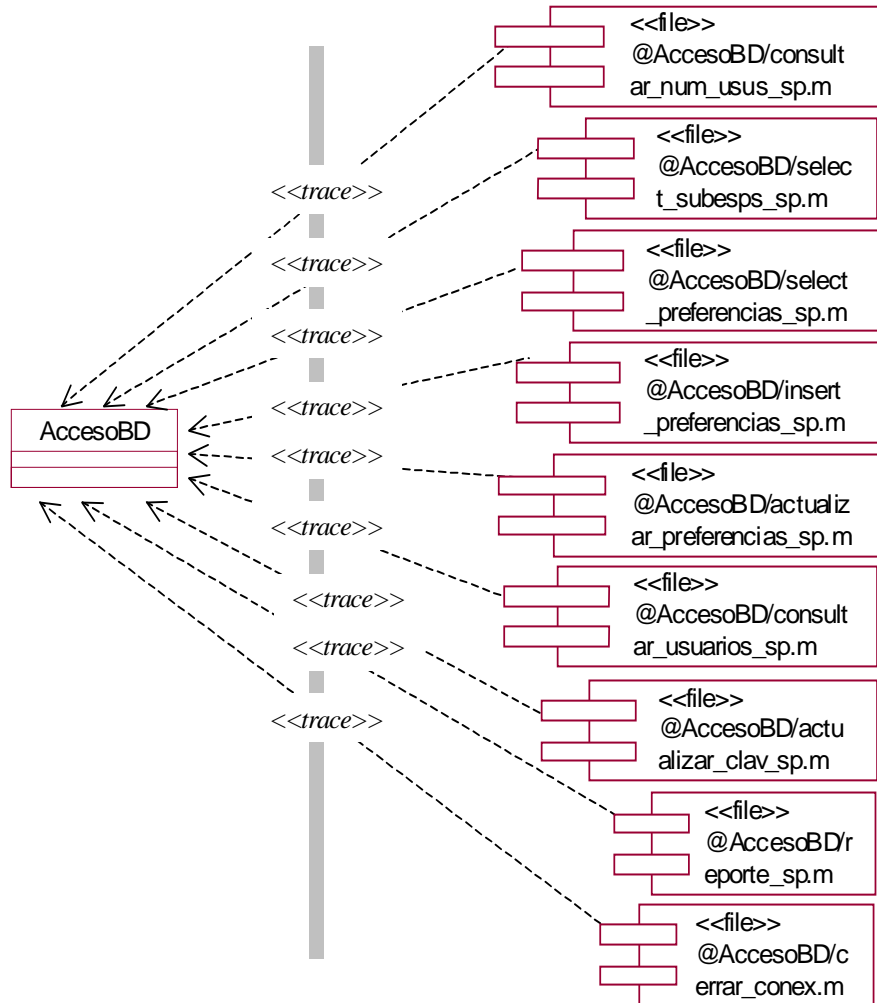


Figura 20.38. Trazas de la clase de diseño AccesoBD (parte II)

