

# **INSTITUTO POLITÉCNICO NACIONAL**

---



## **CENTRO DE INVESTIGACIÓN EN COMPUTACIÓN**

### **Reconocimiento de personas mediante un sistema multibiométrico**

## **T E S I S**

**QUE PARA OBTENER EL GRADO DE  
MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

**PRESENTA:**

**ING. GABRIEL BALTAZAR PÉREZ**

**DIRECTORES DE TESIS:**

**DR. CORNELIO YÁÑEZ MÁRQUEZ  
DR. ITZAMÁ LÓPEZ YÁÑEZ**



**MÉXICO, D.F.**

**DICIEMBRE DE 2015**



# INSTITUTO POLITÉCNICO NACIONAL SECRETARÍA DE INVESTIGACIÓN Y POSGRADO

## ACTA DE REVISIÓN DE TESIS

En la Ciudad de México, D.F. siendo las 10:00 horas del día 03 del mes de diciembre de 2015 se reunieron los miembros de la Comisión Revisora de la Tesis, designada por el Colegio de Profesores de Estudios de Posgrado e Investigación del:

**Centro de Investigación en Computación**

para examinar la tesis titulada:

**“Reconocimiento de personas mediante un sistema multibiométrico”**

Presentada por el alumno:

**BALTAZAR**

Apellido paterno

**PÉREZ**

Apellido materno

**GABRIEL**

Nombre(s)

Con registro: 

B	1	3	0	0	6	1
---	---	---	---	---	---	---

aspirante de: **MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**

Después de intercambiar opiniones los miembros de la Comisión manifestaron **APROBAR LA TESIS**, en virtud de que satisface los requisitos señalados por las disposiciones reglamentarias vigentes.

### LA COMISIÓN REVISORA

Directores de Tesis

  
Dr. Cornelio Yáñez Márquez

  
Dr. Itzamá López Yáñez


  
Dr. Olexsiy Pogrebnyak

  
Dr. Marco Antonio Moreno Ibarra

  
Dra. GuoHua Sun

  
Dr. Amadeo José Argüelles Cruz

PRESIDENTE DEL COLEGIO DE PROFESORES

  
Dr. Luis Alfonso Villa Vargas





**INSTITUTO POLITÉCNICO NACIONAL**  
**SECRETARÍA DE INVESTIGACIÓN Y POSGRADO**

*CARTA CESIÓN DE DERECHOS*

En la Ciudad de México D.F., el día 9 del mes Diciembre del año 2015, el (la) que suscribe Gabriel Baltazar Pérez alumno (a) del Programa de Maestría en Ciencias de la Computación con número de registro B130061, adscrito a Centro de Investigación en Computación, manifiesta que es autor (a) intelectual del presente trabajo de Tesis bajo la dirección de Dr. Cornelio Yáñez Márquez y Dr. Itzamá López Yáñez y cede los derechos del trabajo intitulado Reconocimiento de personas mediante un sistema multibiométrico, al Instituto Politécnico Nacional para su difusión, con fines académicos y de investigación.

Los usuarios de la información no deben reproducir el contenido textual, gráficas o datos del trabajo sin el permiso expreso del autor y/o director del trabajo. Este puede ser obtenido escribiendo a la siguiente dirección gbp1000@gmail.com. Si el permiso se otorga, el usuario deberá dar el agradecimiento correspondiente y citar la fuente del mismo.

  
Gabriel Baltazar Pérez

Nombre y firma

## Resumen

En el presente trabajo de tesis se desarrolla e implementa un sistema multibiométrico para la identificación automática de personas, utilizando técnicas de procesamiento digital de imágenes y reconocimiento de patrones. Los algoritmos de clasificación utilizados para la identificación son las *Alpha-Beta SVM* y algoritmos de *Pattern Matching*.

Se realiza la identificación de personas desconocidas mediante el reconocimiento de iris y huella dactilar como multibiométrico. Se utilizaron bancos de datos de imágenes de ojos y huellas dactilares pertenecientes a CASIA, en las que se aplicaron métodos para mejorar la calidad de las imágenes, extracción de características y tareas de clasificación.

El sistema multibiométrico pretende servir como una herramienta útil en la identificación automática de personas, que sea accesible desde la web o desde algún dispositivo móvil. Los casos particulares donde solo se cuente con un biométrico y la inclusión de más biométricos, son considerados con la finalidad de tener un sistema más robusto para lograr una alta tasa de verdaderos positivos en la identificación de personas.

### **Abstract**

In the current document of thesis is developed and implemented a multibiometric system for an automatic identification of people using digital image processing and pattern recognition algorithms. The classification algorithms used for identification are the Alfa-Beta SVM Associative Machines and Matching of Patterns.

Unknown people identification is performed by iris and fingerprint recognition as multibiometric. The CASIA databases of iris and fingerprints images are used to apply enhancement quality methods, feature extraction and classification tasks.

The multibiometric system is pretended as a tool for an automatic identification of people that is accessible from the web or some mobile device. Cases where only one biometric is available and the inclusion of more biometrics it is considered with the finality to have a robust system that provide a high rate of true positives in the people identification.

## **Agradecimientos**

*A mis padres.*

*Artemia y Librado por haberme apoyado en todo momento, por su perseverancia, sus valores, por la motivación constante que me ha permitido ser una persona de bien, pero más que nada, por ser el pilar fundamental de lo que soy y de su amor incondicional.*

*A mis hermanos.*

*Miguel Ángel y Alberto por ser un ejemplo y de los cuales he aprendido aciertos que me han ayudado en mi vida diaria.*

*A mi novia.*

*Xóchitl por estar conmigo y apoyarme siempre durante esta etapa de mi formación profesional.*

*A mis directores.*

*Dr. Cornelio Yáñez Márquez Y Dr. Itzamá López Yáñez por su gran apoyo y motivación para la culminación de mis estudios y para la elaboración de esta tesis de maestría, así como impulsar el desarrollo de mi formación profesional.*

*A mis amigos.*

*Sergio, Enrique y Alberto, por ser grandes amigos y compartir los buenos y malos momentos conmigo. Y finalmente, pero no menos importante a Adrián, Pablo, Massiel y Arturo a quienes conocí recientemente, pero que me han brindado un gran apoyo para la culminación de mis estudios de maestría.*

*Todo este trabajo ha sido posible gracias a ellos.*

# Índice

1. Introducción .....	13
1.1 Antecedentes .....	13
1.2 Justificación.....	17
1.3 Objetivo General .....	17
1.4 Objetivos específicos.....	17
1.5 Contribuciones .....	18
1.6 Organización del Documento .....	18
2. Estado del Arte .....	19
2.1 Huella dactilar .....	19
2.2 Iris .....	21
2.3 Sistemas multibiométricos.....	24
2.4 Máquinas Asociativas Alfa-Beta SVM.....	24
2.5 Clasificador Gamma.....	25
2.6 Aplicaciones Web y Móviles.....	25
3. Materiales y Métodos .....	26
3.1 Procesamiento Digital de Imágenes.....	26
3.2 Morfología Matemática .....	26
3.3 Procesamiento Digital de imágenes de Huella Dactilar .....	28
3.3.1 Mejora de la calidad .....	29
3.3.2 Extracción de características en Imágenes de Huellas Dactilares .....	31
3.4 Procesamiento Digital de Imágenes de Iris.....	39
3.4.1 Mejora de la calidad .....	39
3.4.2 Generación del código del iris .....	42
3.5 Máquinas Asociativas Alfa-Beta SVM.....	43
3.5.1 Memorias Asociativas .....	43
3.5.2 Operadores Alfa y Beta .....	44
3.5.3 Support Vector Machines (SVMs) .....	44
3.5.4 Máquinas Asociativas Alfa-Beta SVM.....	46
3.6 Clasificador Gamma.....	48
3.6.1 Operador $\mathcal{U}\beta$ .....	48
3.6.2 Operación módulo .....	48

3.6.3 Código binario Johnson-Möbius modificado .....	49
3.6.4 Clasificador Gamma .....	49
3.7 MVC .....	53
3.8 Base de Datos NoSQL.....	54
4. Solución Propuesta.....	55
4.1 Procesamiento digital y generación de patrones a partir de imágenes de Iris y Huella Dactilar .....	55
4.1.1 Iris .....	55
4.1.2 Huella Dactilar .....	61
4.2 Algoritmos de Clasificación para reconocimiento de Iris y Huella Dactilar .....	64
4.2.1 Iris .....	64
4.2.2 Huella Dactilar .....	68
4.3 Aplicación Web.....	69
4.3.1 Modelo .....	70
4.3.2 Vista .....	71
4.3.3 Controlador .....	71
4.4 Aplicación Móvil.....	71
4.5 Arquitectura de las aplicaciones para el sistema multibiométrico .....	72
4.6 Proceso completo .....	73
5. Resultados y Discusión .....	82
5.1 Bancos de Datos .....	82
5.2 Entorno de Prueba .....	82
5.3 Resultados experimentales .....	83
6. Conclusiones y Trabajo a Futuro.....	88
6.1 Conclusiones .....	88
6.2 Trabajo a Futuro .....	89
Referencias.....	90



## Índice de figuras

<i>Figura 1 Modelo genérico para la identificación o verificación de personas en un sistema biométrico .....</i>	<i>14</i>
<i>Figura 2 Tipos de Minucia (Cresta corta, cresta de terminación, bifurcación y espuela).....</i>	<i>20</i>
<i>Figura 3 Imagen del ojo que especifica algunas características útiles en el reconocimiento de iris. ....</i>	<i>22</i>
<i>Figura 4 Imagen del ojo que muestra algunos procesos en el reconocimiento de iris. ....</i>	<i>22</i>
<i>Figura 5 Ventanas utilizadas en el método de Sobel para la obtención de los gradientes horizontal <math>\Delta x</math> y vertical <math>\Delta y</math> .....</i>	<i>32</i>
<i>Figura 6 Ventana de 3x3 donde P1 es el pixel central. ....</i>	<i>34</i>
<i>Figura 7 Ventana de 3x3 usada para encontrar minucias. ....</i>	<i>35</i>
<i>Figura 8 a) Bloque sin minucias, b) Bloque con bifurcación y c) Bloque con terminación.....</i>	<i>36</i>
<i>Figura 9 Tipos de estructuras de minucias falsas. De izquierda a derecha y de arriba hacia abajo tenemos: pico, puente, agujero, rotura, espuela y escalera.....</i>	<i>36</i>
<i>Figura 10 Ejemplo de validación de una minucia de tipo terminal. La figura muestra los cambios en la imagen L después de diferentes pasos del algoritmo.....</i>	<i>38</i>
<i>Figura 11 Ejemplo de validación de una minucia de tipo bifurcación. La figura muestra los cambios en la imagen L después de diferentes pasos del algoritmo.....</i>	<i>38</i>
<i>Figura 12 Conversión de coordenadas polares a cartesianas. ....</i>	<i>41</i>
<i>Figura 13 Memoria Asociativa .....</i>	<i>43</i>
<i>Figura 14 Support Vector Machine. Los vectores de soporte están indicados con un recuadro mientras que el hiperplano óptimo aparece como una línea punteada. ....</i>	<i>45</i>
<i>Figura 15 Ejemplo de función kernel.....</i>	<i>46</i>
<i>Figura 16 Diagrama general de la arquitectura MVC.....</i>	<i>53</i>
<i>Figura 17 Imágenes digitales de Iris pertenecientes a la base de datos CAISA-Iris-Syn. ....</i>	<i>56</i>
<i>Figura 18 Método de Sobel aplicado a las imágenes de la Figura 17. ....</i>	<i>57</i>

<i>Figura 19 Aplicación de la binarización mediante histéresis sobre las imágenes a) y b) de la Figura 18. ....</i>	<i>57</i>
<i>Figura 20 Aplicación de la binarización mediante histéresis sobre la imagen c) de la Figura 18. ...</i>	<i>58</i>
<i>Figura 21 Frontera pupilar y frontera límbica detectadas en las imágenes a) y b) de la Figura 17 por medio de la Transformada de Hough.....</i>	<i>58</i>
<i>Figura 22 Frontera pupilar y frontera límbica detectadas en la imagen c) de la Figura 17 por medio de la Transformada de Hough. ....</i>	<i>59</i>
<i>Figura 23 Imágenes de iris normalizados pertenecientes a las imágenes de la figura 17. ....</i>	<i>59</i>
<i>Figura 24 Aplicación del filtro Laplaciano del Gaussiano en las imágenes de iris normalizados de la Figura 23. ....</i>	<i>60</i>
<i>Figura 25 Códigos de iris generados a partir de las imágenes de la Figura 24. ....</i>	<i>60</i>
<i>Figura 26 Imágenes digitales de Huella Dactilar pertenecientes a la base de datos CAISA-Fingerprint. ....</i>	<i>61</i>
<i>Figura 27 Transformada de Fourier aplicada a las imágenes de la Figura F1. ....</i>	<i>61</i>
<i>Figura 28 Ecuilización del Histograma aplicado a las imágenes de la Figura 27. ....</i>	<i>62</i>
<i>Figura 29 Imagen que refleja la segmentación aplicada sobre las imágenes originales de la Figura F1, donde se puso en blanco las zonas que no pertenecen a la huella dactilar. ....</i>	<i>62</i>
<i>Figura 30 Aplicación de la binarización adaptativa a imágenes previamente mejoradas con la TDF y ecuilización del histograma mostradas en la Figura 29. ....</i>	<i>63</i>
<i>Figura 31 Esqueleto generado por medio del método Guo-Hall aplicado a las imágenes de la Figura 30. ....</i>	<i>63</i>
<i>Figura 32 Imágenes de huellas dactilares en las que se detectaron minucias de tipo bifurcación marcadas con puntos rojos y de tipo terminación marcadas con puntos azules. ....</i>	<i>64</i>
<i>Figura 33 Código de iris generado a partir del preprocesamiento de una imagen de un ojo. ....</i>	<i>65</i>
<i>Figura 34 Ejemplo de conversión de la imagen de un código de iris a un patrón numérico. ....</i>	<i>66</i>
<i>Figura 35 Arquitectura del servidor Web en la nube y las aplicaciones móviles para el sistema multibiométrico para el reconocimiento de personas .....</i>	<i>73</i>
<i>Figura 36 Diagrama 1 del subproceso de registro de plantillas biométricas .....</i>	<i>74</i>