Anexo 21

DIAGRAMAS DE COMPONENTES

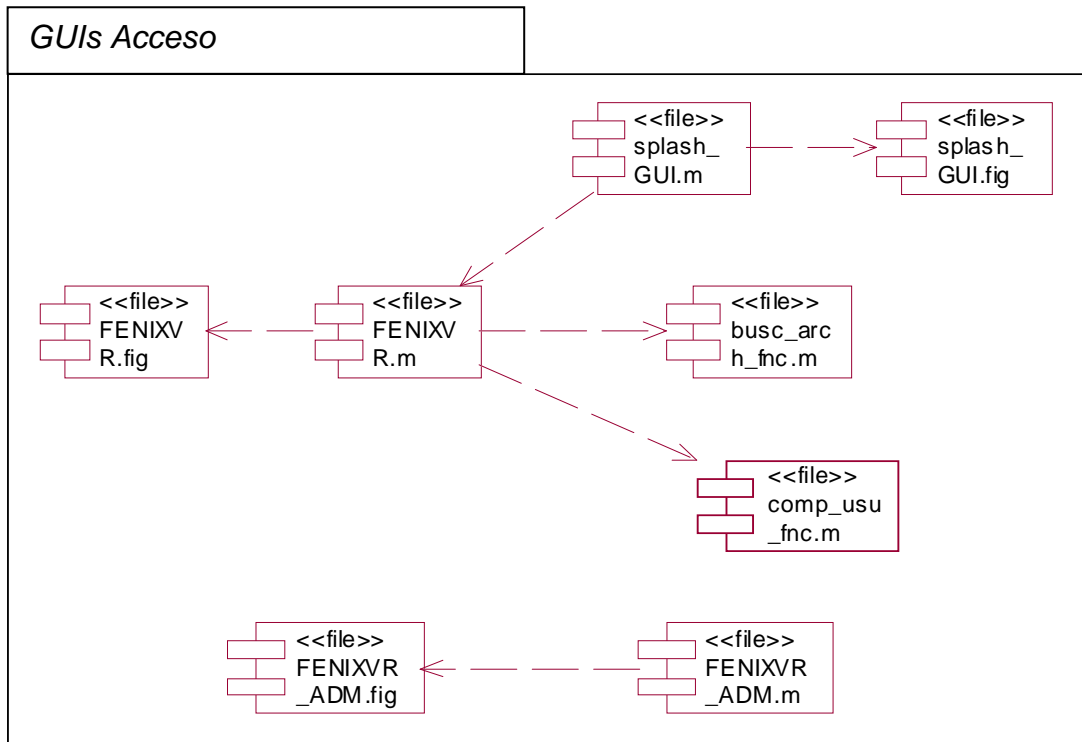


Figura 21.1. Subsistema GUIs Acceso

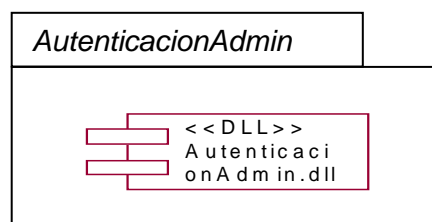


Figura 21.2. Subsistema AutenticacionAdmin

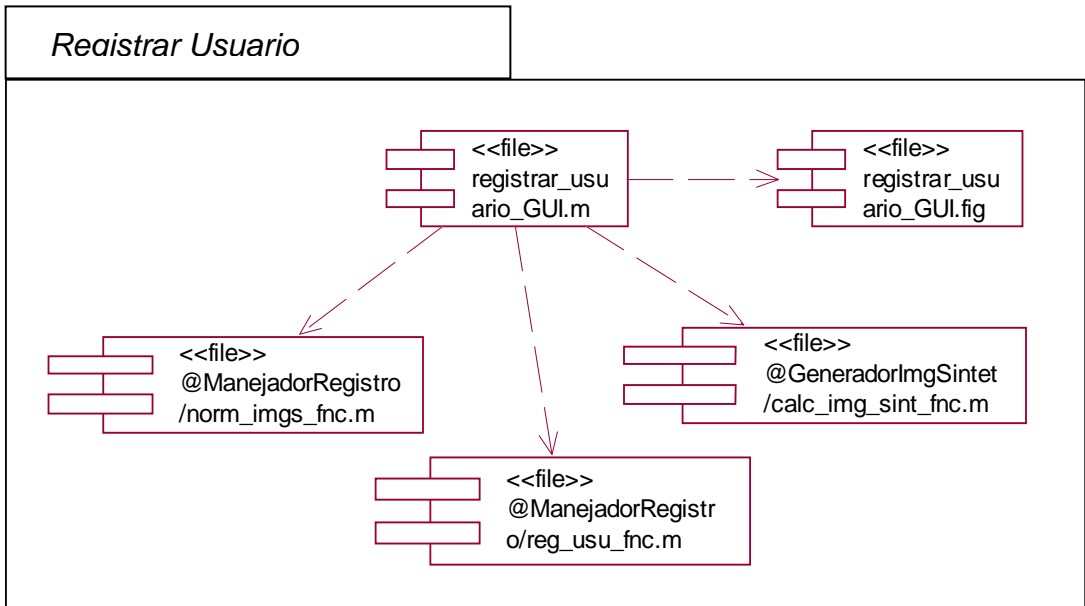


Figura 21.3. Subsistema Registrar Usuario

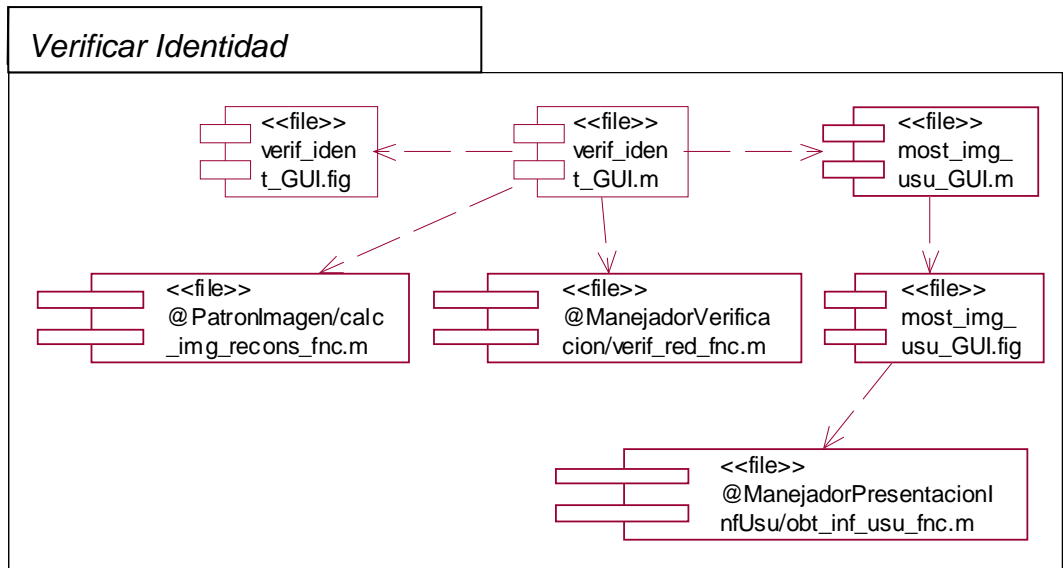


Figura 21.4. Subsistema Verificar Identidad

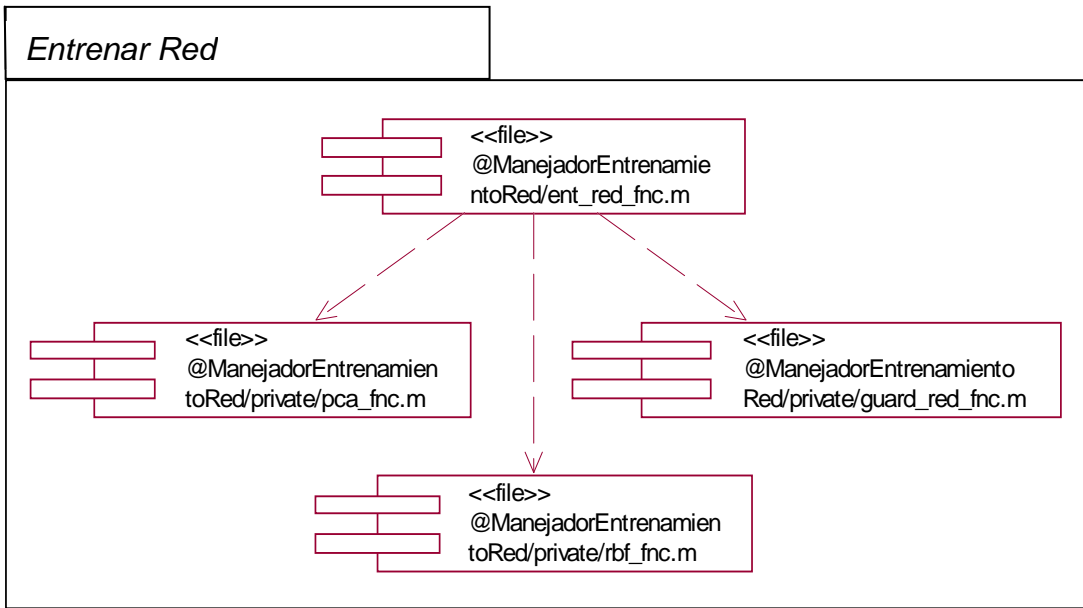


Figura 21.5. Subsistema Entrenar Red

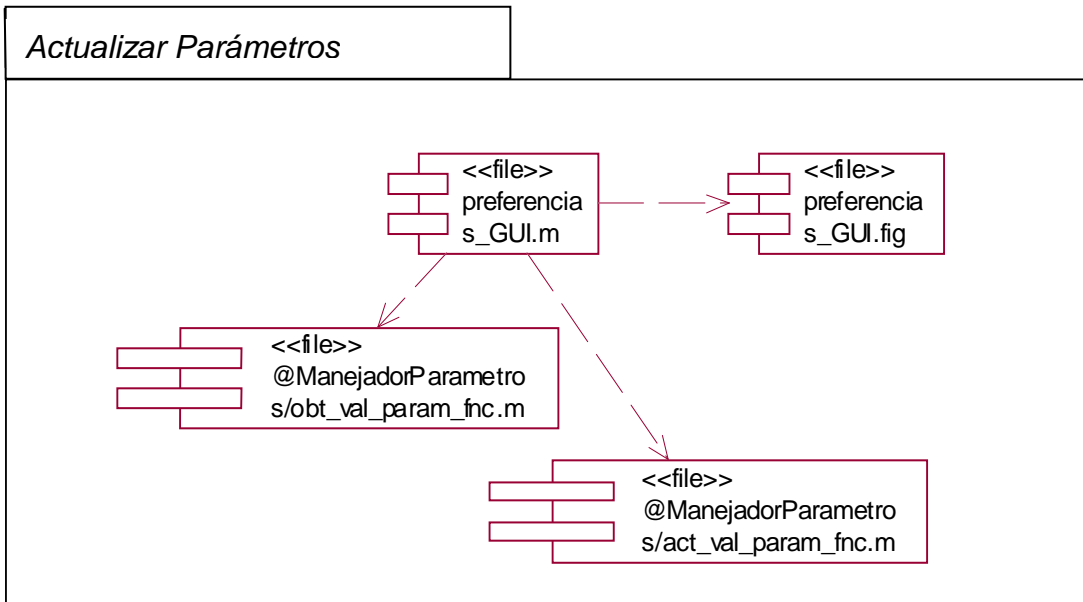


Figura 21.6. Subsistema Actualizar Parámetros

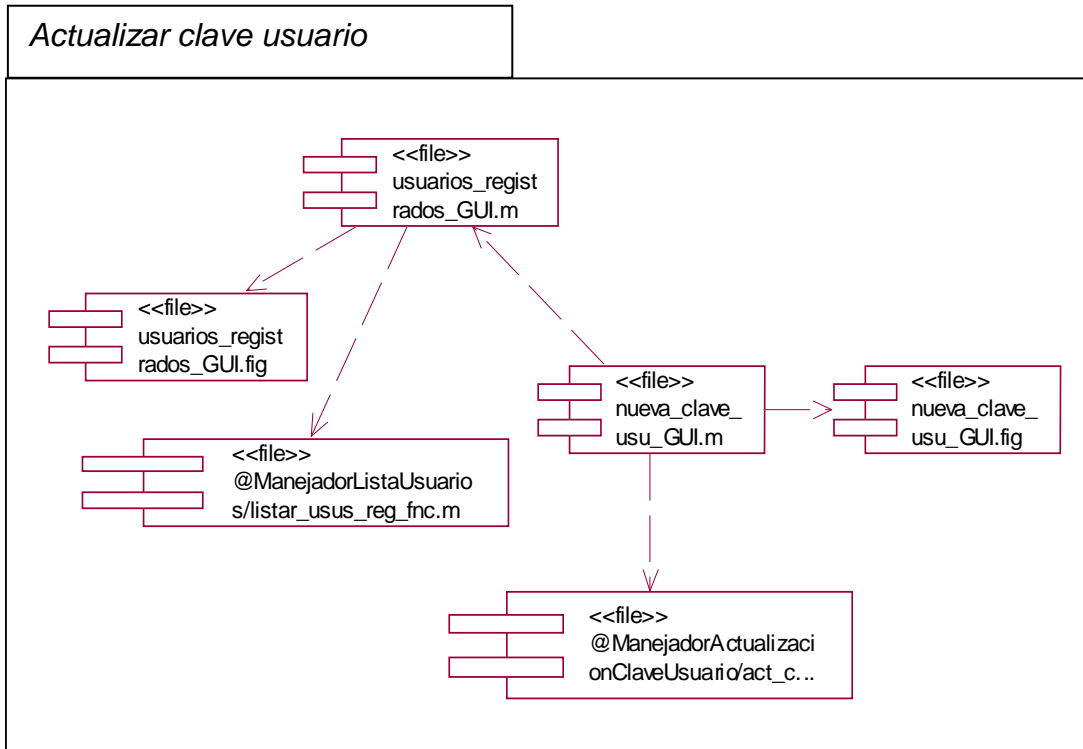


Figura 21.7. Subsistema Actualizar clave usuario

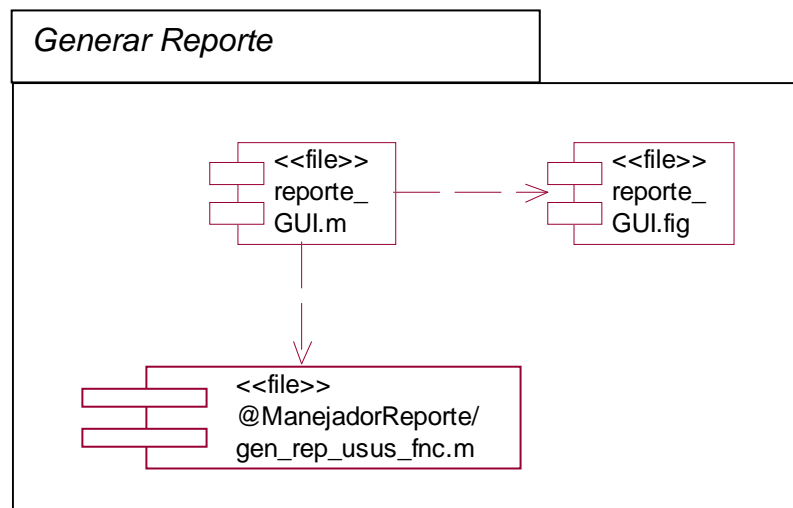


Figura 21.8. Subsistema Generar Reporte

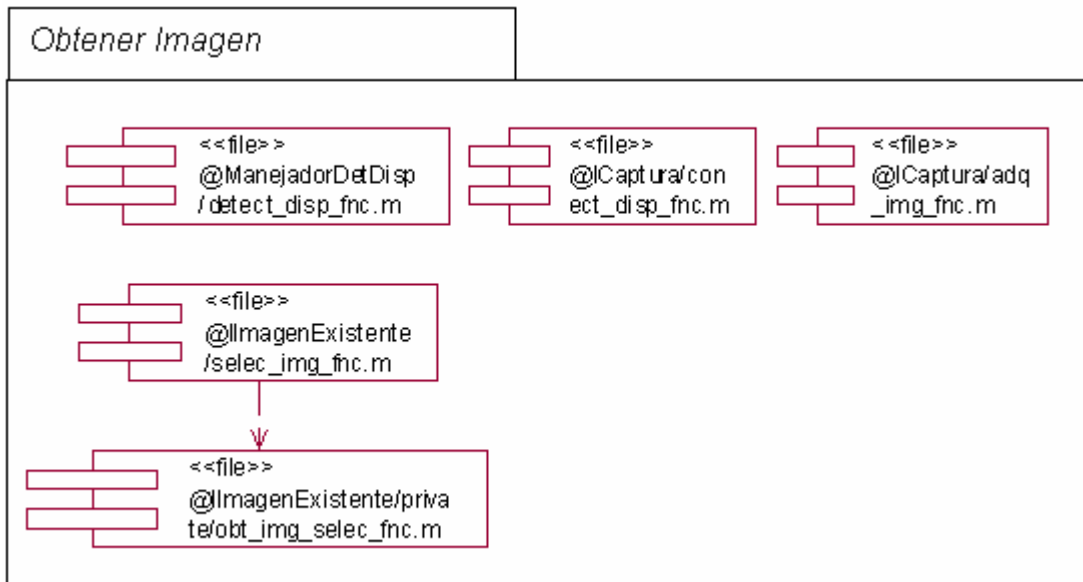


Figura 21.9. Subsistema Obtener Imagen

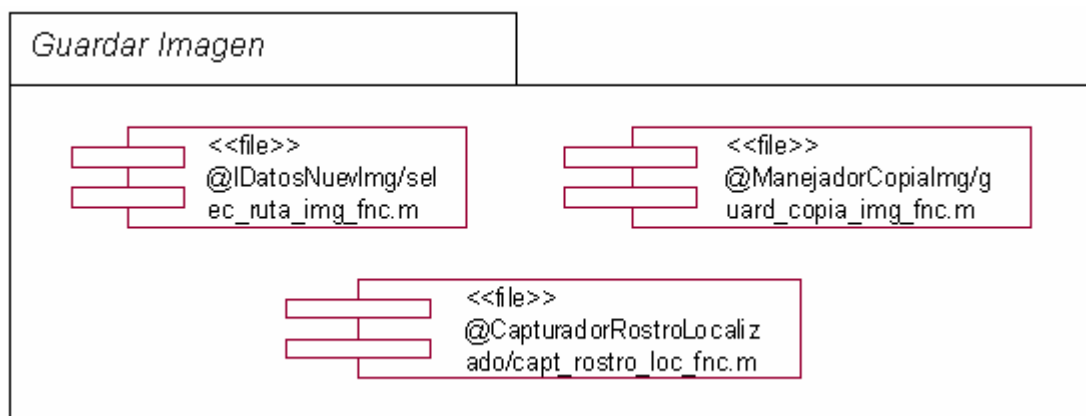


Figura 21.10. Subsistema Guardar Imagen

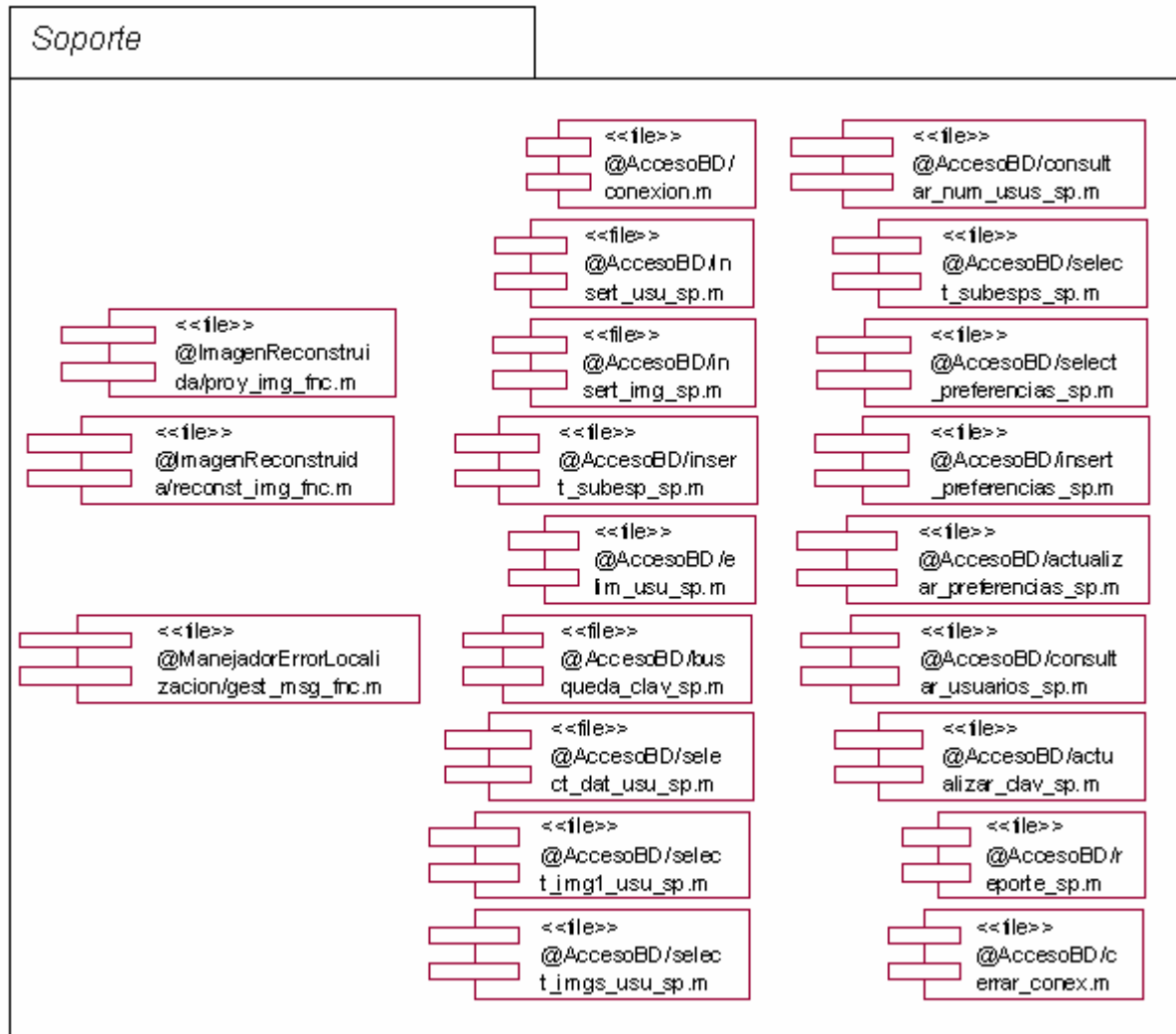


Figura 21.11. Paquete global – Soporte

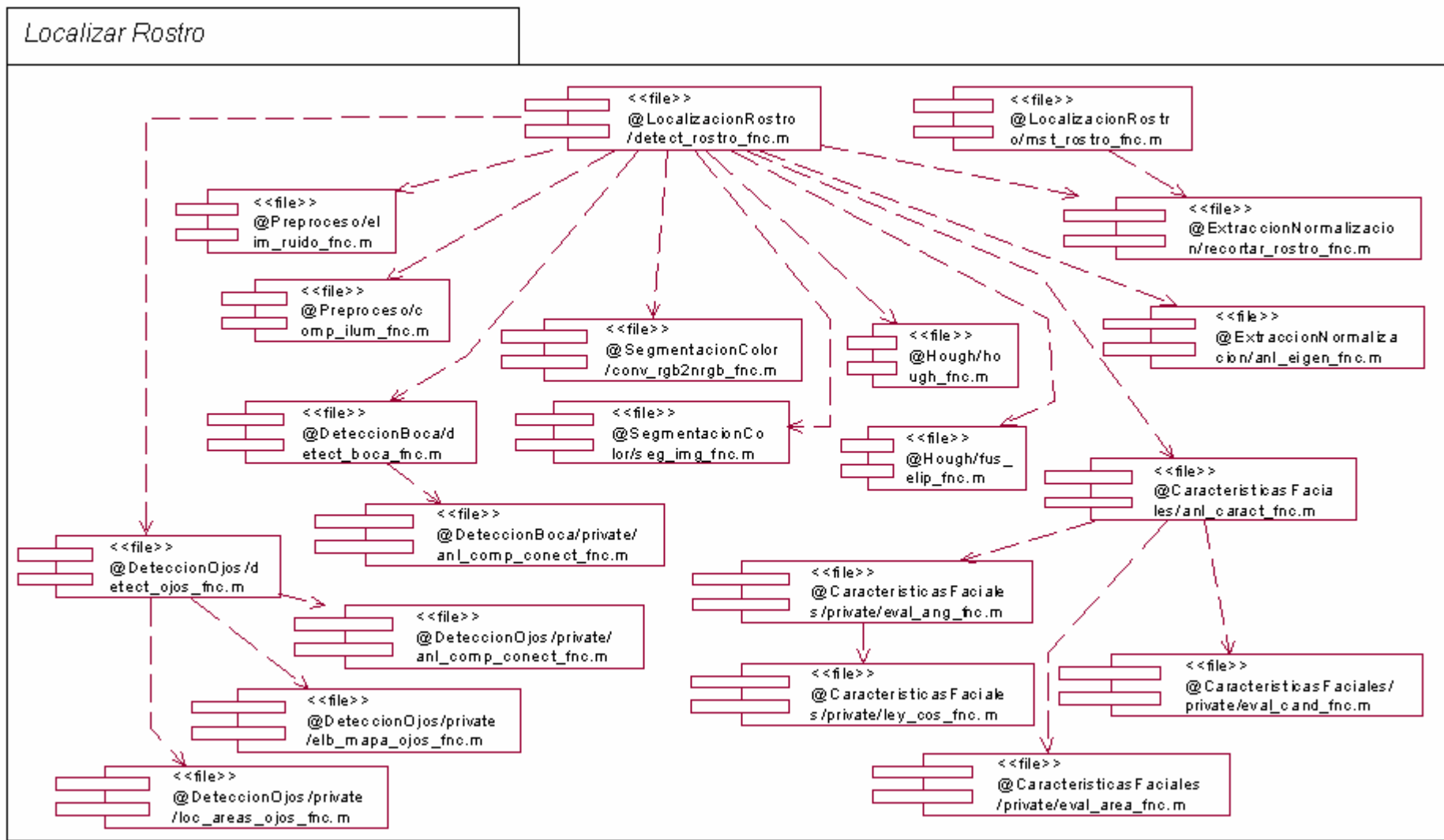


Figura 21.12. Subsistema Localizar Rostro

Anexo 22

PRUEBAS DE UNIDAD CASOS DE PRUEBA

CP - 001		CLASE : FENIXVR	
Objetivo	Solicitar registrar un nuevo usuario		
Entrada	---		
Salida Esperada	Se despliega la ventana de registro		
Condiciones	---		
Procedimiento	<ol style="list-style-type: none">1. Iniciar la solución computacional.2. Cuando aparezca la pantalla del menú principal presionar el botón <i>Registrar Usuario</i>.		

Tabla 22.1. Descripción del Caso de Prueba CP-001

CP - 002		CLASE : FENIXVR	
Objetivo	Solicitar entrenar red neuronal cuando existen usuarios registrados previamente.		
Entrada	---		
Salida Esperada	Mensaje para confirmar si se desea iniciar el proceso de entrenamiento de la red neuronal.		
Condiciones	---		
Procedimiento	<ol style="list-style-type: none">1. Iniciar la solución computacional.2. Cuando aparezca la pantalla del menú principal presionar el botón <i>Entrenar Red Neuronal</i>.		

Tabla 22.2. Descripción del Caso de Prueba CP-002

CP - 003		CLASE : FENIXVR	
Objetivo	Solicitar entrenar red neuronal cuando no existen usuarios registrados previamente.		
Entrada	---		
Salida Esperada	Mensaje de error: No existen usuarios registrados.		
Condiciones	---		
Procedimiento	<ol style="list-style-type: none"> 1. Iniciar la solución computacional. 2. Cuando aparezca la pantalla del menú principal presionar el botón <i>Entrenar Red Neuronal</i>. 		

Tabla 22.3. Descripción del Caso de Prueba CP-003

CP - 004		CLASE : FENIXVR	
Objetivo	Solicitar verificar la identidad de un usuario cuando se ha realizado previamente el entrenamiento de la red neuronal.		
Entrada	---		
Salida Esperada	Se despliega la ventana de verificación de identidad.		
Condiciones	---		
Procedimiento	<ol style="list-style-type: none"> 1. Iniciar la solución computacional 2. Cuando aparezca la pantalla del menú principal presionar el botón <i>Verificar Identidad</i>. 		

Tabla 22.4. Descripción del Caso de Prueba CP-004

CP - 005		CLASE : FENIXVR	
Objetivo	Solicitar verificar la identidad de un usuario cuando no se ha realizado previamente el entrenamiento de la red neuronal.		
Entrada	---		
Salida Esperada	Mensaje de error.		
Condiciones	---		
Procedimiento	<ol style="list-style-type: none"> 1. Iniciar la solución computacional 2. Cuando aparezca la pantalla del menú principal presionar el botón <i>Verificar Identidad</i>. 		

Tabla 22.5. Descripción del Caso de Prueba CP-005

CP - 006		CLASE : FENIXVR
Objetivo	Solicitar actualizar los valores de los parámetros utilizados para la creación y entrenamiento de la red neuronal.	
Entrada	---	
Salida Esperada	Se despliega la ventana de preferencias.	
Condiciones	---	
Procedimiento	<ol style="list-style-type: none"> 1. Iniciar la solución computacional 2. Cuando aparezca la pantalla del menú principal presionar el botón <i>Preferencias</i>. 	

Tabla 22.6. Descripción del Caso de Prueba CP-006

CP - 007		CLASE : registrar_usuario_GUI
Objetivo	Intentar ingresar dígitos en los cuadros de texto designados para el nombre y para el apellido del nuevo usuario (txt_nomb y txt_apel).	
Entrada	Nombre usuario = 123 Apellido usuario = 465	
Salida Esperada	Mensajes de error que solicitan verificar formato.	
Condiciones	---	
Procedimiento	<ol style="list-style-type: none"> 1. Ingresar a la ventana de registro de un nuevo usuario. 2. Obtener una imagen (desde archivo o a través de un dispositivo de captura). 3. Pulsar botón <i>Analizar</i>. 4. Confirmar agregar la imagen del rostro localizado al registro del nuevo usuario. 5. Cuando se habilitan los cuadros de texto para el nombre y apellido del nuevo usuario ingresar: 123 en el cuadro de texto para el nombre y luego 456 en el cuadro de texto para el apellido. 	

Tabla 22.7. Descripción del Caso de Prueba CP-007

CP - 008		CLASE : registrar_usuario_GUI
Objetivo	Solicitar almacenar la información del nuevo usuario cuando todos los datos son correctos.	
Entrada	Nombre = Yolanda Apellido = Alcuacer Clave = 1002116901 Género = Femenino	
Salida Esperada	Se inicia el proceso de almacenamiento.	
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.	
Procedimiento	<ol style="list-style-type: none"> 1. Ingresar a la ventana de registro de un nuevo usuario. 2. Obtener una imagen (desde archivo o a través de un dispositivo de captura). 3. Pulsar botón <i>Analizar</i>. 4. Confirmar agregar la imagen del rostro localizado al registro del nuevo usuario. 5. Ingresar el nombre, apellido, clave y género del nuevo usuario. 6. Pulsar botón <i>Registrar Usuario</i>. 	

Tabla 22.8. Descripción del Caso de Prueba CP-008

CP - 009		CLASE : registrar_usuario_GUI