<i>Figura 37 Diagrama 2 del subproceso de reconocimiento de características extraídas de una huella biométrica (Iris o Huella Dactilar).....</i>	<i>75</i>
<i>Figura 38 Diagrama 3 del subproceso de la toma de decisión para proporcionar un resultado al sistema multibiométrico.....</i>	<i>76</i>
<i>Figura 39 Diagrama 4 del proceso completo del modelo propuesto para el sistema multibiométrico para el reconocimiento de personas.....</i>	<i>77</i>
<i>Figura 40 Pantalla de registro de información básica de un usuario.....</i>	<i>78</i>
<i>Figura 41 Pantalla en la que se capturan las imágenes de iris del usuario que se está registrando.</i>	<i>79</i>
<i>Figura 42 Pantalla en la que se capturan las imágenes de huella dactilar del usuario que se está registrando.....</i>	<i>79</i>
<i>Figura 43 Pantalla que muestra un registro exitoso.....</i>	<i>80</i>
<i>Figura 44 Pantalla de inicio de sesión y menú de opciones para la verificación .....</i>	<i>80</i>
<i>Figura 45 Pantalla que muestra el proceso de verificación por medio del iris en un dispositivo móvil .....</i>	<i>81</i>
<i>Figura 46 Pantalla que muestra el proceso de verificación por medio de la huella dactilar en un dispositivo móvil.....</i>	<i>81</i>

## Índice de tablas

Tabla 1 Siete pilares de un biométrico .....	13
Tabla 2 Definición del operador Alfa .....	44
Tabla 3 Definición del operador Beta.....	44
Tabla 4 Escenarios de operación de un sistema biométrico.....	83
Tabla 5 Resultados experimentales de 10 experimentos con datos de iris, con posibles clases desconocidas (verdaderos negativos) .....	85
Tabla 6 Resultados experimentales de 10 experimentos con datos de huellas dactilares, con posibles clases desconocidas (verdaderos negativos) .....	85
Tabla 7 Resultados experimentales de 10 experimentos con datos de iris y huellas dactilares, con posibles clases desconocidas (verdaderos negativos) .....	86
Tabla 8 Resultados experimentales de 10 experimentos con datos de iris y huellas dactilares, con posibles clases desconocidas (verdaderos negativos) .....	87

## Capítulo 1

### 1. Introducción

En el presente trabajo de tesis se desarrolla e implementa un sistema multibiométrico para la identificación automática de personas, utilizando técnicas de procesamiento digital de imágenes y de reconocimiento de patrones, además de algoritmos de clasificación como las *Alpha-Beta SVM* y ciertos algoritmos de *Pattern Matching*.

#### 1.1 Antecedentes

El término biometría esta derivado de las palabras griegas bio (vida) y metria (medida). De manera general, se entiende por biometría como el conjunto de métodos automatizados basados en determinadas características humanas para la identificación o verificación de personas [1]. La biometría aprovecha la existencia de características biológicas o conductuales que son singulares e inalterables, por lo que pueden ser analizadas y medidas para crear una huella biométrica [2].

Dichas características son difíciles de perder, transferir u olvidar y además son perdurables a través del tiempo. La Tabla 1 es adoptada del trabajo de *Jain et al.* [3], donde se muestran características particulares del cuerpo que deben ser consideradas para que un biométrico sea adecuado:

*Tabla 1 Siete pilares de un biométrico*

<b>Universalidad</b>	Un biométrico debe ser común de encontrar en todos los seres humanos, tales como dedos, iris, cara, ADN, de tal manera que puedan ser utilizados para reconocerlos.
<b>Singularidad</b>	Un biométrico debe ser único y diferenciable entre una persona y otra.
<b>Permanencia</b>	Un biométrico debe perdurar sin cambios a lo largo de la vida de las personas.
<b>Recolectable</b>	Es la facilidad de adquirir, medir y almacenar una huella biométrica.
<b>Rendimiento</b>	Es el grado de precisión, velocidad y robustez; debe ser muy alto para que el sistema pueda ser operacional.
<b>Aceptabilidad</b>	Qué tanta aprobación tiene un biométrico entre el público.
<b>Resistencia a la elusión</b>	Qué tan fácil es engañar al sistema biométrico.

Existen muchas técnicas diferentes y disponibles para identificar y verificar personas basadas en el uso de biométricos. Estas técnicas se categorizan en dos grandes grupos que son [1]:

- **Fisiológicos o morfológicos:** Son aquellas que se apoyan en características físicas inalterables y presentes en la mayoría de los seres humanos, tales como huella dactilar, geometría de la mano, rostro, forma de la oreja, ADN, entre otras más.

- Conductuales: Son aquellas que se apoyan en características pertenecientes a la conducta del ser humano, tales como pulsaciones, voz, entre otros.

En la biometría hay términos de uso frecuente que son:

- Reconocimiento: Es un término genérico que no necesariamente implica una verificación o identificación de un individuo, puesto que todos los sistemas biométricos realizan reconocimiento para tratar de distinguir una persona que ha ingresado previamente al sistema.
- Verificación: Es la confirmación de la identidad de una persona, que se realiza mediante la comparación de una huella biométrica de la persona con una o más plantillas biométricas de dicha persona, las cuales previamente deben estar registradas en el sistema.
- Identificación: Consiste en determinar a quién corresponde una huella biométrica proporcionada, mediante su comparación con un conjunto de plantillas biométricas pertenecientes a diferentes personas previamente registradas en el sistema.

Un sistema biométrico se encarga de reconocer a una persona automáticamente por medio de sus características físicas o conductuales, con la finalidad de verificar o identificar a dicha persona. La Figura 1 muestra el diagrama de un modelo genérico para la identificación de personas mediante el uso de un sistema biométrico; ahí, se muestran las etapas principales que debe tener para tomar una decisión.

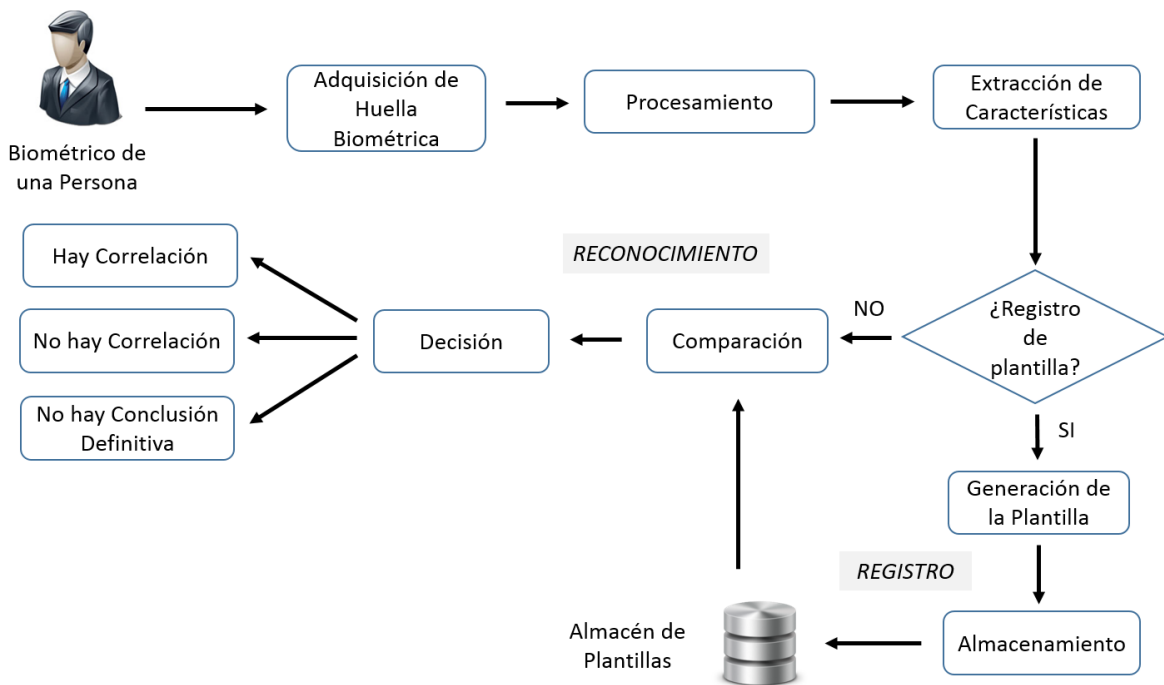


Figura 1 Modelo genérico para la identificación o verificación de personas en un sistema biométrico

Para tomar una decisión, el resultado de cualquiera de las comparaciones que se hagan puede presentar una de tres posibilidades dependiendo el valor que se obtenga en la comparación entre la huella biométrica y la plantilla, así como del umbral que se le haya dado al sistema; las tres posibles alternativas son:

1. Hay correlación: Al comparar la huella biométrica capturada con la(s) plantilla(s) almacenada(s) el valor obtenido se encuentra dentro de los umbrales de coincidencia.
2. No hay correlación: Al comparar la huella biométrica capturada con la(s) plantilla(s) almacenada(s) el valor obtenido esta fuera de los umbrales de coincidencia.
3. No hay conclusión definitiva: Hay falta de información para poder realizar una comparación adecuada.

El uso del concepto de biometría para la identificación o verificación de personas no es nuevo, ya que, a lo largo de la historia, civilizaciones como los antiguos egipcios, sumerios y otros, ciertamente ya practicaban estas técnicas; más recientemente, han ido surgiendo nuevas ideas con respecto a los identificación y verificación de personas [4]. Algunas veces el uso de biométricos ha tenido un mal desempeño, proporcionando falsos negativos al momento de verificar usuarios o identificar a una persona como otra que ciertamente no lo es. Esto de alguna manera ha despertado en la comunidad científica un interés por mejorar y generar nuevos modelos de reconocimiento de personas que permitan identificar o verificar de forma automática la identidad de las personas, haciendo uso de diferentes biométricos o combinación de los mismos, con la finalidad de obtener resultados que sean más confiables y certeros. Comúnmente, los sistemas de reconocimiento para la identificación o verificación de personas hacen uso de un solo biométrico; a continuación, se listan los más utilizados:

- Huella dactilar [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
- Iris [16, 17, 18, 19, 20, 21, 22, 23, 24, 25]
- Cara [26, 27, 28, 29, 30]
- Voz [31, 32, 33, 34]
- Mano [35, 36, 37, 38, 39]
- Venas [40, 41, 42, 43, 44]
- Firma [45, 46, 47, 48, 49]
- Oreja [50, 51, 52, 53, 54]

Sin embargo, recientemente se han desarrollado nuevos sistemas basados en modelos que hacen uso de más de un biométrico en la etapa de reconocimiento de personas, denominados de forma general como sistemas multibiométricos [1]. Hay autores que dicen que un sistema también puede ser considerado multibiométrico aun cuando solo haga uso de un solo biométrico, pero con la restricción de que en su implementación se debe considerar la aplicación de al menos dos modelos sobre el biométrico utilizado para tomar una decisión [4].

Actualmente existen varias técnicas y herramientas de utilidad para la identificación y verificación de personas, algunas de ellas pertenecientes al reconocimiento de patrones:

- *Alfa-Beta SVM* [55, 56, 57, 58]
- Clasificador Gamma [59, 60, 61, 62, 63, 64, 65, 66]
- Morfología Matemática [67, 68, 69, 70, 71, 72]
- Aplicaciones Web y Móviles [23, 10]

El reconocimiento de patrones es una herramienta poderosa utilizada para la tarea de clasificación en sistemas que hacen uso de biométricos, permitiendo al experto identificar o verificar la identidad de las personas. Las técnicas de clasificación utilizadas en el reconocimiento de personas pueden

ser automatizadas por medio de sistemas de cómputo: Dichas técnicas funcionan con una previa fase de entrenamiento, donde se ajusta el tipo de patrones a utilizar, los cuales servirán para actuar posteriormente sobre un conjunto de datos biométricos, los cuales son obtenidos de personas de quienes se desea identificar o verificar la identidad [3].

Cabe mencionar que existen otras tareas que pertenecen al reconocimiento de patrones, como la recuperación, regresión, clustering, sistemas de recomendación, entre otros; siendo la clasificación una tarea asociada en gran parte de la literatura al reconocimiento de patrones [65].

### **Morfología Matemática**

La morfología matemática es una herramienta muy poderosa que hace uso de distintas operaciones morfológicas aplicadas en el procesamiento digital de imágenes, entre las que destacan la segmentación de formas, determinación de bordes y generación de esqueletos [72].

La morfología matemática hace uso de dos operaciones principales sobre las que se derivan las demás, estas son: la dilatación y erosión. La primera operación permite reducir el tamaño de los objetos oscuros sobre fondo claro en una imagen digital y la segunda aumentar el tamaño de los objetos oscuros sobre fondo claro en una imagen digital [68]. Es común encontrar en los sistemas biométricos el uso de operaciones morfológicas útiles para incrementar la eficacia de los mismos.

### **Aplicaciones Web y Móviles**

Las aplicaciones web en la actualidad tienen un papel muy importante y tienen su origen en la web que trabajaba sobre el protocolo HTTP (HiperText Transfer Protocol), donde surgió la necesidad de desarrollar sistemas cada vez más complejos. Este es un protocolo creado por Tim Berners-Lee en la Organización Europea para la Investigación Nuclear en el año de 1989 y un estándar en la WWW (World Wide Web) [73, 74]. Con el tiempo el desarrollo de aplicaciones ha ido evolucionando a la par que lo hacen las nuevas tecnologías, permitiendo que se realicen aplicaciones más complejas y sean de mayor utilidad.

En la actualidad existen tecnologías web que permiten la implementación de algoritmos de reconocimiento de patrones y análisis de imágenes, a fin de crear sistemas de reconocimiento de personas para su identificación o verificación, según sea su uso. En este trabajo se propone la creación de un sistema multibiométrico sobre una aplicación web, que permita ofrecer los servicios de identificación y verificación de personas haciendo uso de las tecnologías actuales.

La aparición de las aplicaciones móviles es mucho más reciente que las aplicaciones web. Las aplicaciones móviles se iniciaron con aplicaciones simples como: una agenda, juegos de arcade, y pequeños editores, los cuales cumplían con funciones muy elementales y tenían un diseño bastante simple [23].

El avance tecnológico en el desarrollo de protocolos, ha permitido que las aplicaciones móviles y las aplicaciones web puedan comunicarse por distintos medios según las necesidades. Estas ventajas permiten que se puedan implementar algoritmos complejos de procesamiento digital de imágenes sobre plataformas móviles; teniendo en cuenta, las limitaciones de los recursos y capacidades con los que cuentan los dispositivos móviles actuales para correr una aplicación móvil [42].



## 1.2 Justificación

El reconocimiento de personas es muy importante en distintos lugares donde no se cuenta con la tecnología, infraestructura o profesionales con el conocimiento necesario para resolver dicha tarea. Factores importantes como el tiempo y la eficacia también son de suma importancia, por lo que un resultado preciso en el momento indicado es de gran utilidad como cuando se quiere identificar un criminal. Por lo que el sistema propuesto es una herramienta que permitirá aminorar, y en otros casos resolver, los problemas mencionados anteriormente.

En el estado del arte hay métodos de procesamiento digital de imágenes sobre biométricos que han demostrado tener buenos resultados [5-72]; sin embargo, el uso de memorias asociativas en el reconocimiento de personas no ha sido muy utilizado en la tarea de reconocimiento de personas, pero se ha demostrado que exhiben un buen rendimiento en distintas áreas del reconocimiento de patrones. Aquí se presentan métodos de procesamiento digital de imágenes y clasificadores basados en las memorias asociativas para el reconocimiento de personas.

## 1.3 Objetivo General

Desarrollar e implementar un sistema multibiométrico que permita el reconocimiento de personas, con el uso de métodos de procesamiento digital de imágenes y algoritmos de reconocimiento de patrones como las *Alpha-Beta SVM* y algoritmos de *Pattern Matching*, sobre una aplicación web y aplicaciones móviles.

## 1.4 Objetivos específicos

- Investigar en el estado del arte sobre los métodos de procesamiento digital de imágenes utilizados en el reconocimiento de personas mediante el iris o huella dactilar.
- Investigar en el estado del arte sobre los algoritmos de clasificación relacionados con el reconocimiento de personas con el uso de iris o huella dactilar.
- Desarrollar una aplicación web que permita la ejecución del proceso de reconocimiento de personas con el uso de dos biométricos, mediante los métodos de procesamiento digital de imágenes y algoritmos de clasificación propuestos.
- Desarrollar una aplicación móvil que permita ejecutar los métodos de procesamiento digital de imágenes para la obtención de patrones.
- Realizar un análisis sobre las ventajas de implementar el sistema web sobre una plataforma de cómputo en la nube.
- Realizar experimentos sobre un banco de datos para probar el sistema multibiométrico, los métodos de procesamiento digital de imágenes y los algoritmos de clasificación propuestos.
- Describir el método utilizado para tomar una decisión a partir de los resultados generados por el uso de más de un biométrico.

## **1.5 Contribuciones**

- Aplicación web que permite el reconocimiento de personas haciendo uso del iris o la huella dactilar.
- Aplicaciones móviles que permiten el reconocimiento de personas haciendo uso del iris o la huella dactilar.
- Proceso basado en los métodos de procesamiento digital de imágenes utilizados en el reconocimiento de personas mediante el iris o la huella dactilar.
- Modelo de la implementación del sistema en la nube.
- Algoritmo de clasificación para la toma de una decisión cuando se tiene más de un biométrico.

## **1.6 Organización del Documento**

En el presente capítulo se describieron los antecedentes, justificación, objetivos, contribuciones que este trabajo de tesis aporta. El resto del documento está organizado como sigue:

En el capítulo 2 se presenta el estado del arte referente al panorama actual de investigación de los diferentes métodos, técnicas y herramientas en el reconocimiento de personas mediante el iris y huella dactilar.

En el capítulo 3 se presentan aquellas herramientas matemáticas y conceptuales necesarias para el desarrollo de este trabajo.

En el capítulo 4 se desarrolla la parte principal de esta tesis, dado que aquí se presentan los métodos de procesamiento digital de imágenes y los algoritmos de clasificación utilizados en el sistema multibiométrico. Asimismo, se describe la arquitectura, diseño y flujo de trabajo de la aplicación web y las aplicaciones móviles.

En el capítulo 5 se realiza un análisis sobre los resultados experimentales obtenidos con el proceso de reconocimiento de personas propuesto sobre el sistema multibiométrico.

Finalmente, en el capítulo 6 se exponen las conclusiones, aportaciones y trabajos a futuro.

## Capítulo 2

### 2. Estado del Arte

En la actualidad el reconocimiento de personas mediante el uso de sistemas biométricos se ha utilizado cada vez más, para las tareas de verificación e identificación de personas. Más recientemente se han ido desarrollando nuevos sistemas que permitan obtener mejores resultados denominados sistemas multibiométricos, los cuales hacen uso de más de un biométrico para las tareas antes mencionadas.

En la verificación se pretende confirmar que un usuario es quien dice ser, proporcionando una muestra biométrica de algún tipo; por ejemplo, una organización que solicita una muestra de una huella dactilar que es capturada con algún dispositivo, la cual es utilizada para determinar si se da acceso a un lugar, donde dicha muestra proporcionada es comparada con otra huella biométrica del mismo tipo previamente registrada y que es perteneciente a la persona que tiene acceso a dicho lugar, siendo el resultado de la comparación el valor que determinara si el usuario puede acceder o no.

El proceso de identificación conlleva dificultades adicionales, debido a que se pretende identificar a la persona a quien pertenece una muestra o huella biométrica, sin conocer a esa persona. Para lograr esto, normalmente la huella biométrica que se quiere identificar debe ser comparada con un conjunto grande de huellas biométricas del mismo tipo que las que se quiere identificar, las cuales pertenecen a personas de quienes se posee cierta información. En el caso de la identificación, no hay garantía de obtener un resultado concluyente debido a que puede ser que no se tenga información de la muestra que se desea identificar.

A continuación, se presentan dos de los biométricos más relevantes del estado del arte, los cuales son comúnmente utilizados en el reconocimiento de personas, tanto para la identificación como la verificación. Posteriormente, se presentarán los avances sobre los sistemas multibiométricos y se analizarán las razones por las que están siendo utilizados cada vez más.

#### 2.1 Huella dactilar

La huella dactilar es uno de los primeros biométricos utilizados para la implementación de sistemas de reconocimiento de personas, y en la actualidad uno de los más utilizados por su alta eficacia al momento de identificar o verificar personas [5].

El uso de huellas dactilares posee muchas ventajas sobre otros biométricos, por lo que se ha convertido en un campo de investigación bastante maduro. Una de las principales ventajas es la singularidad que poseen las huellas dactilares, debido a que la probabilidad de que dos huellas dactilares coincidan es vagamente pequeña: algunos autores han encontrado que la probabilidad de que dos huellas dactilares sean iguales es de 1 en  $1.9 \times 10^{15}$  [6, 7]. Estas características hacen que el uso de huellas dactilares sea extremadamente efectivo en áreas donde el alto grado de seguridad es un problema.

Una huella dactilar se compone de muchas crestas, valles y minucias, que son algunos puntos anormales en las crestas; estas minucias permiten establecer distinciones entre dos huellas diferentes [8, 9]. En la Figura 2 se muestran los tipos de minucias más comunes que se pueden encontrar en una imagen de huella dactilar.





Tipo de Minucia	Cresta Corta	Cresta de Terminación	Bifurcación	Espuela
Ejemplo				

Figura 2 Tipos de Minucia (Cresta corta, cresta de terminación, bifurcación y espuela).

En el trabajo de *Anil et al.* [10], se presenta el desarrollo de un sistema que permite la verificación de huellas dactilares en línea, el cual opera por medio de dos etapas: extracción de minucias y emparejamiento de minucias. Ratha desarrolló una versión mejorada de este algoritmo [11], el cual es mucho más rápido y fiable. Además de mejorar la extracción de minucias, para la etapa del emparejamiento de minucias se desarrolló un algoritmo denominado coincidencia elástica basado en la alineación. Este algoritmo se dice es capaz de encontrar las correspondencias entre las minucias en la imagen de entrada al sistema y la plantilla almacenada, sin recurrir a una búsqueda exhaustiva y que además tiene la capacidad de adaptarse para compensar las deformaciones no lineales y transformaciones inexactas que haya entre las huellas dactilares. El sistema se probó con dos conjuntos de huellas dactilares capturadas con escáneres propios. La conclusión a la que se llegó, es que la exactitud de la verificación es aceptable, tomando en promedio ocho segundos para la verificación, lo cual cumple con los requisitos de tiempo de respuesta aceptables para la verificación en línea.

Para que un sistema de identificación y verificación de huellas usadas en el reconocimiento de personas sea robusto con respecto a las imágenes de huellas dactilares de entrada, es esencial incorporar al menos un algoritmo que permita mejorar la calidad de las huellas dactilares presentes en las imágenes, logrando con ello mejores resultados en sistemas de identificación y verificación. En el trabajo desarrollado por *Hong et al.* [5] se presenta el desarrollo de un algoritmo rápido de mejora de la calidad en imágenes de huellas dactilares, cuyo funcionamiento permite de forma adaptativa mejorar la claridad de las estructuras de crestas y valles presentes en las imágenes de entrada, basándose para ello en la estimación local de la orientación y frecuencia de las crestas. El rendimiento del algoritmo de la mejora de la imagen se midió utilizando el índice de bondad de las minucias extraídas y la precisión de un sistema desarrollado para la verificación de huellas en línea; y los resultados mostraron que, al incorporar el algoritmo de mejora, el índice de bondad y la exactitud de la verificación se incrementó desde un 1% hasta un 15%.

En el trabajo de *Tan et al.* [12], se propone un enfoque de reconocimiento de huellas digitales basado en la aplicación de algoritmos genéticos, para tratar de encontrar la transformación óptima entre dos huellas diferentes. Este algoritmo ataca problemas comunes como la traslación y rotación de minucias. Con la finalidad de atacar el problema cuando se tienen imágenes de baja calidad afectando la detección correcta de minucias de manera significativa, se diseñó una función *fitness* basada en las propiedades locales de cada triplete de minucias. El resultado demostró que el enfoque propuesto puede lograr un buen rendimiento incluso cuando una gran parte de huellas dactilares en la base de datos sea de mala calidad, pero también mostrando ser mejor que el enfoque normalmente usado, basándose en la estimación del error cuadrático medio.

La mayoría de las técnicas de reconocimiento de huellas dactilares se basan en la coincidencia de minucias ya que han sido bien estudiadas. Sin embargo, estas tecnologías aún sufren de problemas relacionados con la manipulación de imágenes digitales de huella dactilar con mala calidad. Uno de estos problemas es la distorsión, dado que puede cambiar tanto la posición geométrica como la

orientación, provocando dificultades para crear un emparejamiento entre múltiples imágenes de huella dactilar pertenecientes al mismo dedo. En el trabajo de *Kaur et al.* [13] se combinan varios métodos para la construcción de un sistema que permita extraer minucias y compararlas. Un método utilizado es la Transformada Discreta de Fourier (DFT) para mejorar la calidad de las imágenes. También se incluye la segmentación mediante operaciones morfológicas, adelgazamiento y algoritmos que permiten la eliminación de minucias falsas; en este sentido, [13] incluye un método que adelgaza las crestas de las huellas dactilares con un pixel de ancho, con lo cual la mayoría de las bifurcaciones son detectadas, con la posterior eliminación de las minucias falsas.

En el trabajo de *Chatterjee et al.* [9] se propone un método para la identificación de huellas dactilares por medio de la extracción de características usando el entrenamiento de redes neuronales mediante el algoritmo backpropagation. La imagen es binarizada y posteriormente adelgazada, para luego estimar la orientación local de las crestas y localizar la región de interés. Las minucias consideradas para la extracción fueron: crestas cortas, crestas de terminación, bifurcaciones y espurias. Los valores obtenidos de las minucias obtenidas sirven como datos de entrada para la red neuronal propuesta para su entrenamiento y posteriormente ser utilizada para el reconocimiento de huellas dactilares. Los resultados experimentales demuestran que el porcentaje de muestras reconocidas con este método es del 95% en comparación a un sistema original utilizando redes neuronales cuyo rendimiento fue del 92.5%.

Como se ha visto anteriormente, lo normal para la identificación o verificación de huellas dactilares, es el uso de minucias extraídas por diversos algoritmos existentes; sin embargo, cuando lo que se requiere es que la etapa del emparejamiento sea rápida, esto no es suficiente. *Chatterjee et al.* [14] solventan este problema mediante la localización del centro de la huella dactilar denominado *Core Point*, y con ello presentan una mejora en los tiempos de respuesta y una reducción de la tasa de falsos negativos [15].

## 2.2 Iris

Otro de los primeros biométricos utilizados en la implementación de sistemas de reconocimiento de personas es el iris, siendo su uso muy común en el reconocimiento de personas, ya sea para la identificación o verificación. La siguiente descripción está basada fuertemente en [16-25].

El iris es un anillo de tejido de color ubicado alrededor de la pupila, lugar donde entra la luz hacia el interior del ojo. Normalmente, la pupila tiene una apariencia más oscura que el iris y esta característica es útil para ubicar el iris. Una cubierta transparente llamada córnea cubre la pupila y el iris. La córnea genera reflejos de luz, esto dificulta la adquisición de imágenes del iris, pero también puede ser de ayuda para localizarlo. La finalidad del iris es controlar la cantidad de luz que entra dentro del ojo en el cual hay dos músculos que sirven para dilatarlo o contraerlo, lo cual genera una deformación que también es un problema para la extracción de características invariantes a agentes externos.

La esclerótica es una región blanca con vasos sanguíneos que rodea al iris. La frontera interior del iris (entre el iris y la pupila) se denomina frontera pupilar. Y la frontera exterior del iris (entre el iris y la esclerótica) se denomina frontera límbica. Uno de los pasos iniciales para la comparación de iris es la localización de estas dos fronteras. La Figura 3 muestra la imagen de un ojo y algunas de sus características.

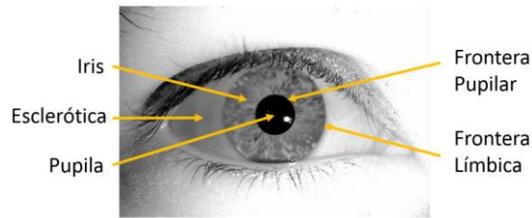


Figura 3 Imagen del ojo que especifica algunas características útiles en el reconocimiento de iris.

La superficie del iris contiene un patrón muy rico de surcos, contornos y pecas de pigmento. Este patrón es el que sirve para generar una identificación que tendría que ser única e invariable para cada persona. Se cree que este patrón se determina de forma aleatoria durante el desarrollo fetal del ojo. Se considera que es diferente para todas las personas y también entre el ojo izquierdo y el derecho de la misma persona. El color del iris varía según la cantidad de pigmento que tenga. Normalmente, aumenta durante la infancia. No obstante, durante una gran parte del período de vida de las personas, la apariencia del iris es relativamente estable. En la Figura 4 se muestran algunos de los procesos a realizar en el reconocimiento de iris.

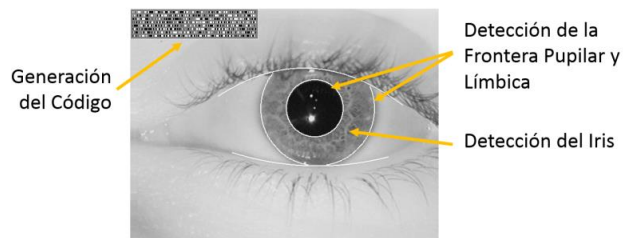


Figura 4 Imagen del ojo que muestra algunos procesos en el reconocimiento de iris.