Objetivo	Solicitar almacenar la información del nuevo usuario cuando la clave ingresada ya existe en la base de datos.	
Entrada	Nombre = Juan Apellido = Alcuacer Clave = 1002116901 Género = Masculino	
Salida Esperada	Mensaje de error: Clave de usuario no válida.	
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.	
Procedimiento	<ol style="list-style-type: none"> 1. Ingresar a la ventana de registro de un nuevo usuario. 2. Obtener una imagen (desde archivo o a través de un dispositivo de captura). 3. Pulsar botón <i>Analizar</i>. 4. Confirmar agregar la imagen del rostro localizado al registro del nuevo usuario. 5. Ingresar el nombre, apellido, clave y género del nuevo usuario. 6. Pulsar botón <i>Registrar Usuario</i>. 	

Tabla 22.9. Descripción del Caso de Prueba CP-009

CP - 010		CLASE : ImagenExistente
Objetivo	Seleccionar una imagen desde una unidad de almacenamiento existente (ruta correcta).	
Entrada	ruta = C:\IMAGENES\imagen.jpg	
Salida Esperada	Se obtiene la imagen desde la ruta especificada y se obtienen sus dimensiones.	
Condiciones	Esta prueba se ejecuta dentro de la ventana de registro de usuario.	
Procedimiento	<ol style="list-style-type: none"> 1. Dentro de la ventana de registro de usuario seleccionar en el menú <i>Archivo</i>, la opción <i>Obtener Imagen -> Carpeta</i>. 2. Dentro del cuadro de diálogo que aparece seleccionar C:\IMAGENES\imagen.jpg 3. Pulsar botón <i>Abrir</i> del cuadro de diálogo. 	

Tabla 22.10. Descripción del Caso de Prueba CP-010

CP - 011		CLASE : ImagenExistente
Objetivo	Tratar de obtener una imagen desde una ruta incorrecta.	
Entrada	ruta = AA:\imagen.jpg	
Salida Esperada	Mensaje de error: Ruta no válida.	
Condiciones	Esta prueba se ejecuta dentro de la ventana de registro de usuario.	
Procedimiento	<ol style="list-style-type: none"> 1. Dentro de la ventana de registro de usuario seleccionar en el menú <i>Archivo</i>, la opción <i>Obtener Imagen -> Carpeta</i>. 2. Dentro del cuadro de diálogo que aparece ingresar AA:\imagen.jpg 3. Pulsar botón <i>Abrir</i> del cuadro de diálogo. 	

Tabla 22.11. Descripción del Caso de Prueba CP-011

CP - 012		CLASE : ICaptura
Objetivo	Adquirir una imagen a través de una cámara web conectada al computador donde se ejecuta la solución computacional.	
Entrada	---	

Salida Esperada	Se obtiene una imagen a través de la cámara web y se obtienen las dimensiones de dicha imagen.
Condiciones	No se permite el uso de la cámara web por otras aplicaciones. Esta prueba se ejecuta dentro de la ventana de registro de usuario.
Procedimiento	<ol style="list-style-type: none"> 1. Dentro de la ventana de registro de usuario seleccionar en el menú <i>Archivo</i>, la opción <i>Obtener Imagen -> Video</i>. 2. En el cuadro de diálogo que indica el nombre del dispositivo detectado (cámara web compatible conectada) pulsar el botón <i>Aceptar</i>.