Uno de los trabajos más importantes del reconocimiento del iris es la patente de Daugman, presentada en 1994 [16]. En esta patente, se describe un sistema de reconocimiento de iris con bastante detalle y el cual ha sido mejorado constantemente por Daugman [17, 18, 19].

El sistema que describe ha inspirado todos los sistemas que se han ido desarrollando posteriormente puesto que ha marcado un modelo estándar. Debido a que esta patente se mantuvo durante un tiempo por la misma empresa (IrisCodes®), se puede decir que la mayoría de los sistemas actuales se basan en la tecnología de Daugman [20].

La alta aleatoriedad de los patrones de iris lo hace uno de los biométricos con rasgos más fiables. Sin embargo, la compleja estructura de la imagen del iris y las diversas variaciones intra-clase provocan cierta dificultad para representar el iris. Aunque hay varios métodos de reconocimiento de iris propuestos, se han encontrado ciertos algoritmos de reconocimiento de iris que usan técnicas multiescala, los cuales proveen una adecuada representación del iris para el reconocimiento del iris. En el trabajo de *Nabti et al.* [21] se presenta un enfoque de detección de bordes multiescala empleando una etapa de pre-procesamiento para localizar el iris seguido de una técnica de extracción la cual se basa en una combinación de algunas de las técnicas más utilizadas para la extracción de características multiclase. Esta combinación usa los filtros espaciales de Gabor y componentes de wavelet máxima. Adicionalmente, un esquema de emparejamiento basado en la operación OR

exclusiva para computar los bits de similitud es propuesto y donde el resultado de la experimentación fue llevado a cabo utilizando una base de datos de CASIA. Los resultados experimentales muestran que el sistema propuesto da un rendimiento atractivo y puede ser utilizado para la identificación personal de una manera eficiente y efectiva en comparación con los mejores algoritmos de reconocimiento de iris encontrados en la literatura.

Los biométricos que no requieren del contacto como la cara y el iris poseen beneficios adicionales sobre los biométricos basados en el contacto como la huella dactilar o la geometría de la mano. Sin embargo, tres importantes desafíos deben ser abordados en un sistema de autenticación basado en biométricos que no requieren contacto. En el trabajo de *Pillai et al.* [22] se propone un framework unificado basado en proyecciones aleatorias y representaciones escasas, que permite abordar simultáneamente las 3 cuestiones antes mencionadas en relación al biométrico del iris. La medida de calidad propuesta puede controlar los errores de segmentación y una amplia variedad de artefactos encontrados durante la adquisición del iris. Se demostró que los enfoques propuestos pueden ser fácilmente extendidos para manejar variaciones de alineación y el reconocimiento de iris en videos, dando como resultado un sistema robusto y preciso. El enfoque propuesto incluye mejoras para la privacidad y seguridad proporcionando maneras de crear plantillas de iris cancelables. Los resultados muestran beneficios significantes utilizando el enfoque propuesto en bases de datos públicas.

Las comunicaciones inalámbricas han madurado de una curiosidad a una herramienta de negocio seria. Los asistentes digitales personales han reemplazado casi por completo agendas y cuadernos. Pero la seguridad es aún carente o débil, siendo la automatización deseada poco fiable para aplicaciones críticas. Añadiendo una fuerte seguridad y autenticación, estas herramientas facilitan métodos fiables para el correo electrónico, transacciones financieras, datos médicos, e incluso recetas. En el trabajo de *Kang et al.* [23] se considera el limitado poder de cómputo de dispositivos móviles y portátiles, con un simple pero eficiente método de pre-procesamiento introducido para la localización del iris en imágenes digitales. Una base de datos con dichas consideraciones se utilizó para las pruebas. El método de pre-procesamiento de iris propuesto se implementó mediante los siguientes pasos: a) Segmentación automática de la región de la pupila, b) Detección y extracción de datos, c) Detección de párpados y emparejamiento de características. Los resultados experimentales mostraron que el método de pre-procesamiento funciona bien y es estable utilizando diferentes bases de datos.

En el trabajo de *Singh et al.* [24] se describen varias técnicas desarrolladas para crear un sistema de reconocimiento de iris y un análisis de los resultados obtenido. Se utilizó un mecanismo con la fusión de un sistema de detección de bordes mediante Canny y la transformada de Hough, para detectar los límites del iris en una imagen digital. Posteriormente fue aplicado el wavelet de Haar con el fin de extraer patrones determinísticos a partir del iris de una persona en forma de un vector de características. Mediante la comparación de vectores cuantificados usando el operador de la distancia de Hamming, se determinó finalmente si dos iris eran similares, concluyendo que el sistema creado era eficaz.

En el reconocimiento de iris, siendo la etapa de identificación que en general y que normalmente se efectúa mediante comparativas elemento a elemento con el uso de algún método de distancia conocido como matching; lo cual resulta en un alto costo computacional. En el trabajo de *Santiago*

*et al.* [25] se propone un algoritmo de reconocimiento automático basado en memorias asociativas conocido como clasificador asociativo de patrones multiclase, para realizar dicha etapa de reconocimiento de iris. Además, se aplica un algoritmo micro-evolutivo para encontrar patrones idóneos de los descriptores que en este caso son los filtros de Gabor, y con ello obtener los vectores característicos adecuados para el clasificador para que éste pudiera recuperar adecuadamente el conjunto fundamental en la base de datos utilizada.

### **2.3 Sistemas multibiométricos**

Si bien el uso de sistemas que hacen uso de un solo biométrico presentan resultados buenos en las tareas de autenticación y verificación de personas, el rendimiento puede ser mejorado mediante la combinación de dos o más biométricos. Otro factor muy importante y que hay que tomar en consideración es que la mayoría de los sistemas que se basan en un solo biométrico y cuyo objetivo es obtener una mayor seguridad y privacidad, en la actualidad son seriamente amenazados por ataques de tipo spoof, que en términos generales consiste en suplantar la huella biométrica que se le presenta a un sistema biométrico con una huella generada artificialmente mediante algún método

La tecnología biométrica se utiliza tanto para identificar o verificar la identidad de individuos con eficacia en comparación con los métodos tradicionales existentes. En el trabajo de *Ravi et al.* [75] se propone un sistema basado en el uso de dos biométricos (Iris y Huella Dactilar), donde las plantillas, tanto del iris como de la huella dactilar, son generadas por separado con diferentes algoritmos de preprocesamiento aplicados sobre imágenes correspondientes a los biométricos. Una vez que se obtienen las características del iris y la huella dactilar por separado, éstas son fusionadas para obtener un solo conjunto final de características único. Para la tarea del reconocimiento, el vector de características que se desea identificar o verificar es comparado con otros previamente obtenidos mediante la distancia euclidiana. Los resultados obtenidos presentan mejoras en comparación a los algoritmos utilizados de forma separada para cada biométrico. Los resultados fueron medidos con las siguientes métricas: Total Success Rate (TSR), False Rejection Rate (FRR) y False Acceptance Rate (FAR).

Los sistemas multibiométricos hacen uso de más de un solo biométrico que pueden ser: la cara, el iris, la retina o la huella dactilar. Estos sistemas ayudan a resolver algunas de las limitaciones de los sistemas biométricos individuales, tales como: presencia de datos ruidosos, la no universalidad, ataques fraudulentos y la tasa de error inaceptable; además, se mejora la precisión del reconocimiento en comparación con los sistemas que hacen uso de un solo biométrico. En el trabajo de *Mohamad et al.* [76] se presenta el desarrollo de un sistema multibiométrico con el uso del iris y la huella dactilar mostrando buenos resultados en cuanto a la precisión. Para la obtención de los resultados se utilizó un nivel de decisión, lógica difusa y la ponderación de los resultados obtenidos de manera individual para fusionar los resultados de ambos biométricos.

### **2.4 Máquinas Asociativas Alfa-Beta SVM**

Las Máquinas Asociativas Alfa-Beta SVM (*Alpha-Beta SVM*) toman como punto de partida a las memorias asociativas Alfa-Beta [77], específicamente los operadores alfa y beta y sus propiedades; y del modelo SVM (Support Vector Machines) se toma el concepto de vector de soporte, aunque el concepto se maneja de forma distinta a como lo manejan las SVM, mediante el uso de los operadores alfa y beta [77]. Si bien las Máquinas Asociativas Alfa-Beta SVM no han tenido mucha presencia en el desarrollo de sistemas biométricos, han mostrado tener un buen rendimiento ante



ciertos problemas correspondientes a otras áreas de investigación [55, 56, 57, 58]; lo cual da pie a hipotetizar que es posible que puedan funcionar en el reconocimiento del iris y la huella dactilar.

## 2.5 Clasificador Gamma

Al igual que las Máquinas Asociativas Alfa-Beta SVM, el clasificador Gamma toma como punto de partida a las Memorias Asociativas Alfa-Beta [77], específicamente los operadores alfa y beta y sus propiedades, a fin de proponer el operador gamma de similitud, así como del operador  $\mathcal{U}_\beta$ ; además, el clasificador Gamma usa el concepto de módulo y el código Johnson-Möbius, que como se mencionó anteriormente permite solucionar parcialmente problemas derivados de alteraciones aditivas, sustractivas o una mezcla de ambas. De manera similar a lo que ocurre con las Máquinas Asociativas Alfa-Beta SVM, el clasificador Gamma no ha tenido gran presencia en la implementación de sistemas biométricos; sin embargo, en diferentes publicaciones se ha verificado que exhibe un buen rendimiento en otras áreas [59, 60, 61, 62, 63, 64, 65, 66], por lo que se espera que en el reconocimiento de personas mediante la huella dactilar pueda ser de utilidad.

## 2.6 Aplicaciones Web y Móviles

Si bien las primeras aplicaciones web eran simples páginas que servían contenido estático, y el usuario no podía interactuar directamente con ellas. Con el tiempo fue surgiendo la necesidad de tener una mayor interacción con los sistemas desarrollados, sin embargo, la naturaleza misma del protocolo HTTP es unidireccional, lo cual implica que sólo se permite el envío de paquetes hacia el cliente, lo cual dio paso a la creación de protocolos alternativos para distintos tipos de problemas que se presentaron para agregar mayor funcionalidad [78, 79].

En la actualidad, algunos de los protocolos existentes son: FTP (File Transfer Protocol) para la transferencia de archivos, SMTP (Simple Mail Transfer Protocol), para transferencia simple de correo, POP3 (Post Office Protocol) para el correo electrónico, Telnet para acceder a equipos remotos, entre otros. Varios de estos protocolos tuvieron su auge a principios del siglo XXI, cuando la WWW fue apoyada por un conjunto de tecnologías creadas a lo que se le denominó como AJAX (Asynchronous Javascript And XML) que permite simular una comunicación bidireccional entre el cliente y el servidor.

Durante la última década, el término Web 2.0 [80, 81] se hizo popular y el cual es utilizado para referirse a aquellos sitios de Internet que generan su contenido mediante la participación de los usuarios. Este concepto se popularizó gracias al auge que se tuvo con la creación de blogs, redes sociales, plataformas colaborativas y similares. Algunas tecnologías utilizadas en la Web 2.0 son: AJAX, RSS, JSON, XML, Javascript, entre otras más.

Recientemente, con la versión 5 del lenguaje HTML (HTML5) que se ha ido popularizando en Internet [82, 83, 84], ha cambiado el funcionamiento de algunas aplicaciones que existen actualmente, aunque no muy apreciable por los usuarios. HTML5 representa una mejora en los navegadores de Internet que lo soportan, así como en los servidores HTTP que permiten actuar conjuntamente para sincronizar contenido entre el cliente y el servidor para acceder a archivos vía remota y funcionar de manera online.

## Capítulo 3

### 3. Materiales y Métodos

En el presente capítulo se describen los materiales y métodos necesarios para el desarrollo del sistema multibiométrico que se presentará en el capítulo 4. En la primera sección se describe de manera general lo que es el procesamiento digital de imágenes. En la segunda sección se explica la morfología matemática y las operaciones morfológicas más utilizadas. Posteriormente, en las siguientes dos secciones se explicarán los métodos de procesamiento digital de imágenes utilizados en las distintas etapas del reconocimiento de huellas dactilares y de iris, respectivamente. Después, se describirán los Clasificadores de Patrones de manera general, para posteriormente explicar el funcionamiento de las Máquinas Asociativas Alfa-Beta SVM y el Clasificador Gamma de Patrones utilizados en el algoritmo propuesto para la identificación y verificación de la identidad de personas. Finalmente, en las secciones posteriores se describirán las tecnologías que se utilizarán en el sistema web y en los dispositivos móviles.

#### 3.1 Procesamiento Digital de Imágenes

El contenido de esta sección se basa fuertemente en [85]. El procesamiento digital de imágenes es el conjunto de métodos que se aplican a las imágenes digitales, con el objetivo de mejorar su calidad o hacer más evidentes detalles para hacer más fácil la búsqueda de información.

Una imagen puede ser definida como una función bidimensional  $f(x, y)$ , donde  $x$  y  $y$  son coordenadas espaciales (plano), y la amplitud de  $f$  en algún par de coordenadas  $(x, y)$  es llamado la intensidad o nivel de gris de la imagen en ese punto. Cuando  $x$ ,  $y$  y la intensidad de los valores de  $f$  son todos finitos, cantidades discretas, decimos entonces que la imagen es una imagen digital.

El campo del procesamiento de una imagen digital se refiere al procesamiento digital de imágenes por medio de una computadora digital. Notar que la imagen digital está compuesta de un número finito de elementos, cada uno de los cuales tiene una ubicación y valor en particular. Estos elementos son llamados elementos de la imagen, pels o píxeles. Pixel es el término comúnmente más utilizado para denotar los elementos de una imagen digital.

Para manipular una imagen digital representada por una función bidimensional  $f(x, y)$ , se considera que la variable  $x$  representa a las filas en la imagen digital y  $y$  a las columnas, y en lugar de manipular la imagen como una función  $f$ , se utiliza un arreglo bidimensional  $I$  de tamaño  $M \times N$ , donde  $M$  representará la cantidad de filas y  $N$  el tamaño de columnas totales de la imagen digital. Por notación y claridad se tiene que  $x = 0, 1, 2, \dots, M - 1$  y  $y = 0, 1, 2, \dots, N - 1$ . Siguiendo con la notación, se tiene que el origen de la imagen digital es en la posición  $(0, 0)$ , y la siguiente posición conservando la misma fila será  $(0, 1)$ . En general el valor en la posición  $(x, y)$  de la imagen digital está denotado por  $I(x, y)$ , siendo  $x$  y  $y$  valores enteros.

#### 3.2 Morfología Matemática

De acuerdo con [72], la morfología matemática es una herramienta muy poderosa para realizar distintos procesos en el procesamiento digital de imágenes, entre las que destacan la segmentación de formas, la determinación de bordes y la generación de esqueletos.

La morfología matemática hace uso de dos operaciones principales que son: la dilatación y erosión, las cuales son la base de todas las operaciones morfológicas. En morfología binaria, una imagen  $A$  es vista como un subconjunto de un espacio euclidiano para alguna dimensión  $d$ :

$$A \subseteq Z^d$$

Un elemento de estructura (SE – Structure Element) es una imagen que permite investigar la morfología de otra imagen.

Algunas de las operaciones básicas son las siguientes:

### Erosión

La erosión es una operación morfológica que permite reducir el tamaño de los objetos oscuros sobre fondo claro en una imagen, quitando las partes que no caben en el elemento de estructura proporcionado.

Sea  $X$  un espacio euclidiano en los enteros que representa una imagen y  $A$  una imagen binaria que se encuentra en  $X$ , la erosión de  $A$  con un elemento de estructura  $B$  es denotado por  $A \ominus B$ , conocida como la resta de Minkowski de  $A$  y  $B$ , es decir:

$$A \ominus B = \{x \in X \mid x + b \in A, \forall b \in B\}$$

La erosión también puede ser definida como una inserción de translaciones del elemento de estructura mediante la fórmula:

$$A \ominus B = \bigcap_{b \in B} (A)_{-b}$$

### Dilatacion

La dilatación es la operación morfológica dual de la erosión, que permite aumentar el tamaño de los objetos oscuros sobre fondo claro, dentro de una imagen, agregando las partes del elemento de estructura proporcionado que no estén incluidas en el objeto.

Sea  $X$  un espacio euclidiano en los enteros que representa una imagen y  $A$  una imagen binaria que se encuentra en  $X$ , la dilatación de  $A$  con un elemento de estructura  $B$  es denotado por  $A \oplus B$ , conocida como la suma de Minkowski de  $A$  y  $B$ , es decir:

$$A \oplus B = \{x = a + b \in X \mid a \in A \wedge b \in B\}$$

La dilatación también puede ser definida como una unión de translaciones del elemento de estructura mediante la fórmula:

$$A \oplus B = \bigcup_{b \in B} (A)_b$$

### Apertura y Cerradura

La apertura es una operación morfológica que permite resaltar las partes claras de una imagen, reduciendo las partes oscuras en la misma. La idea principal de la apertura es realizar una dilatación a una imagen previamente erosionada con el mismo elemento de estructura para recuperar tanto como sea posible la imagen original.

La apertura de un conjunto  $A$  por un elemento de estructura  $B$  se denota por  $A \circ B$  y está definida como la erosión de  $A$  con  $B$ , seguida de una dilatación con el mismo elemento de estructura, definido como:

$$A \circ B = (A \ominus B) \oplus B$$

Algunas de las propiedades que cumple la apertura son:

- La idempotencia, que significa que las aperturas adicionales con el mismo elemento de estructura no afectan el resultado.
- La apertura dará como resultado siempre una imagen más pequeña que la original.

Los efectos de la apertura en una imagen dada se pueden resumir en cuatro aspectos:

- Elimina islas de tamaño menor al elemento de estructura.
- Elimina picos o cabos más delgados que el elemento de estructura.
- Rompe istmos cuya anchura sea menor al diámetro del elemento de estructura.
- Alisa el contorno convexo de la imagen.

De manera similar la cerradura de un conjunto  $A$  por un elemento de estructura  $B$  se denota por  $A \cdot B$  y está definida como la dilatación de  $A$  con  $B$ , seguida de una erosión con el mismo elemento de estructura, definido como:

$$A \cdot B = (A \oplus B) \ominus B$$

La cerradura produce efectos diferentes a los que produce la apertura, estos efectos son:

- Rellena lagos o huecos de tamaño menor al elemento de estructura.
- Rellena rajaduras o golfos más delgados que el elemento de estructura.
- Funde estrechos cuya anchura sea menor al diámetro del elemento de estructura.
- Alisa el contorno cóncavo de la imagen, rellenando rompimientos.

Dado que las operaciones morfológicas son relativamente fáciles de computar, la morfología matemática es usada como herramienta útil en el procesamiento en imágenes [67, 68, 69, 70, 71].

### **3.3 Procesamiento Digital de imágenes de Huella Dactilar**

En esta sección se presentan algunos de los métodos de procesamiento digital de imágenes utilizados en el reconocimiento de huella dactilar: la mejora de la calidad de las imágenes, extracción de características y generación de los patrones.

### 3.3.1 Mejora de la calidad

El rendimiento de los algoritmos que sirven para la extracción de características y otras técnicas usadas en el reconocimiento de huellas dactilares dependen en gran medida de la calidad de la imagen. En una imagen de huella ideal, crestas y valles se alternan y fluyen en una dirección localmente constante. El objetivo de esta etapa es mejorar la claridad de las estructuras de las crestas en las regiones recuperables y marcar las regiones irrecuperables para su posterior procesamiento [85]. Para la mejora de las imágenes se utilizará un filtro en el dominio de la frecuencia por medio de la Transformada de Fourier [75, 76] y una técnica de ecualización del histograma [75, 76, 86].

Antes de proceder con los métodos de mejora de la calidad de las imágenes de huella dactilar es necesario explicar previamente lo que es la convolución.

#### Convolución

La convolución es una operación denominada  $c$ , donde para cada pixel  $(x, y)$  de una imagen digital  $I$ , se obtiene un nuevo valor, utilizando una ventana deslizante  $W$  (normalmente de  $3 \times 3$ ); lo anterior se expresa con la siguiente fórmula utilizando una ventana deslizante de convolución  $W$  de tamaño  $3 \times 3$ :

$$c(x, y, W) = W0 * I[x - 1, y + 1] + W1 * I[x, y + 1] + W2 * I[x + 1, y + 1] + W3 * I[x - 1, y] + W4 * I[x, y] + W5 * I[x + 1, y] + W6 * I[x - 1, y - 1] + W7 * I[x, y - 1] + W8 * I[x + 1, y - 1]$$

Es importante mencionar que el resultado de la convolución es una imagen  $C$  que no necesariamente tendrá valores en el rango  $[0 - 255]$ . Por lo que es necesario en ocasiones normalizar dicha imagen.

Para la normalización es utilizado un proceso simple y explicado mediante los siguientes pasos:

- 1) Se localiza el valor  $m$  más pequeño dentro de la imagen  $C$ .
- 2) Se suma el valor absoluto de  $m$  a todos los valores de la imagen  $C$ , lo cual hará que la imagen solo tenga valores no negativos.
- 3) Posteriormente se localiza el valor  $n$  más grande en la imagen obtenida en el paso anterior.
- 4) Para cada pixel de la imagen se aplica la siguiente fórmula.

$$i(x, y) = \frac{c(x, y) * n}{255}$$

- 5) El resultado del paso anterior será una nueva imagen  $I$  normalizada cuyos valores se encuentran en el rango  $[0 - 255]$ .

#### Ecualización del histograma

Este método usualmente incrementa el contraste global de muchas imágenes, permitiendo que áreas de bajo contraste local ganen un mayor contraste sin afectar el contraste global. La ecualización del

histograma logra esto mediante la distribución uniforme de la manera más efectiva posible de los valores de intensidad más frecuentes en la imagen [86].

El histograma de una imagen digital con niveles de intensidad en el rango  $[0, L - 1]$  es una función discreta  $h(r_k) = n_k$ , donde  $r_k$  es el  $k$ -ésimo valor de intensidad y  $n_k$  es el número de píxeles en la imagen con el valor de intensidad  $r_k$ . Es una práctica común el normalizar un histograma dividiendo cada uno de sus componentes por el número total de píxeles en la imagen, denotado por el producto  $MN$ , donde  $M$  son las filas y  $N$  las columnas de la imagen. La normalización del histograma está dada por  $p(r_k) = n_k/MN$ , para  $k = 1, 2, \dots, L - 1$ . En pocas palabras,  $p(r_k)$  es un estimado de la probabilidad de ocurrencia de la intensidad del nivel  $r_k$  en una imagen. La suma de todas las componentes de un histograma normalizado es igual a 1 [85].

Consideremos que la variable  $r$  denota los valores continuos de intensidad de una imagen que será procesada. Asumimos que  $r$  está en el rango  $[0, L - 1]$ , con  $r = 0$  representando el negro y  $r = L - 1$  representando el blanco. Para  $r$  que satisface estas condiciones, centramos la atención en las transformaciones (mapeo de intensidades) de la forma:

$$s = T(r), 0 \leq r \leq L - 1$$

Esto produce una salida de nivel de intensidad  $s$  para cada pixel en la imagen de entrada con intensidad  $r$ . Asumimos que:

- 1)  $T(r)$  es una función monótona creciente en el intervalo  $0 \leq r \leq L - 1$ ; y
- 2)  $0 \leq T(r) \leq L - 1$ , para  $0 \leq r \leq L - 1$ .
- 3) El requerimiento en la condición (1) en el que  $T(r)$  debe ser monótonamente creciente garantiza que la salida de intensidad de los valores nunca será menor con respecto a los valores de entrada. La condición (2) garantiza que el mapeo de  $s$  con respecto a  $r$  será uno a uno, para prevenir ambigüedades.

### **Transformada Discreta de Fourier**

El rendimiento en algoritmos de extracción de minucias y otras técnicas de reconocimiento de huella dactilar depende fuertemente de la calidad de las imágenes de entrada para poder distinguir la alternancia entre crestas y valles presentes en las imágenes de huella dactilar y poder determinar de manera correcta el flujo y la dirección local de las mismas [86].

La Transformada Discreta de Fourier (TDF) es una importante herramienta en el procesamiento digital de imágenes, la cual se utiliza para convertir una imagen de su dominio espacial al dominio de la frecuencia y viceversa. Sin embargo, computacionalmente es muy costosa su implementación, por lo que en su lugar es utilizado el algoritmo de la Transformada Rápida de Fourier (TRF). Una TRF calcula rápidamente un gran número de transformaciones siendo a menudo muy utilizada para muchas aplicaciones de ingeniería, ciencia y matemáticas [13, 76].

Si bien la TRF se puede aplicar en toda la imagen, es posible dividir la imagen en bloques (no obligatorio). Para aplicar la TRF en imágenes de huella dactilar, en este trabajo de tesis se propone

utilizar la imagen  $I$  de tamaño  $M \times N$  dividida en bloques pequeños (32 x 32 pixeles, con los que se obtuvo mejor resultado) que serán procesados por la TRF, de acuerdo con la siguiente ecuación:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(\frac{ux}{M} + \frac{vy}{N})}, \text{ para } u = 0, 1, 2, \dots, 31 \text{ y } v = 0, 1, 2, \dots, 31.$$

Para mejorar un bloque específico en el dominio de la frecuencia, se propone multiplicar el resultado de la TRF del bloque por su magnitud un número de veces  $k$ . Lo anterior definido como:

$$|F(u, v)|^k$$

Obteniendo así un bloque mejorado acorde con la siguiente ecuación:

$$g(x, y) = F^{-1}\{F(u, v) * |F(u, v)|^k\}$$

Donde  $F^{-1}(F(u, v))$  es la Transformada Inversa de Fourier, dada por la siguiente ecuación:

$$f(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u, v) e^{j2\pi(\frac{ux}{M} + \frac{vy}{N})}, \text{ para } x = 0, 1, 2, \dots, 31 \text{ y } y = 0, 1, 2, \dots, 31.$$

El valor de  $k$  utilizado en la magnitud de la TRF, es una constante que se determina experimentalmente, donde entre mayor sea el número del valor de  $k$ , la apariencia de las crestas mejorará; sin embargo, no es muy recomendable que sea tan alto ya que las crestas pueden llegar a unirse y dar la apariencia de que son la misma cresta en lugar de varias.

### 3.3.2 Extracción de características en Imágenes de Huellas Dactilares

Para la extracción de características en imágenes de huellas dactilares, se hace uso de técnicas de procesamiento digital de imágenes, que permiten agilizar el proceso de la extracción de las características, las cuales son: la segmentación que permite detectar en la imagen la zona donde hay huella dactilar, la binarización que analizará sólo la zona donde haya huella dactilar en la imagen teniendo como resultado una imagen donde las crestas serán de color negro y el fondo y valles de color blanco. Posteriormente se aplicará un algoritmo que permite adelgazar la imagen para que las crestas tengan sólo un pixel de ancho y finalmente el algoritmo que permite la extracción de las minucias a partir de la imagen adelgazada.

La binarización es un proceso que convierte una imagen en escala de grises en una imagen binaria, donde el valor en cada pixel de la imagen sólo puede tener dos valores: 0 o 255. La mayoría de los algoritmos de extracción de minucias operan sobre imágenes binarias donde hay únicamente dos niveles de interés: pixeles negros que representan las crestas, y los pixeles blancos que representan los valles. Esto mejora el contraste entre las crestas y los valles en las imágenes de huella dactilar, y consecuentemente facilita la extracción de minucias.

#### Método de Sobel

Este método permite calcular el gradiente  $\Delta(x, y)$  en cada pixel  $(x, y)$  de la imagen de entrada. El gradiente se define como un vector bidimensional tal que:

$$\Delta(x, y) = [\Delta_x(x, y), \Delta_y(x, y)]$$

Donde los componentes  $\Delta_x$  y  $\Delta_y$  corresponden a las derivadas horizontal y vertical respectivamente. A partir del gradiente se puede determinar la magnitud, la dirección y el sentido donde hay un mayor cambio posible en las tonalidades de oscuro a claro en la imagen.

Para calcular las componentes  $\Delta_x$  y  $\Delta_y$  se hace uso de la convolución de la imagen de entrada  $I$  con una ventana  $W$  definidas para la obtención de  $\Delta_x$  y  $\Delta_y$ . La Figura 5 muestra las ventanas utilizadas para la obtención de estos gradientes.

-1	0	1
-2	0	2
-1	0	1

$\Delta_x$

-1	-2	-1
0	0	0
1	2	1

$\Delta_y$

Figura 5 Ventanas utilizadas en el método de Sobel para la obtención de los gradientes horizontal  $\Delta_x$  y vertical  $\Delta_y$

Finalmente el gradiente  $\Delta(x,y)$  se obtiene como se mencionó anteriormente a partir de los gradientes horizontal y vertical, definido más específicamente por medio de la siguiente fórmula:

$$\Delta(x,y) = \sqrt{\Delta_x(x,y)^2 + \Delta_y(x,y)^2}$$

El resultado muestra cómo cambia una imagen en cada punto analizado y, en consecuencia, cuán probable es que éste represente un borde en la imagen; y también muestra la orientación que tiende ese borde. En la práctica, el cálculo de la magnitud (probabilidad de un borde) es más fiable y sencillo de interpretar que el cálculo de la dirección y sentido.