Tabla 22.12. Descripción del Caso de Prueba CP-012

CP - 013		CLASE : ICaptura	
Objetivo	Tratar de adquirir una imagen a través de una cámara web que no se encuentra disponible. El computador donde se ejecuta la solución computacional no tiene una cámara web compatible conectada.		
Entrada			---
Salida Esperada	Mensaje de error: No se ha detectado dispositivo.		
Condiciones	Esta prueba se ejecuta dentro de la ventana de registro de usuario.		
Procedimiento	<ol style="list-style-type: none"> 1. Dentro de la ventana de registro de usuario seleccionar en el menú <i>Archivo</i>, la opción <i>Obtener Imagen -> Video</i>. 		

Tabla 22.13. Descripción del Caso de Prueba CP- 013

CP - 014		CLASE : verif_ident_GUI	
Objetivo	Solicitar verificar la identidad de un usuario empleando una clave existente en la base de datos.		
Entrada	Clave = 1002116901		
Salida Esperada	Se inicia el proceso de verificación de identidad.		

Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.
Procedimiento	<ol style="list-style-type: none"> 1. Ingresar a la ventana de verificación de identidad. 2. Obtener una imagen (desde archivo o a través de un dispositivo de captura). 3. Pulsar botón <i>Analizar</i>. 4. Confirmar emplear la imagen del rostro localizado para la verificación de identidad. 5. Cuando se habilita el cuadro de texto para la clave de usuario, ingresar: 1002116901. 6. Pulsar el botón <i>Verificar Identidad</i>

Tabla 22.14. Descripción del Caso de Prueba CP-014

CP - 015		CLASE : verif_ident_GUI
Objetivo	Solicitar verificar la identidad de un usuario empleando una clave que no existe en la base de datos.	
Entrada	Clave = 1332886901	
Salida Esperada	Mensaje de error: No existe un usuario previamente registrado con la clave ingresada.	
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.	
Procedimiento	<ol style="list-style-type: none"> 1. Ingresar a la ventana de verificación de identidad. 2. Obtener una imagen (desde archivo o a través de un dispositivo de captura). 3. Pulsar botón <i>Analizar</i>. 4. Confirmar emplear la imagen del rostro localizado para la verificación de identidad. 5. Cuando se habilita el cuadro de texto para la clave de usuario, ingresar: 1332886901. 6. Pulsar el botón <i>Verificar Identidad</i> 	

Tabla 22.15. Descripción del Caso de Prueba CP-015

CP - 016		CLASE : verif_ident_GUI
Objetivo	Solicitar verificar la identidad de un usuario empleando una clave que contiene únicamente letras.	
Entrada	Clave = abc	
Salida Esperada	Mensaje de error: Verificar formato de la clave.	
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.	
Procedimiento	<ol style="list-style-type: none"> 1. Ingresar a la ventana de verificación de identidad. 2. Obtener una imagen (desde archivo o a través de un dispositivo de captura). 3. Pulsar botón <i>Analizar</i>. 4. Confirmar emplear la imagen del rostro localizado para la verificación de identidad. 5. Cuando se habilita el cuadro de texto para la clave de usuario, ingresar: <i>abc</i>. 6. Pulsar el botón <i>Verificar Identidad</i> 	

Tabla 22.16. Descripción del Caso de Prueba CP-016

CP - 017		CLASE : FENIXVR_ADM
Objetivo	Solicitar actualizar la clave de un usuario.	
Entrada	---	
Salida Esperada	Se despliega la ventana de usuarios registrados.	
Condiciones	---	
Procedimiento	<ol style="list-style-type: none"> 1. Iniciar el módulo de administración de la solución computacional. 2. Cuando aparezca la pantalla del menú de administración presionar el botón <i>Actualizar Clave</i>. 	

Tabla 22.17. Descripción del Caso de Prueba CP-017

CP - 018		CLASE : FENIXVR_ADM
Objetivo	Solicitar el reporte de usuarios registrados.	
Entrada	---	
Salida Esperada	Se despliega el reporte de usuarios registrados	

Condiciones	---
Procedimiento	<ol style="list-style-type: none"> 1. Iniciar el módulo de administración de la solución computacional. 2. Cuando aparezca la pantalla del menú de administración presionar el botón <i>Ver Reporte</i>.

Tabla 22.18. Descripción del Caso de Prueba CP-018

CP - 019 CLASE : nueva_clave_usu_GUI	
Objetivo	Solicitar actualizar la clave de un usuario empleando una clave nueva con formato correcto.
Entrada	Nueva clave = 25468
Salida Esperada	Se inicia el proceso de actualización de clave.
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.
Procedimiento	<ol style="list-style-type: none"> 1. En la ventana para establecer la nueva clave del usuario ingresar 25468 como nueva clave y confirmarla. 2. Pulsar el botón <i>Aceptar</i>.

Tabla 22.19. Descripción del Caso de Prueba CP-019

CP - 020 CLASE : nueva_clave_usu_GUI	
Objetivo	Solicitar actualizar la clave de un usuario empleando una clave nueva con formato incorrecto.
Entrada	Nueva clave = abcdw23
Salida Esperada	Mensaje de error: Verificar formato de la nueva clave.
Condiciones	---
Procedimiento	<ol style="list-style-type: none"> 1. En la ventana para establecer la nueva clave del usuario ingresar <i>abcdw23</i> como nueva clave y confirmarla. 2. Pulsar el botón <i>Aceptar</i>.

Tabla 22.20. Descripción del Caso de Prueba CP-020

CP - 021		CLASE : BuscadorUsuario
Objetivo	Buscar un usuario que está registrado en la base de datos.	
Entrada	Clave = 1002116901	
Salida Esperada	Retorna valor = 1	
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.	
Procedimiento	<ol style="list-style-type: none"> 1. Iniciar el entorno de Matlab. 2. Crear un objeto de la clase BuscadorUsuario dándole como atributo la clave de búsqueda. 3. Crear un objeto de la clase AccesoBD 4. Invocar la operación validar_clave_fnc() 	

Tabla 22.21. Descripción del Caso de Prueba CP-021

CP - 022		CLASE : BuscadorUsuario
Objetivo	Buscar un usuario empleando una clave que no pertenece a ningún usuario registrado.	
Entrada	Clave = 123456	
Salida Esperada	Retorna valor = 0	
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.	
Procedimiento	<ol style="list-style-type: none"> 1. Iniciar el entorno de Matlab. 2. Crear un objeto de la clase BuscadorUsuario dándole como atributo la clave de búsqueda. 3. Crear un objeto de la clase AccesoBD 4. Invocar la operación validar_clave_fnc() 	

Tabla 22.22. Descripción del Caso de Prueba CP-022

CP - 023		CLASE : ManejadorDetDisp
Objetivo	Detectar una cámara web compatible conectada al computador donde se ejecuta la solución computacional.	
Entrada	---	

Salida Esperada	Retorna un objeto de la clase ManejadorDetDisp cuyo atributo disp_id_i es el identificador asociado al dispositivo detectado (cámara web).
Condiciones	No se permite el uso de la cámara web por otras aplicaciones.
Procedimiento	<ol style="list-style-type: none"> 1. Iniciar el entorno de Matlab. 2. Crear un objeto de la clase ManejadorDetDisp dándole como atributo el nombre de la interfaz soportada ('winvideo'). 3. Invocar la operación detect_disp_fnc ()

Tabla 22.23. Descripción del Caso de Prueba CP-023

CP - 024		CLASE : ManejadorDetDisp	
Objetivo	Intentar detectar una cámara web que está siendo utilizada por otra aplicación.		
Entrada	---		
Salida Esperada	Retorna un objeto de la clase ManejadorDetDisp cuyo atributo num_adap_disp_l tiene valor = 0.		
Condiciones	---		
Procedimiento	<ol style="list-style-type: none"> 1. Iniciar el entorno de Matlab. 2. Crear un objeto de la clase ManejadorDetDisp dándole como atributo el nombre de la interfaz soportada ('winvideo'). 3. Invocar la operación detect_disp_fnc () 		

Tabla 22.24. Descripción del Caso de Prueba CP-024

CP - 025		CLASE : LocalizacionRostro	
Objetivo	Localizar el rostro contenido en una imagen.		
Entrada	Imagen RGB que contiene un rostro humano (<i>rostro.bmp</i>).		
Salida Esperada	Se ha localizado el rostro contenido en la imagen. Despliega en pantalla la imagen RGB con el área del rostro señalada y retorna un objeto de la clase LocalizacionRostro cuyos atributos representan la información requerida sobre la imagen del rostro.		
Condiciones	---		

Procedimiento	<ol style="list-style-type: none"> 1. Iniciar el entorno de Matlab. 2. Crear un objeto de la clase LocalizaciónRostro 3. Invocar la operación detect_rostro_fnc () proporcionándole el objeto antes creado, la imagen <i>rostro.bmp</i>, y el manejador de la figura o eje donde se desea desplegar la imagen con el rostro señalado.
---------------	--

Tabla 22.25. Descripción del Caso de Prueba CP-025

CP - 026		CLASE : LocalizacionRostro
Objetivo	Intentar localizar un rostro en una imagen que no contiene rostros humanos.	
Entrada	Imagen RGB de una casa (<i>casa.bmp</i>)	
Salida Esperada	No se ha localizado un rostro en la imagen. Retorna un objeto de la clase LocalizacionRostro en el que el atributo <i>band_err_i</i> es ≤ 10 .	
Condiciones	---	
Procedimiento	<ol style="list-style-type: none"> 1. Iniciar el entorno de Matlab. 2. Crear un objeto de la clase LocalizaciónRostro 3. Invocar la operación detect_rostro_fnc () proporcionándole el objeto antes creado, la imagen <i>casa.bmp</i>, y el manejador de una figura o eje. 	