### Segmentación

La segmentación de imágenes digitales es una de las etapas más importantes en la implementación de un sistema de reconocimiento de huellas dactilares, el cual tiene como propósito seleccionar solo el área de interés en la imagen (en este caso de una huella dactilar) y desechar lo indeseable como pueden ser manchas, sombras o cualquier otra forma u objeto que no forme parte de la huella dactilar y que podría causar problemas en etapas posteriores en el proceso de reconocimiento.

A continuación, se listan a detalle los principales pasos del algoritmo de segmentación utilizado en este trabajo de tesis [87]:

- 1) Dada una imagen  $I$  de huella dactilar de tamaño  $M \times N$ , se calcula el gradiente horizontal ( $\Delta_x$ ) y vertical ( $\Delta_y$ ) de la imagen original para la obtención del gradiente  $\Delta(x,y)$  mediante el método de Sobel.
- 2) Los valores resultantes del gradiente  $\Delta(x,y)$  deberán ser normalizados, para asegurarse de que se encuentren en un rango de 0 a 255:
  - i. Encontrar el valor mínimo de intensidad en la imagen.



- ii. Restar el valor mínimo encontrado a cada uno de los valores de la imagen.
  - iii. Encontrar el valor máximo de intensidad en los nuevos valores de la imagen y dividirlo entre 255.
  - iv. Por último, será necesario multiplicar cada uno de los valores de la imagen por el resultado obtenido de la división en el paso anterior.
- 3) Se binariza la imagen normalizada, utilizando como umbral el cálculo de la media de la imagen. Es decir, los valores que se encuentren por debajo de dicho umbral serán 0 y los valores que están por encima 255.
  - 4) Se aplica un filtro de promediado a la imagen binarizada, utilizando una ventana de convolución con coeficientes de valor  $1/fc$ , donde  $f$  representa el número de filas de la ventana y  $c$  representa el número de columnas de la ventana.
  - 5) Se binariza nuevamente la imagen filtrada, obteniendo nuevamente la media de la imagen para que ésta sea usada como umbral.
  - 6) Se obtiene la matriz de etiquetas de la imagen binaria utilizando algún algoritmo de etiquetado, para esto se utilizó el mismo algoritmo que en [87, 88].
  - 7) Se encuentra la etiqueta con el mayor número de ocurrencias en la matriz de etiquetas, descartando las pertenecientes al fondo.
  - 8) Se detecta la región más grande y se envían al fondo todas las regiones restantes detectadas, rescatando así sólo la información que pertenece a la huella dactilar.

Todo el proceso anteriormente mencionado dará como resultado una imagen en la que se marcarán con blanco las zonas donde se detectó la presencia de huella dactilar y con negro el resto; siendo de gran utilidad para que los procesos posteriores descarten las zonas donde la presencia de huella dactilar se detectó como nula en la imagen.

### **Binarización Adaptativa**

Para la binarización de la imagen de huella dactilar, en este trabajo de tesis se implementó un método adaptativo donde existe un umbral para el cual todos los valores de los pixeles por debajo de este valor se ponen en cero y para los que están por encima en 255 [86].

Este algoritmo consiste en analizar cada pixel de la imagen en niveles de gris que se quiere binarizar, tomando en cuenta los valores cercanos al pixel analizado en el momento. A continuación, se tienen los pasos utilizados para la binarización adaptativa.

- 1) Sea  $I$  la imagen en niveles de gris que se desea binarizar.
- 2) Se propone una ventana cuadrada alrededor (normalmente de tamaño impar no mayor a la imagen  $I$ ).
- 3) Por cada pixel  $i(x, y)$  en la imagen  $I$ , se crea una matriz (normalmente de tamaño  $n \times n$ , donde  $n$  toma valores impares) con los valores de  $I$  que coinciden con la ventana propuesta, teniendo como valor central el valor del pixel  $i(x, y)$ .

- 4) Para cada matriz creada, se crea un umbral mediante la suma de todos los valores de la matriz y dividiéndola por el número de elementos en la matriz.
- 5) Finalmente el umbral obtenido en cada caso es comparado con el respectivo pixel  $i(x, y)$ , donde si el valor del pixel  $i(x, y)$  es menor al valor del umbral se ponen en 0 y si es mayor se pone en 255.

Para obtener una binarización adecuada es necesario probar el algoritmo con distintos tamaños de ventana y observar el que presenta mejores resultados.

Una variación de este algoritmo es hacer uso del método de Sobel para determinar la orientación de cada pixel en la imagen que servirá de utilidad para que en lugar de crear una ventana cuadrada cuyo ángulo de inclinación es 0, la ventana utilizada para cada pixel  $i(x, y)$  sea una ventana orientada con respecto al ángulo de  $i(x, y)$  [11].

### Adelgazamiento

En imágenes de huella dactilar, el adelgazamiento representa las crestas en las imágenes de huella dactilar como líneas de un pixel de ancho. En el adelgazamiento de huellas dactilares usualmente se implementan operaciones morfológicas como la erosión y la dilatación para reducir el ancho de las crestas a un pixel, preservando la extensión y conectividad original de la forma.

El adelgazamiento se obtiene a partir de una imagen binaria. Para el adelgazamiento se utilizará, en este trabajo de tesis, un método morfológico denominado Guo-Hall que utiliza una serie de operaciones y condiciones para adelgazar la imagen [70].

El algoritmo consiste en una ventana de  $3 \times 3$  que representa una vecindad de 8 pixeles denominada vecindad 8-conectado, cuyo centro es el pixel que se está analizando. En la Figura 6 se muestra una ventana y el orden que se tomará en cuenta para explicar el procedimiento.

P9	P2	P3
P8	P1	P4
P7	P6	P5

Figura 6 Ventana de  $3 \times 3$  donde P1 es el pixel central.

Los pasos del adelgazamiento mediante Guo-Hall, en pseudocódigo, es como sigue:

1. **while** haya puntos eliminados **do**
2. **for** todos los pixeles  $P(i, j)$  **do**
3. **if** (a)  $C(P1) = 1$   
     (b)  $2 \leq N(P1) \leq 3$   
     (c) Aplicar uno de los siguientes:
  1.  $(P2 | P3 | !P5) \& P4 = 0$  en iteraciones impares
  2.  $(P6 | P7 | !P9) \& P8 = 0$  en iteraciones pares
4. **then**

Eliminar el pixel  $P(i, j)$   
 5. end **if**  
 6. end **for**  
 7. end **while**

Donde:

- $P(i, j)$  representa cada pixel de la imagen binaria.
- $C(P1) = !P2 \& (P3 | P4) + !P4 \& (P5 | P6) + !P6 | (P1 | P2)$
- $N1(P1) = (P9 | P2) + (P3 | P4) + (P5 | P6) + (P7 | P8)$
- $N2(P1) = (P2 | P3) + (P4 | P5) + (P6 | P7) + (P8 | P9)$
- $N(P1) = N1(P1) < N2(P1) ? N1(P1) : N2(P1)$

Lo anterior dará como resultado una imagen donde las crestas tengan un pixel de ancho.

### Obtención de minucias

La extracción de características consiste en obtener las minucias de una huella dactilar a partir del esqueleto de la misma obtenida a partir de los procesos anteriormente mencionados.

El algoritmo consiste en calcular el número de pixeles que cruzan el pixel central en una ventana cuadrada de  $3 \times 3$ , que se calcula con la siguiente ecuación [13].

$$Pc = \sum_{i=1}^8 p(i)$$

Donde  $Pc$  es el pixel central y  $p(i)$  el pixel en la posición  $i = 1, 2, \dots, 8$ .

Dependiendo del resultado que se obtenga se tiene lo siguiente:

- Es un bloque con terminación si  $Pc = 7$ .
- Es un bloque sin minucia si  $Pc = 6$ .
- Es un bloque con bifurcación si  $Pc \leq 5$ .

En la ecuación los pixeles  $p(1)$  a  $p(8)$  son una secuencia ordenada de pixeles que definen el bloque de 8 vecinos del pixel central. En la Figura 7 se puede observar la configuración utilizada para localizar bifurcaciones y terminaciones; y en la Figura 8 se pueden apreciar algunas de las posibles configuraciones que se pueden encontrar con este algoritmo:

P1	P2	P3
P8	P0	P4
P7	P6	P5

Figura 7 Ventana de  $3 \times 3$  usada para encontrar minucias.

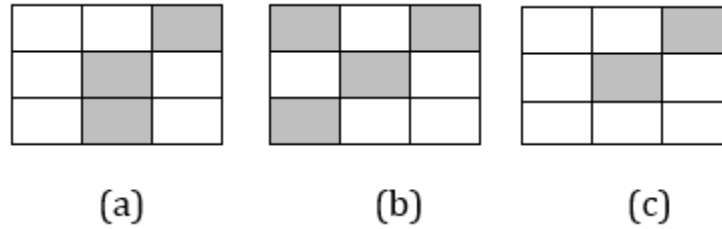


Figura 8 a) Bloque sin minucias, b) Bloque con bifurcación y c) Bloque con terminación.

El algoritmo analiza todos los píxeles de la imagen del esqueleto que no sean parte del fondo, creando una ventana cuadrada de 3x3 alrededor del píxel que se está analizando.

Sin embargo, en ocasiones se pueden encontrar minucias falsas, las cuales deberán ser detectadas y eliminadas para no afectar el resultado cuando éstas sean utilizadas en algún algoritmo de reconocimiento de patrones.

Los tipos más comunes de estructuras que se pueden encontrar de falsas minucias se muestran en la Figura 9. Cada estructura puede generar dos o más minucias falsas [89].

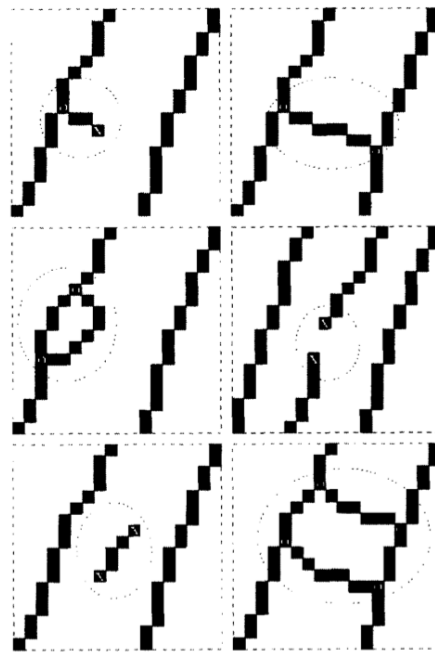


Figura 9 Tipos de estructuras de minucias falsas. De izquierda a derecha y de arriba hacia abajo tenemos: pico, puente, agujero, rotura, espuela y escalera.

A continuación, se explica el problema que se puede generar con cada tipo de estructura de minucia falsa:

- La estructura de pico se puede generar en el adelgazamiento de una cresta no lisa.
- Las estructuras de puente y escalera se suelen producir entre dos crestas que son estrechas.
- Las crestas que son muy anchas pueden generar estructuras ocultas y los valles muy amplios pueden generar espuelas.

- La presencia de cicatrices en la huella dactilar puede crear pausas y crear roturas al realizar el adelgazamiento de la imagen.

Es importante mencionar que la presencia de falsas minucias también se debe a las distintas técnicas de procesamiento de imágenes utilizadas, desde que se obtiene la imagen en escala de grises hasta el adelgazamiento de las crestas en las imágenes de las distintas huellas dactilares.

El algoritmo consiste en los siguientes pasos:

Por cada minucia candidata (terminal o bifurcación):

- 1) Crear e inicializar con 0 una imagen  $L$  de tamaño  $W \times W$ . Cada pixel de  $L$  corresponde a un pixel de la imagen adelgazada compuesta por  $W \times W$  vecinos centrada en la minucia candidata.
- 2) Etiquetar con -1 el pixel central de  $L$  (Figura 10a y 11a). Este es el pixel correspondiente al punto inicial de la minucia candidata en el mapa de la imagen adelgazada.
- 3) Si la minucia candidata es de tipo terminación, entonces:
  - a) Etiquetar con 1 todos los pixeles en  $L$  que correspondan a los pixeles conectados con la minucia de terminación candidata en la imagen adelgazada (Figura 10b).
  - b) Contar el número de transiciones de 0 a 1 ( $T01$ ) conocidos haciendo un recorrido completo en el borde de  $L$  en el sentido de las manecillas del reloj (Figura 10c).
  - c) Si  $T01 = 1$ , entonces la minucia candidata es considerada como una verdadera minucia de tipo terminación.
- 4) Si la minucia candidata es de tipo bifurcación, entonces:
  - a) Hacer un recorrido completo en el sentido de las manecillas del reloj con los 8 pixeles vecinos de la minucia candidata, y etiquetar en  $L$  con 1, 2 y 3 respectivamente los 3 componentes conectados encontrados en el recorrido (Figura 11b).
  - b) Para  $m = 1, 2, 3$  (Figuras 11c, 11d y 11e), etiquetar con  $m$  todos los pixeles de  $L$  en donde:
    - i. Tiene etiqueta 0;
    - ii. Están conectados a un pixel con etiqueta  $m$ .
    - iii. Corresponde a un valor en los pixeles del mapa de la imagen adelgazada.
  - c) Contar el número de transiciones de 0 a 1, 0 a 2 y 0 a 3 conocidos haciendo un recorrido completo en el borde de  $L$  en el sentido de las manecillas del reloj. Esto es denotado por  $T01$ ,  $T02$  y  $T03$  respectivamente y se muestra en la figura 11f.

- d) Si  $T01 = 1$ ,  $T02 = 1$  y  $T03 = 1$ , entonces la minucia candidata es considerada como una verdadera minucia de tipo bifurcación.

La dimensión de  $W$  para el vecindario analizado por cada minucia es escogida tomando 2 veces le promedio de la distancia entre 2 crestas vecinas. De esta manera el algoritmo logra cancelar las minucias cercanas que pertenecen a la misma cresta.