Tabla 22.26. Descripción del Caso de Prueba CP-026

CP - 027		CLASE : LocalizacionRostro
Objetivo	Intentar localizar un rostro en una imagen en escala de grises.	
Entrada	Imagen en escala de grises que contiene un rostro humano (<i>rostro_gris.bmp</i>).	
Salida Esperada	No se ha encontrado un rostro en la imagen. La imagen no contiene áreas del color de la piel. Retorna un objeto de la clase LocalizacionRostro en el que el atributo <i>band_err_i</i> es ≤ 10 .	
Condiciones	---	

Procedimiento	<ol style="list-style-type: none"> 1. Iniciar el entorno de Matlab. 2. Crear un objeto de la clase LocalizaciónRostro 3. Invocar la operación detect_rostro_fnc () proporcionándole el objeto antes creado, la imagen <i>rostro_gris.bmp</i>, y el manejador de una figura o eje.
---------------	--

Tabla 22.27. Descripción del Caso de Prueba CP-027

CP - 028		CLASE : ManejadorEntrenamientoRed
Objetivo	Realizar el entrenamiento de la red neuronal empleando el mismo número de imágenes por cada usuario.	
Entrada	Base de datos que contiene cinco imágenes por cada usuario. Número total de usuarios = 10.	
Salida Esperada	Se ha almacenado en disco el archivo que contiene a la red neuronal, el archivo del eigenspacio y el archivo referente al rostro promedio.	
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.	
Procedimiento	<ol style="list-style-type: none"> 1. Iniciar el entorno de Matlab. 2. Crear un objeto de la clase ManejadorEntrenamientoRed, un objeto de la clase AccesoBD y uno de la clase ImagenReconstruida. 3. Invocar la operación ent_red_fnc () proporcionándole los objetos antes creado. 	

Tabla 22.28. Descripción del Caso de Prueba CP-028

CP - 029		CLASE : ManejadorEntrenamientoRed
Objetivo	Realizar el entrenamiento de la red neuronal empleando un número diferente de imágenes por cada usuario.	

Entrada	Base de datos que contiene un número diferente de imágenes por cada usuario. Número total de usuarios = 10. Usuarios 1 y 6 = 1 imagen Usuarios 2 y 7 = 2 imágenes Usuarios 3 y 8 = 3 imágenes Usuarios 4 y 9 = 4 imágenes Usuarios 5 y 10 = 5 imágenes
Salida Esperada	Se ha almacenado en disco el archivo que contiene a la red neuronal, el archivo del eigenspacio y el archivo referente al rostro promedio.
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.
Procedimiento	<ol style="list-style-type: none"> 1. Iniciar el entorno de Matlab. 2. Crear un objeto de la clase <code>ManejadorEntrenamientoRed</code>, un objeto de la clase <code>AccesoBD</code> y uno de la clase <code>ImagenReconstruida</code>. 3. Invocar la operación <code>ent_red_fnc ()</code> proporcionándole los objetos antes creado.

Tabla 22.29. Descripción del Caso de Prueba CP-029

CP - 030		CLASE : AutenticacionAdmin	
Objetivo	Verificar que el usuario actual tiene perfil de administrador del equipo o dominio, de acuerdo a la autenticación Windows.		
Entrada	---		
Salida Esperada	Retorna un valor = 1		
Condiciones	La sesión bajo la que se está ejecutando la solución computacional pertenece a un usuario con perfil de administrador del equipo o dominio de acuerdo a la autenticación Windows.		
Procedimiento	<ol style="list-style-type: none"> 1. Iniciar una sesión como administrador del equipo. 2. Iniciar el entorno de Matlab. 3. Registrar el dll <i>AutenticarAdmin.dll</i>. 4. Invocar la operación <code>IsAdmin()</code> 		

Tabla 22.30. Descripción del Caso de Prueba CP-030

CP - 031		CLASE : AutenticacionAdmin	
Objetivo	Verificar que el usuario actual no tiene perfil de administrador del equipo o dominio, de acuerdo a la autenticación Windows.		
Entrada	---		
Salida Esperada	Retorna un valor = 0		
Condiciones	La sesión bajo la que se está ejecutando la solución computacional pertenece a un usuario con perfil diferente al de un administrador del equipo o dominio de acuerdo a la autenticación Windows.		
Procedimiento	<ol style="list-style-type: none"> 1. Iniciar una sesión como invitado. 2. Iniciar el entorno de Matlab. 3. Registrar el dll <i>AutenticarAdmin.dll</i>. 4. Invocar la operación <code>IsAdmin()</code> 		

Tabla 22.31. Descripción del Caso de Prueba CP-031

Anexo 23

PRUEBAS DE INTEGRACIÓN Y DE SISTEMA CASOS DE PRUEBA

23.1 CASO DE USO REGISTRAR USUARIO

CP - 032	ESCENARIO : Flujo básico (imagen existente)
Objetivo	Registrar correctamente un nuevo usuario empleando una imagen obtenida desde una unidad de almacenamiento.
Entrada	Nombre = Yolanda Apellido = Alcuacer Clave = 1002116901 Género = Femenino Imágenes del Rostro = 1 imagen
Salida Esperada	Usuario registrado (información almacenada)
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.
Procedimiento	<ol style="list-style-type: none">1. Ingresar a la ventana de registro de un nuevo usuario.2. Obtener una imagen: Seleccionar en el menú <i>Archivo</i>, la opción <i>Obtener Imagen -> Carpeta</i>.3. Luego de cargada la imagen pulsar el botón <i>Analizar</i>.4. Confirmar agregar la imagen del rostro localizado al registro del nuevo usuario.5. Ingresar el nombre, apellido, clave y género del nuevo usuario.6. Pulsar el botón <i>Registrar Usuario</i>

Tabla 23.1. Descripción del Caso de Prueba *Registrar Usuario empleando una imagen existente*.

CP - 033		ESCENARIO : Flujo básico (capturar imagen)
Objetivo	Registrar correctamente un nuevo usuario empleando una imagen adquirida a través de una cámara web.	
Entrada	Nombre = Juan Apellido = Alcuacer Clave = 1002115801 Género = Masculino Imágenes del Rostro = 1 imagen (cámara web).	
Salida Esperada	Usuario registrado (información almacenada)	
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas, ni la utilización de la cámara web por otras aplicaciones.	
Procedimiento	<ol style="list-style-type: none"> 1. Ingresar a la ventana de registro de un nuevo usuario. 2. Obtener una imagen: Seleccionar en el menú <i>Archivo</i>, la opción <i>Obtener Imagen -> Video</i>. 3. Una vez que ha sido detectada la cámara web, confirmar la utilización de ésta para capturar una imagen pulsando el botón <i>Aceptar</i> del cuadro de diálogo desplegado. 4. Luego de cargada la imagen pulsar el botón <i>Analizar</i>. 5. Confirmar agregar la imagen del rostro localizado al registro del nuevo usuario. 6. Ingresar el nombre, apellido, clave y género del nuevo usuario. 7. Pulsar el botón <i>Registrar Usuario</i> 	

Tabla 23.2. Descripción del Caso de Prueba *Registrar Usuario – capturar imagen*.

CP - 034		ESCENARIO : Extensión (No se localizó ningún rostro en la imagen)
Objetivo	Intentar registrar un usuario empleando una imagen que no contiene un rostro humano.	
Entrada	Ruta = C:\casa.bmp	
Salida Esperada	Mensaje de error. No se localizó un rostro en la imagen. Botón <i>Registrar Usuario</i> deshabilitado.	

Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.
Procedimiento	<ol style="list-style-type: none"> 1. Ingresar a la ventana de registro de un nuevo usuario. 2. Obtener una imagen. Seleccionar en el menú <i>Archivo</i>, la opción <i>Obtener Imagen -> Carpeta</i>. 3. En el cuadro de diálogo ingresar C:\casa.bmp y pulsar el botón <i>Abrir</i>. 4. Una vez cargada la imagen pulsar el botón <i>Analizar</i>.

Tabla 23.3. Descripción del Caso de Prueba *Registrar Usuario – No se localizó ningún rostro en la imagen.*

23.2 CASO DE USO VERIFICAR IDENTIDAD

CP - 035		ESCENARIO : Flujo básico (imagen existente)	
Objetivo	Verificar la identidad de un usuario empleando una imagen obtenida desde una unidad de almacenamiento y una clave válida.		
Entrada	Imagen = C:\rostro_j.bmp Clave = 1002115801		
Salida Esperada	Mensaje : Acceso Concedido. Se presenta una ventana con toda la información almacenada del usuario autenticado.		
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.		

Procedimiento	<ol style="list-style-type: none"> 1. Ingresar a la ventana de verificación de identidad. 2. Obtener una imagen: Seleccionar en el menú <i>Archivo</i>, la opción <i>Obtener Imagen -> Carpeta</i>. 3. En el cuadro de diálogo ingresar C:\rostro_j.bmp y pulsar el botón <i>Abrir</i>. 4. Una vez cargada la imagen pulsar el botón <i>Analizar</i>. 5. Confirmar emplear la imagen del rostro localizado para la verificación de identidad. 6. Cuando se habilita el cuadro de texto para la clave de usuario, ingresar: 1002115801 7. Pulsar el botón <i>Verificar Identidad</i>
---------------	---

Tabla 23.4. Descripción del Caso de Prueba *Verificar Identidad empleando una imagen existente*.

CP - 036		ESCENARIO : Usuario no válido
Objetivo	Intentar verificar la identidad de un usuario empleando una imagen obtenida desde una unidad de almacenamiento y una clave no registrada en la base de datos (clave no válida).	
Entrada	Imagen = C:\rostro_j.bmp Clave = 8002415801	
Salida Esperada	Mensaje: Acceso Denegado.	
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.	
Procedimiento	<ol style="list-style-type: none"> 1. Ingresar a la ventana de verificación de identidad. 2. Obtener una imagen: Seleccionar en el menú <i>Archivo</i>, la opción <i>Obtener Imagen -> Carpeta</i>. 3. En el cuadro de diálogo ingresar C:\rostro_j.bmp y pulsar el botón <i>Abrir</i>. 4. Una vez cargada la imagen pulsar el botón <i>Analizar</i>. 5. Confirmar emplear la imagen del rostro localizado para la verificación de identidad. 6. Cuando se habilita el cuadro de texto para la clave de usuario, ingresar: 8002415801 7. Pulsar el botón <i>Verificar Identidad</i> 	

Tabla 23.5. Descripción del Caso de Prueba *Usuario no válido*.

CP - 037		ESCENARIO : Flujo básico (capturar imagen)	
Objetivo	Verificar la identidad de un usuario empleando una imagen adquirida a través de una cámara web, y una clave válida.		
Entrada	Imagen = cuadro de video Clave = 1002115801		
Salida Esperada	Mensaje: Acceso Concedido. Se presenta una ventana con toda la información almacenada del usuario autenticado.		
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.		
Procedimiento	<ol style="list-style-type: none"> 1. Ingresar a la ventana de verificación de identidad. 2. Obtener una imagen. Seleccionar en el menú <i>Archivo</i>, la opción <i>Obtener Imagen -> Video</i>. 3. Una vez que ha sido detectada la cámara web, confirmar la utilización de ésta para capturar una imagen pulsando el botón <i>Aceptar</i> del cuadro de diálogo desplegado. 4. Luego de cargada la imagen pulsar el botón <i>Analizar</i>. 5. Confirmar emplear la imagen del rostro localizado para la verificación de identidad. 6. Cuando se habilita el cuadro de texto para la clave de usuario, ingresar: 1002115801 7. Pulsar el botón <i>Verificar Identidad</i> 		

Tabla 23.6. Descripción del Caso de Prueba *Verificar Identidad – capturar imagen*.

CP - 038		ESCENARIO : Extensión (No se localizó ningún rostro en la imagen)	
Objetivo	Intentar verificar la identidad de un usuario empleando una imagen que no contiene un rostro humano.		
Entrada	Ruta = C:\casa.bmp		
Salida Esperada	Mensaje de error: No se localizó un rostro en la imagen. Botón <i>Verificar Identidad</i> deshabilitado.		
Condiciones	---		

Procedimiento	<ol style="list-style-type: none"> 1. Ingresar a la ventana de verificación de identidad. 2. Obtener una imagen: Seleccionar en el menú <i>Archivo</i>, la opción <i>Obtener Imagen -> Carpeta</i>. 3. En el cuadro de diálogo ingresar C:\casa.bmp y pulsar el botón <i>Abrir</i> 4. Una vez cargada la imagen pulsar el botón <i>Analizar</i>.
---------------	---

Tabla 23.7. Descripción del Caso de Prueba *Verificar Identidad – No se localizó ningún rostro en la imagen*

23.3 CASO DE USO OBTENER IMAGEN

CP - 039		ESCENARIO : Extensión (No se ha detectado un dispositivo)
Objetivo	Intentar verificar la identidad de un usuario tratando de emplear una imagen adquirida a través de una cámara web que está siendo utilizada por otra aplicación.	
Entrada	---	
Salida Esperada	Mensaje de error: No se ha detectado dispositivo.	
Condiciones	---	
Procedimiento	<ol style="list-style-type: none"> 1. Ingresar a la ventana de verificación de identidad. 2. Obtener una imagen: Seleccionar en el menú <i>Archivo</i>, la opción <i>Obtener Imagen -> Video</i>. 	

Tabla 23.8. Descripción del Caso de Prueba *No se ha detectado un dispositivo de captura*

23.4 CASO DE USO COMPROBAR PERFIL PARA ACCESO

CP - 040		ESCENARIO : Flujo básico (iniciado por el Administrador)
Objetivo	Acceder al menú de administración de la solución computacional.	

Entrada	---
Salida Esperada	Se presenta la ventana del menú de administración.
Condiciones	La sesión bajo la que se está ejecutando la solución computacional pertenece a un usuario con perfil de administrador del equipo o dominio, de acuerdo a la autenticación Windows.
Procedimiento	1. Solicitar el menú de administración.

Tabla 23.9. Descripción del Caso de Prueba *Ingreso al menú de administración*

CP - 041		ESCENARIO : Flujo básico (el usuario no cuenta con el perfil adecuado)	
Objetivo	Intentar acceder al menú de administración de la solución computacional sin contar con el perfil de usuario adecuado.		
Entrada	---		
Salida Esperada	Acceso denegado.		
Condiciones	La sesión bajo la que se está ejecutando la solución computacional no pertenece a un usuario con perfil de administrador del equipo o dominio, de acuerdo a la autenticación Windows.		
Procedimiento	1. Solicitar el menú de administración.		

Tabla 23.10. Descripción del Caso de Prueba *Usuario con perfil no valido*

23.5 CASO DE USO ACTUALIZAR PARÁMETROS DE LA RED NEURONAL

CP - 042		ESCENARIO : Flujo básico	
Objetivo	Actualizar exitosamente los valores de los parámetros de la red neuronal.		
Entrada	Num_neur = 4 Goal = 0.7		
Salida Esperada	Se han actualizado los valores.		
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.		

Procedimiento	<ol style="list-style-type: none"> 1. Ingresar a la ventana de preferencias. 2. Cuando aparezca esta ventada cambiar los valores de los parámetros a: Número de neuronas = 4 y goal = 0.7 3. Presionar el botón <i>Salir</i>
---------------	---

Tabla 23.11. Descripción del Caso de Prueba *Actualizar exitosamente los valores de los parámetros de la red neuronal.*

CP - 043		ESCENARIO : Extensión	
Objetivo	Intentar actualizar los parámetros de la red neuronal con valores fuera de los rangos permitidos.		
Entrada	Num_neur = 50	Goal = 0.7	
Salida Esperada	Mensaje de error: Verificar valores ingresados.		
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.		
Procedimiento	<ol style="list-style-type: none"> 1. Ingresar a la ventana de preferencias. 2. Cuando aparezca esta ventada cambiar los valores de los parámetros a: Número de neuronas = 50 y goal = 0.7 3. Presionar el botón <i>Salir</i> 		

Tabla 23.12. Descripción del Caso de Prueba *Valores erróneos de los parámetros de la red neuronal*

23.6 CASO DE USO GUARDAR IMAGEN INICIAL

CP - 044		ESCENARIO : Flujo básico	
Objetivo	Guardar una copia de la imagen obtenida desde una unidad de almacenamiento y que es empleada durante el registro de un nuevo usuario.		
Entrada	Imagen original = C:\rostro_j.bmp Nombre copia = copia_imagen Ubicación copia = C:\ Formato copia = .jpg		
Salida Esperada	Se ha creado una copia de la imagen inicial en la ubicación especificada.		

Condiciones	---
Procedimiento	<ol style="list-style-type: none"> 1. En la ventana de registro de un nuevo usuario, seleccionar en el menú <i>Archivo</i> la opción <i>Guardar Imagen -> Imagen Original</i>. 2. En el cuadro de diálogo que se despliega ingresar el nombre, ubicación y formato de la copia a crear. 3. Presionar el botón <i>Guardar</i>

Tabla 23.13. Descripción del Caso de Prueba *Guardar Imagen Inicial*

23.7 CASO DE USO GUARDAR IMAGEN ROSTRO LOCALIZADO

CP - 045		ESCENARIO : Flujo básico	
Objetivo	Guardar una copia de la imagen que muestra el rostro localizado.		
Entrada	Imagen original = frame capturado Nombre copia = copia_rostro Ubicación copia = C:\ Formato copia = .jpg		
Salida Esperada	Se ha creado una copia de la imagen con el rostro localizado en la ubicación especificada.		
Condiciones	---		
Procedimiento	<ol style="list-style-type: none"> 1. En la ventana de registro de un nuevo usuario, seleccionar en el menú <i>Archivo</i> la opción <i>Guardar Imagen -> Rostro Detectado</i>. 2. En el cuadro de diálogo que se despliega ingresar el nombre, ubicación y formato de la copia a crear. 3. Presionar el botón <i>Guardar</i> 		

Tabla 23.14. Descripción del Caso de Prueba *Guardar Imagen Rostro Localizado*

23.8 CASO DE USO ACTUALIZAR CLAVES DE USUARIO

CP - 046		ESCENARIO : Flujo básico
Objetivo	Actualizar exitosamente la clave de un usuario previamente registrado.	
Entrada	Nombre = Yolanda Apellido = Alcuacer Clave Actual = 1002116901 Clave Nueva = 58796 Confirmación Clave = 58796	
Salida Esperada	Se ha actualizado la clave	
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.	
Procedimiento	<ol style="list-style-type: none"> 1. Iniciar el módulo de administración de la solución computacional. 2. Cuando aparezca la ventana del menú de administración presionar el botón <i>Actualizar Clave</i>. 3. En la ventana de Usuarios Registrados, seleccionar de la lista de usuarios a <i>Yolanda Alcuacer</i>. 4. Una vez que aparezca la ventana para establecer la nueva clave ingresar: 58796 y confirmar: 58796. 5. Presionar el botón <i>Aceptar</i>. 	