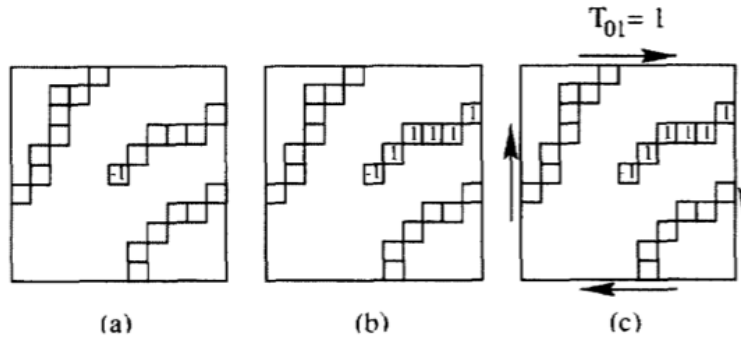


Figura 10 Ejemplo de validación de una minucia de tipo terminal. La figura muestra los cambios en la imagen  $L$  después de diferentes pasos del algoritmo.

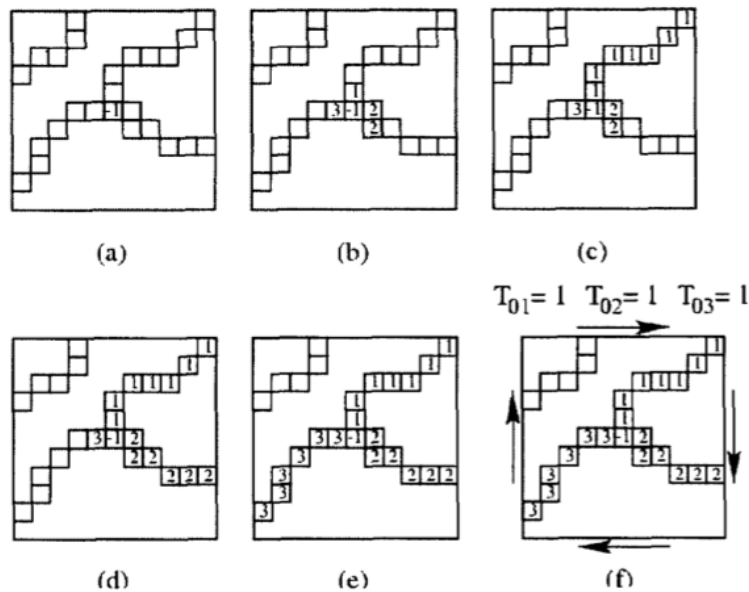


Figura 11 Ejemplo de validación de una minucia de tipo bifurcación. La figura muestra los cambios en la imagen  $L$  después de diferentes pasos del algoritmo.

Finalmente, el resultado de aplicar el método de extracción de minucias y el de detección de las minucias falsas serán las minucias de las cuales se conoce su posición con coordenadas  $(x, y)$ , la orientación obtenida a partir del método de Sobel y el tipo de minucia que es: terminal o bifurcación.

El conjunto de minucias encontradas por imágenes de Huella Dactilar será el patrón que identificará a cada una de las huellas dactilares; este patrón será utilizado a conveniencia por algún clasificador.

### **3.4 Procesamiento Digital de Imágenes de Iris**

En esta sección se presentan los métodos de procesamiento digital de imágenes utilizados en el reconocimiento de iris: la mejora de la calidad de las imágenes, extracción de características y generación de los patrones.

#### **3.4.1 Mejora de la calidad**

El mejoramiento de la calidad de la imagen del iris es una de las etapas fundamentales de muchos modelos de reconocimiento de iris, ya que este mejoramiento permite en gran medida obtener buenos resultados en el reconocimiento. Previo a esto, es necesario segmentar la imagen para encontrar la parte de la imagen que corresponde al iris; esto se logra aproximándose a las fronteras del iris (límbico y pupilar) por medio de un círculo, basándose en la suposición de que el iris es completamente redondo. De este modo, la frontera se describe sólo por dos parámetros: el radio y la posición bidimensional del centro del radio.

Para encontrar estos dos parámetros de la frontera, se propone el uso de la Transformada de Hough para la detección de círculos en la imagen. Este se aplica dos veces, uno para localizar la frontera pupilar y el otro para localizar la frontera límbica [92].

Previo a la detección de la frontera pupilar y frontera límbica mediante la transformada de Hough para círculos y obtener los mejores resultados posibles al momento de detectar ambas fronteras, es necesario preprocesar la imagen, mediante la generación de una imagen binaria, misma que será utilizada para aplicar la transformada de Hough para la detección de las fronteras pupilar y límbica. A continuación, se explica el método de Sobel y la binarización por medio de histéresis que son los dos algoritmos de procesamiento previos a la detección de las fronteras [90, 92].

#### **Binarización mediante histéresis**

Una binarización mediante histéresis requiere dos umbrales (uno alto y otro bajo). Haciendo la suposición de que los bordes deben ser importantes a lo largo de curvas continuas en la imagen, lo cual nos permite seguir una sección débil de una línea determinada y descartar algunos píxeles ruidosos que no constituyen una línea, pero han producido grandes gradientes [90]. Por lo tanto, se comienza aplicando un umbral alto. Esto para marcar los bordes que se tiene por seguro son genuinos. A partir de estos, utilizando la información de la dirección en cada punto de la imagen (obtenida del método de Sobel), los bordes pueden ser rastreados a través de la imagen. Mientras se realiza un recorrido de un borde, se aplica el umbral más bajo, lo que permite trazar secciones débiles de bordes, siempre y cuando nos encontramos con un punto de partida.

Una vez completado este proceso tenemos una imagen binaria, donde cada píxel está marcado ya sea como un píxel de borde o un píxel de no borde. Cabe mencionar que, para reducir el tiempo de procesamiento de la imagen, en este trabajo de tesis la técnica se aplicó sólo en la parte central de la imagen ya que es donde se encuentra el iris, y que es la parte que nos interesa para posteriormente detectar las fronteras pupilar y límbica. También se tomó en cuenta que las imágenes utilizadas cumplieran con esta condición para evitar problemas de que el iris estuviera ubicado en otra zona distinta al centro de la imagen.

## Transformada de Hough

Una vez aplicados los algoritmos de procesamiento anteriores, se tiene lo necesario para aplicar el algoritmo de la transformada de Hough [24, 91], el cual usa una matriz llamada acumulador, cuya dimensión es igual al número de parámetros desconocidos del problema. En este caso, se usará la transformada de Hough para círculos puesto que se quieren encontrar circunferencias en la imagen (frontera pupilar y frontera límbica); entonces, los parámetros son 3 (el radio, el centro en el eje x y el centro en el eje y, todas pertenecientes a la frontera), tomando como referencia la fórmula para la detección de circunferencias que se muestra a continuación:

$$(x - a)^2 + (y - b)^2 = r^2$$

Donde:

- $r$ : radio
- $x$ : posición en el eje x
- $y$ : posición en el eje y
- $a$ : centro de la circunferencia en el eje x
- $b$ : centro de la circunferencia en el eje y

Este algoritmo se aplica sobre la imagen binarizada obtenida mediante la binarización por histéresis dando valores a los 3 parámetros de la transformada de Hough de la siguiente forma: los valores de  $x$  y  $y$  deben estar en el rango del ancho máximo de la imagen y el alto máximo de la imagen respectivamente. Para el caso de  $r$  es conveniente proporcionar un rango aproximado del tamaño de la circunferencia que se quieren detectar (Radio aproximado para la frontera pupilar y radio aproximado para la frontera límbica). El resultado obtenido de esta transformada de Hough será la obtención del radio  $r$ , y las posiciones  $x$  e  $y$  donde se localiza la circunferencia más aproximada encontrada en la imagen binaria con los datos proporcionados, los cuales pueden diferir al cambiar los parámetros proporcionados al algoritmo.

Es importante mencionar que, para agilizar la rapidez de detección de las fronteras límbica y pupilar, se consideró encontrar primero la frontera pupilar, dado que se sabe que el centro de la frontera límbica se encuentra en algún punto dentro de la pupila y que por lo tanto hará que la detección de la frontera límbica sea mucho más rápida.

## Normalización del iris

Para la normalización del iris se realizó un cambio de coordenadas polares a coordenadas rectangulares, tomando como parámetros el centro de la pupila e iris con coordenadas  $(x_0, y_0)$  de la imagen y los radios  $r_p$  y  $r_s$  correspondientes a la frontera pupilar y la frontera límbica respectivamente, que se obtuvieron de la transformada de Hough para círculos. Asumiendo que las coordenadas  $(x_0, y_0)$  pertenecen al centro del iris. Entonces se define a los píxeles de la imagen con el sistema polar de la siguiente manera [91]:

- 1) Se calcula el ángulo como:  
 $\alpha = \text{atan2}(y - y_0, x - x_0), \alpha \in [\pi, -\pi]$



La función atan2 calcula el arco tangente de  $\frac{y-y_0}{x-x_0}$  pero teniendo en cuenta el signo que tengan el numerador y el denominador. De esta forma el ángulo resultante se encontrará en alguno de los cuatro cuadrantes de la circunferencia.

- 2) Se calcula el radio como:

$$r = \sqrt{(x - x_0)^2 + (y - y_0)^2}, r \in [0, \infty)$$

Una vez que se ha definido la imagen con el sistema polar, para llenar los píxeles de la nueva imagen en coordenadas rectangulares a la que llamaremos  $V(x', y')$ , hay que obtener dos variables  $x'$  y  $y'$  de la siguiente manera:

- 1) La coordenada  $x'$  se calcula como:

$$x' = \frac{(M - 1)(\alpha + \pi)}{2\pi}, M \in [0, M - 1]$$

Donde  $M$  corresponde al ancho máximo de la nueva imagen.

- 2) La coordenada  $y'$  se calcula como:

$$y' = \frac{(N - 1)(r - r_p)}{r_s - r_p}, N \in [0, N - 1], r_s > r_p$$

Donde  $N$  corresponde al alto máximo de la nueva imagen,  $r_s$  corresponde al radio límbico y  $r_p$  al radio pupilar.

En la Figura 12 se observan características del proceso de Normalización antes mencionado para la conversión de coordenadas polares a rectangulares.

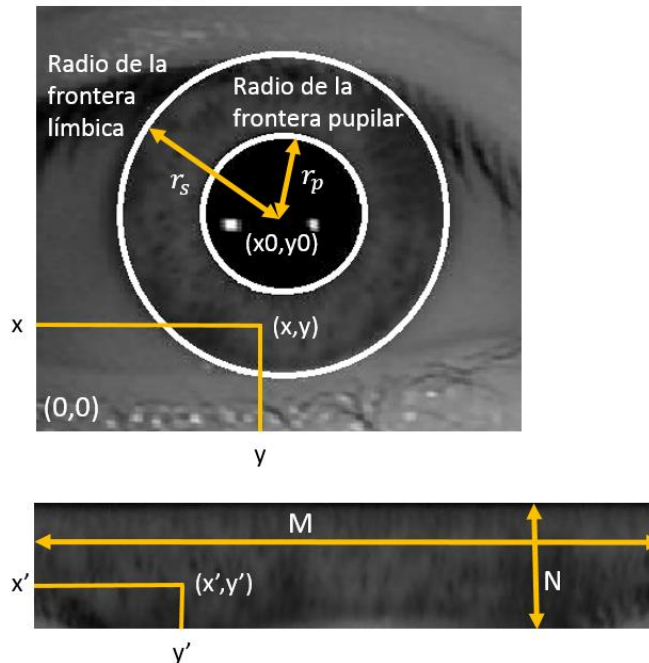


Figura 12 Conversión de coordenadas polares a cartesianas.

En el caso de imágenes de iris no se realiza una extracción de características específicas como ocurre con la huella dactilar, sino que se genera un código que representara a cada iris obtenido a partir de la normalización del mismo. Para esto se hace uso del Filtro del Laplaciano del Gaussiano que permite resaltar los detalles las características que hay en el iris y que no se pueden apreciar tan fácilmente. El uso de este filtro es de gran utilidad al momento de generar el código que servirá para identificar el iris.

### **Filtro del Laplaciano del Gaussiano para resaltar las características del iris**

El Laplaciano de una función bidimensional  $f$  es un operador de derivación isotrópico (independiente de la dirección de la discontinuidad en la imagen) definido por [92, 93]:

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$$

El Laplaciano usualmente no es utilizado directamente en la práctica por ser muy sensible al ruido, por lo que se utiliza sumado o restado (según una ventana usada) con la imagen original para realzar los contornos. Por esta razón, en lugar de utilizar directamente la definición del Laplaciano, se usa primero un filtro Gaussiano para eliminar ruido, lo que da lugar al filtro llamado Laplaciano del Gaussiano denotado como *LoG*, cuyo núcleo puede calcularse componiendo ambos:

$$LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[ 1 - \frac{x^2 + y^2}{2\sigma^2} \right] e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

El resultado de aplicar el Laplaciano del Gaussiano de una imagen destacará las regiones donde hay cambios bruscos de intensidad y por tanto resaltarán las características como surcos, contornos y pecas de pigmento en el iris normalizado.

#### **3.4.2 Generación del código del iris**

Para la generación del código del iris se realizó un procedimiento sencillo en el que hay que tomar en cuenta el tamaño del Gaussiano utilizado anteriormente. El procedimiento es el siguiente:

- 1) Se calculó la media de toda la imagen para obtener un valor que representaremos como  $s$ , que será utilizado como umbral para la generación del código.
- 2) La imagen a la que se le aplicó el Laplaciano del Gaussiano que denominaremos  $L$ , será dividida en bloques de tamaño  $n \times n$  que se recomienda sea igual al tamaño del gaussiano que se utilizó para generar la imagen  $L$ , donde cada bloque corresponderá a un bit del código que será generado.
- 3) Se creará una nueva imagen  $R$  en la que se almacenará el código resultante cuyo tamaño será de  $M \times N$ , donde  $M$  corresponde al número máximo de bloques formados de manera horizontal y  $N$  al número máximo de bloques generados de manera vertical en la imagen  $L$  del paso anterior.
- 4) Para cada bloque  $b$  de la imagen  $L$ , se calculará la media y el valor resultante se comparará con el valor  $s$  obtenido de la media de toda la imagen, si el valor de la media del bloque  $b$  es mayor que el de  $s$ , se colocará un 255 en la posición correspondiente en la imagen  $R$ , en caso contrario se colocará un 0.

Una vez que se hayan analizado todos los bloques tendremos nuestro código generado, destacando que la imagen resultante será menor a la imagen  $L$ , dependiendo del tamaño de bloque que se utilice para realizar este proceso.

Finalmente, el código generado por cada imagen de Iris será el patrón que lo identificará para ser usado por algún clasificador en una forma conveniente.

### 3.5 Máquinas Asociativas Alfa-Beta SVM

Previo a la explicación del algoritmo de las Máquinas Asociativas Alfa-Beta SVM es necesario explicar lo que es una Memoria Asociativa que es sobre lo que se basan fuertemente, los operadores Alfa y Beta que son ampliamente utilizados y el uso de SVMs.