Tabla 23.15. Descripción del Caso de Prueba *Actualización exitosa de la clave de usuario*

CP - 047		ESCENARIO : Extensión
Objetivo	Intentar actualizar erróneamente la clave de un usuario. (clave nueva y confirmación no coinciden)	
Entrada	Nombre = Yolanda Apellido = Alcuacer Clave Actual = 1002116901 Clave Nueva = 58796 Confirmación Clave = 58769	
Salida Esperada	Mensaje: Error en la clave nueva o en la confirmación (no coinciden).	
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.	

Procedimiento	<ol style="list-style-type: none"> 1. Iniciar el módulo de administración de la solución computacional. 2. Cuando aparezca la ventana del menú de administración presionar el botón <i>Actualizar Clave</i>. 3. En la ventana de Usuarios Registrados, seleccionar de la lista de usuarios a <i>Yolanda Alcuacer</i> 4. Una vez que aparezca la ventana para establecer la nueva clave ingresar: 58796 y confirmar: 58769. 5. Presionar el botón <i>Aceptar</i>
---------------	--

Tabla 23.16. Descripción del Caso de Prueba *Actualización errónea de la clave de usuario (clave nueva y confirmación no coinciden)*

CP - 048		ESCENARIO : Extensión
Objetivo	Intentar actualizar erróneamente la clave de un usuario (formato incorrecto de la nueva clave).	
Entrada	Nombre = Yolanda Apellido = Alcuacer Clave Actual = 1002116901 Clave Nueva = 5f8796 Confirmación Clave = 5f8796	
Salida Esperada	Mensaje de error: Verificar el formato de la nueva clave.	
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.	
Procedimiento	<ol style="list-style-type: none"> 1. Iniciar el módulo de administración de la solución computacional. 2. Cuando aparezca la ventana del menú de administración presionar el botón <i>Actualizar Clave</i>. 3. En la ventana de Usuarios Registrados, seleccionar de la lista de usuarios a <i>Yolanda Alcuacer</i>. 4. Una vez que aparezca la ventana para establecer la nueva clave ingresar: 5f8796 y confirmar: 5f8796. 5. Presionar el botón <i>Aceptar</i> 	

Tabla 23.17. Descripción del Caso de Prueba *Actualización errónea de la clave de usuario (error en formato)*

23.9 CASO DE USO GENERAR REPORTE USUARIOS REGISTRADOS

CP - 049		ESCENARIO : Flujo básico	
Objetivo	Generar el reporte de usuarios registrados.		
Entrada	---		
Salida Esperada	Reporte de usuarios registrados.		
Condiciones	No se permite el acceso a la base de datos por parte de otras aplicaciones mientras se realizan las pruebas.		
Procedimiento	<ol style="list-style-type: none">1. Iniciar el módulo de administración de la solución computacional.2. Cuando aparezca la ventana del menú de administración presionar el botón <i>Ver Reporte</i>.		

Tabla 23.18. Descripción del Caso de Prueba *Generar Reporte Usuarios Registrados*

Anexo 24

EVALUACIÓN DE LA SOLUCIÓN COMPUTACIONAL CONDICIONES

24.1 CONDICIONES

En esta sección se describen las condiciones bajo las cuales se llevó a cabo la evaluación, así como también los materiales empleados.

24.1.1 Base de rostros FENIXVR

Integrada por 379 imágenes pertenecientes a 25 personas que colaboraron para la elaboración de esta base de rostros, siendo el 88% mujeres y el 12% hombres. Las edades de quienes conforman este grupo se encuentran en el rango de 20 a 50 años, existiendo una prevalencia de personas cuya edad oscila entre 20 a 38 años.

Las imágenes fueron almacenadas en formato de mapa de bits (bmp), con dimensiones 320x240. Cada archivo gráfico tiene un tamaño de 226KB.

Todas las imágenes contienen el rostro de una sola persona, presentan fondos poco complejos, el área del rostro es claramente visible, presentan pequeñas variaciones con relación a la distancia entre una persona y la cámara, leves variaciones en el grado de inclinación del rostro, cambios de estilos de peinado, uso de lentes y variaciones en expresiones faciales. En general cumplen las restricciones establecidas para la solución

computacional descritas en detalle en la sección 2.1.5.5 del presente trabajo.

(*) La creación de la base de rostros se llevó a cabo como parte del proceso de evaluación de la solución computacional (durante la evaluación del subsistema de localización de rostros).

24.1.2 Base de rostros disponible a través de Internet

Parte de esta base fue empleada para la evaluación del subsistema de localización de rostros.

Contiene imágenes de 114 personas, 7 imágenes por cada persona, con diferentes expresiones faciales.

El formato de las imágenes es jpeg y tienen una resolución de 640x480 píxeles.

Las personas que conforman esta base en su mayoría son hombres (alrededor del 90%) que oscilan entre los 18 años (aunque existen personas de mayor edad).

24.1.3 Ambiente de prueba

La adquisición de las imágenes para la base de rostros FENIXVR se realizó durante tres días, tratando de mantener condiciones similares de iluminación en el transcurso de todo el proceso de adquisición. Sin embargo la limitación de infraestructura adecuada para obtener imágenes con iluminación normal y constante, así como la limitación en el tiempo, dio como resultado ciertas imágenes en las que la variación de iluminación no es irrelevante (esto significa que influye en la verificación de identidad), factor que se debió considerar durante la evaluación de los resultados.

Se debe mencionar también que si bien el proceso de adquisición duró tres días las imágenes capturadas por cada usuario corresponden a un mismo día debido a que fueron tomadas en secuencia, a excepción de algunas imágenes que fueron tomadas en días diferentes, las mismas que se especificarán más adelante (véase Tabla 24.1).

Adicionalmente se logró mantener dentro de los límites permitidos la distancia entre la cámara web empleada y el rostro de las personas. A cada persona se solicitó ver de frente a la cámara y a partir de una cara neutra, hacer diversas expresiones faciales, como mostrar seriedad, sonreír, reírse, y finalmente hablar (véase Figura 24.1). Se solicitó repetir este conjunto de expresiones hasta obtener una secuencia de imágenes por cada persona; también se solicitó mostrar variaciones en el estilo del peinado, cuando fuera posible (véase Figura 24.2).



Figura 24.1. Imágenes que forman el conjunto de prueba para una persona determinada



Figura 24.2. Imágenes que muestran a una persona con diferentes estilos de peinado

El conjunto de imágenes obtenido por cada persona se dividió en dos grupos: las imágenes del conjunto de entrenamiento y las del conjunto de prueba.

El conjunto de entrenamiento se formó con cinco imágenes de cada persona registrada como usuario de la solución computacional, cada imagen en este conjunto fue etiquetada con un número asignado a la persona seguido del número de imagen y de un sufijo que indica la expresión facial captada, por ejemplo 1_1N.bmp se refiere a la primera imagen de la persona 1 con la expresión facial neutra.

En tanto que el conjunto de prueba se formó con las imágenes restantes de todas las personas, a su vez estas imágenes se etiquetaron con un número asignado a cada persona seguido del número de imagen, así 1_1.bmp se refiere a la primera imagen de prueba de la persona 1.

Finalmente se seleccionaron aleatoriamente tres personas que no serían registradas como usuarios y que servirían para evaluar el desempeño de la solución al rechazar personas desconocidas (usuarios no registrados).

Número de personas	Imágenes de entrenamiento	Imágenes de prueba
1	5	9
2 (*)	5	12
3	5	8
4	5	5
5	0	14
6	5	10
7	5	9
8 (*)	5	17
9	0	2
10	5	12
11	5	10
12	0	13
13 (*)	5	13
14	5	14
15	5	13
16	5	8

17	5	13
18	5	14
19	5	10
20	5	11
21	5	7
22	5	10
23	5	10
24	5	15
25	5	10
Total Imágenes	110	269
	379	

(*) Algunas imágenes del conjunto de prueba fueron adquiridas durante días diferentes al que se adquirió las imágenes de entrenamiento.

Tabla 24.1. División de la base de rostros FENIXVR en conjunto de entrenamiento y conjunto de prueba

24.1.4 Configuración

Para la evaluación de la solución computacional se emplearon las siguientes configuraciones de hardware y software:

Hardware

Procesador: Pentium IV de 2.0 Ghz

Memoria RAM: 512 MB de memoria de acceso aleatorio

Disco Duro: 80 GB

Monitor: LCD de resolución 1280x1024, color a 32 bits de profundidad

Para la adquisición de las imágenes se utilizó una cámara web USB marca *Creative* cuyas características principales se detallan a continuación:

- Sensor CMOS (VGA) 640 x 480
- Captura de imagen fija de hasta 1024x768

- Anillo de foco manual
- Conexión USB 1.1

Software

Sistema Operativo: Microsoft Windows XP Profesional versión 2002 SP1.

Base de datos: Microsoft Access 2000

Entorno: Motor y toolboxes de Matlab 7.0