#### 3.5.1 Memorias Asociativas

Una Memoria Asociativa puede formularse, para su operación, como un sistema de entrada y salida, la Figura 13 muestra un esquema básico de lo que es una memoria asociativa:

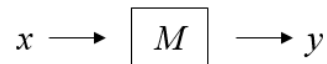


Figura 13 Memoria Asociativa

Los patrones de entrada y salida están representados por vectores columna denotados por  $x$  y  $y$ , respectivamente. Cada uno de los patrones de entrada forma una asociación con el correspondiente patrón de salida, la cual es similar a un par ordenado: por ejemplo, la pareja  $(x, y)$ .

Para realizar una manipulación de los patrones de entrada y de salida, se denotarán con las mismas letras, pero agregando a las misma un superíndice para representar cada asociación. Por ejemplo, un patrón de entrada  $x^2$  es correspondiente a un patrón de salida  $y^2$ , formando la asociación  $(x^2, y^2)$ .

Una memoria asociativa  $M$  se representa mediante una matriz, generada a partir de un conjunto finito de asociaciones conocidas previamente, este conjunto es llamado conjunto de aprendizaje o fundamental. De esta manera, los elementos de la matriz  $M$  cuyos índices  $ij$ , se representarán como  $m_{ij}$ . El conjunto fundamental se representa de la siguiente manera:

$$\{(x^\mu, y^\mu) | \mu = 1, 2, \dots, p\}$$

Donde  $p$  es un número entero positivo que representa la cardinalidad del conjunto fundamental.

Dentro de este conjunto, existe un caso particular para el cual cada patrón de entrada es el mismo que el de salida, es decir,  $x^\mu = y^\mu \forall \mu \in \{1, 2, \dots, p\}$ . Si se dicha condición mencionada anteriormente, se dice que la memoria es autoasociativa; en otro caso se dice que la memoria es heteroasociativa [77].

Una de las ventajas más significativas de las memorias asociativas es la capacidad de trabajar con patrones de entrada ruidosos, es decir, versiones alteradas de  $x$ , cuya notación es una tilde sobre el patrón de la siguiente manera:  $\tilde{x}^k$ , que representaría la versión alterada del patrón original  $x^k$ .

Los tipos de ruido más comunes para patrones binarios son: aditivo, sustractivo y combinado. Cuando se le presenta un patrón ruidoso  $\tilde{x}^w$  como entrada a una memoria  $M$ , y esta responde con el correspondiente patrón de salida fundamental  $y^w$ , se puede decir que la recuperación es correcta; de esta manera, si se cumple para todos los patrones del conjunto fundamental, se dice que la memoria tiene recuperación correcta [94].

### 3.5.2 Operadores Alfa y Beta

Los operadores Alfa ( $\alpha$ ) y Beta ( $\beta$ ) son una parte fundamental para la implementación del clasificador Alfa-Beta SVM. Los operadores descritos son la base de las Memorias Asociativas Alfa-Beta, presentados en [77]. En las Tablas 2 y 3 se presenta la definición de los operadores mencionados, dados los siguientes conjuntos:

$$A = \{0, 1\} \quad B = \{0, 1, 2\}$$

Tabla 2 Definición del operador Alfa

$\alpha: A \times A \rightarrow B$		
$x$	$y$	$\alpha(x, y)$
0	0	1
0	1	0
1	0	2
1	1	1

Tabla 3 Definición del operador Beta

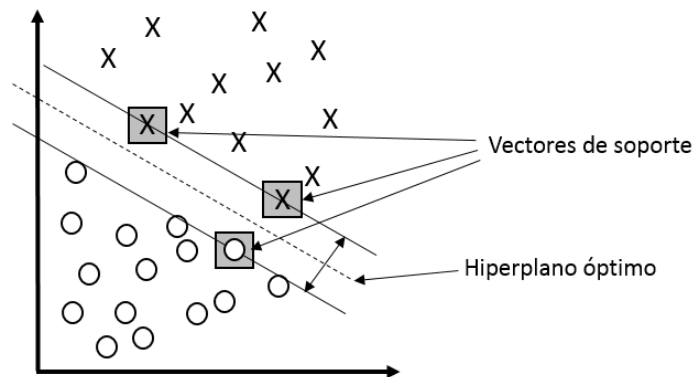
$\beta: B \times A \rightarrow A$		
$x$	$y$	$\beta(x, y)$
0	0	0
0	1	0
1	0	0
1	1	1
2	0	1
2	1	1

### 3.5.3 Support Vector Machines (SVMs)

El algoritmo utilizado por las SVMs funciona como un clasificador biclase que minimiza simultáneamente el error empírico de clasificación y maximiza algunas características de las métricas involucradas [55]. Este algoritmo está fuertemente basado en la teoría de aprendizaje estadístico desarrollada por Vapnik, Chervonenkis y otros, que dio lugar al desarrollo de las SVMs [95, 96].

Este modelo ataca el problema básico de separar un hiperplano n-dimensional en dos clases por medio de un hiperplano n-1-dimensional. Sin embargo, puede existir más de un hiperplano que

cumplan con esta función, por lo que el objetivo de las SVMs es encontrar el hiperplano óptimo que mejor generalice la clasificación. Para esto se utiliza el concepto de vector de soporte, que hace referencia a los patrones más cercanos al hiperplano que se busca, dichos patrones se localizan en la frontera de las clases. Para encontrar el hiperplano óptimo, se maximiza la distancia a los vectores de soporte. Puede darse el caso que existan patrones, cercanos a la frontera de las clases cuyo comportamiento es más como una excepción. Para solucionar este problema, se incluye cierto margen de error, con lo cual se permite la admisión de que ciertos patrones que están cerca de la frontera no sean tomados como vectores de soporte, siempre y cuando se mantengan a una distancia menor o igual al margen de error que se haya establecido. Así, el problema de encontrar el hiperplano que divide las dos clases de manera óptima, se resuelve con la maximización de la distancia entre dicho hiperplano y los patrones más cercanos a la frontera de las clases conocidos como vectores de soporte, a la vez que el error es minimizado [96, 97]. La Figura 14 muestra un ejemplo de SVM.



*Figura 14 Support Vector Machine. Los vectores de soporte están indicados con un recuadro mientras que el hiperplano óptimo aparece como una línea punteada.*

En caso de que el problema no sea linealmente separable en el espacio original, se utiliza una transformación por medio de una función kernel, para llevar los patrones a un espacio en una dimensión mayor, en el cual el problema sea linealmente separable. La Figura 15 muestra un espacio de características de dos dimensiones, donde los datos no pueden ser separados por una función lineal en el espacio de entrada, pero si pueden serlo en el espacio de características, donde la separación es mucho más fácil.

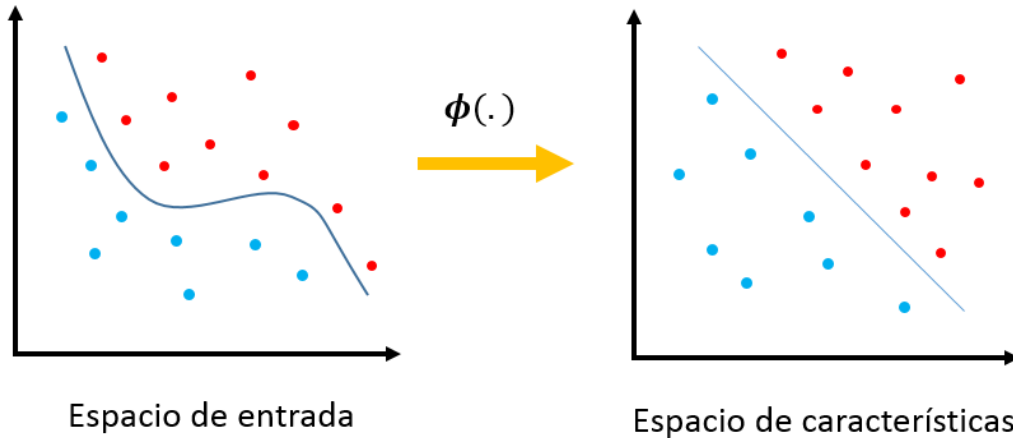


Figura 15 Ejemplo de función kernel

### 3.5.4 Máquinas Asociativas Alfa-Beta SVM

Para explicar el funcionamiento de las Máquinas Asociativas Alfa-Beta SVM, se asume que se tiene un problema de reconocimiento de patrones, donde el conjunto fundamental es de la forma  $\{(x^\mu, y^\mu) | \mu = 1, 2, \dots, p\}$ , con  $x^\mu \in A^n \forall \mu \in \{1, 2, \dots, p\}$ , donde  $n, p \in \mathbb{Z}^+$  y  $A = \{0, 1\}$ . La explicación del algoritmo se divide en dos fases, una de aprendizaje y otra de recuperación [55].

#### Fase de aprendizaje

1. A partir del conjunto fundamental, calcular el vector de soporte  $S$ , de acuerdo con la definición la siguiente definición

Se define el vector soporte  $S$ , como aquel cuyas componentes  $S_i$ , con  $1 \leq i \leq n$  se calculan, a partir del conjunto fundamental, de acuerdo con la siguiente expresión:

$$S_i = \begin{cases} \bigwedge_{k=1}^{p/2} \beta(x_i^{2k-1}, x_i^{2k}) & \text{si } p \text{ es par} \\ \beta \left[ \bigwedge_{k=1}^{(p-1)/2} \beta(x_i^{2k-1}, x_i^{2k}), x_i^p \right] & \text{si } p \text{ es non} \end{cases}$$

2. Para cada  $\mu \in \{1, 2, \dots, p\}$ , obtener  $x^\mu|_S$  de acuerdo con las siguientes dos definiciones:
  - 1) Se define la *Restricción* de  $y$  por  $x$ , y se denota como  $y|x$ , al vector que resulta, a partir del vector  $y$ , de aplicar iteradamente la operación *Eliminación* en  $y$  según  $x$ , para todas y cada una de las componentes de  $x$  con valor 1.
  - 2) Se define la *Restricción* de  $x$  por 0 como:  $x|_0 = x$ ; asimismo, se define la *Expansión* de  $x$  por 0 como:  $x|_0^0 = x$ .

A partir de estos resultados se forma el conjunto fundamental restringido  $\{(x^\mu|_S, x^\mu|_S) | \mu = 1, 2, \dots, p\}$ .

3. Para cada  $\mu \in \{1, 2, \dots, p\}$ , obtener  $\overline{x^\mu}$ , el vector negado de  $x^\mu$ , usando la definición:

Se define el vector negado de  $x$  como el vector denotado por  $\bar{x}$  tal que, para cada índice  $i \in \{1, 2, 3, \dots, n\}$ , las componentes de ambos vectores cumplen la expresión  $\bar{x}_i = \neg x_i$ , donde  $\neg$  es la operación de negación booleana.

Con los  $p$  vectores negados, se forma el conjunto fundamental negado  $\{(\overline{x^\mu}|_S, \overline{x^\mu}|_S) | \mu = 1, 2, \dots, p\}$ .

4. A partir del conjunto fundamental negado, calcular el vector de soporte  $\hat{S}$  de acuerdo a la definición utilizada en el paso 1.
5. Para cada  $\mu \in \{1, 2, \dots, p\}$ , obtener  $\overline{x^\mu}|_{\hat{S}}$  de acuerdo a las definiciones utilizadas en el paso 2. A partir de estos resultados se forma el conjunto fundamental negado restringido  $\{(\overline{x^\mu}|_{\hat{S}}, \overline{x^\mu}|_{\hat{S}}) | \mu = 1, 2, \dots, p\}$ .

### Fase de recuperación

Sea  $\tilde{x} \in A^n$  un patrón de entrada cuyo patrón asociado  $x^\mu$  se desconoce de antemano.

1. Obtener la *Restricción*  $\tilde{x}|_S$  de acuerdo a las definiciones usadas en el paso 2 de la fase de aprendizaje.

2. Para cada  $\mu \in \{1, 2, \dots, p\}$ , obtener  $\tau(\tilde{x}|_S, x^\mu|_S)$  de acuerdo a la siguiente definición:

La transformada  $\tau$  de  $x$  con respecto a  $y$  da como resultado un vector  $\tau(x, y)$  de dimensión  $n$ , cuya  $i$ -ésima componente se calcula como:  $[\tau(x, y)] = \beta[x_i, \alpha(0, y_i)]$ .

3. Para cada  $\mu \in \{1, 2, \dots, p\}$ , obtener  $\tau(x^\mu|_S, \tilde{x}|_S)$  de acuerdo a la definición utilizada en el paso anterior.

4. Para cada  $\mu \in \{1, 2, \dots, p\}$ , obtener  $\theta(\tilde{x}|_S, x^\mu|_S)$  de acuerdo a la siguiente definición:

La transformada de  $\theta$  de  $x$  con respecto a  $y$  es un escalar  $\theta(x, y)$  definido como:  $\theta(x, y) = \sigma_n[\tau(x, y)] + \sigma_n[\tau(y, x)]$ .

5. Encontrar  $\psi \in \{1, 2, \dots, p\}$  tal que  $\theta(\tilde{x}|_S, x^\psi|_S) = \bigwedge_{\mu=1}^p \theta(\tilde{x}|_S, x^\mu|_S)$ .

6. Obtener  $\tilde{\bar{x}}$ , el vector negado de  $\tilde{x}$ , usando la definición utilizada en el paso 3 de la fase de aprendizaje.

7. Obtener la *Restricción*  $\tilde{\bar{x}}|_{\hat{S}}$  de acuerdo con las definiciones usadas en el paso 2 de la fase de aprendizaje.

8. Para cada  $\mu \in \{1, 2, \dots, p\}$ , obtener  $\tau(\tilde{\bar{x}}|_{\hat{S}}, \overline{x^\mu}|_{\hat{S}})$ , de acuerdo a la definición del paso 2 de esta fase.

9. Para cada  $\mu \in \{1, 2, \dots, p\}$ , obtener  $\tau(\overline{x^\mu}|_{\hat{S}}, \tilde{\bar{x}}|_{\hat{S}})$ , de acuerdo a la definición del paso 2 de esta fase.

10. Para cada  $\mu \in \{1, 2, \dots, p\}$ , obtener  $\theta(\bar{x}|\mathcal{S}, \bar{x}^\mu|\mathcal{S})$ , de acuerdo a la definición del paso 4 de esta fase.
11. Encontrar  $\varphi \in \{1, 2, \dots, p\}$  tal que  $\theta(\bar{x}|\mathcal{S}, \bar{x}^\varphi|\mathcal{S}) = \bigwedge_{\mu=1}^p \theta(\bar{x}|\mathcal{S}, \bar{x}^\mu|\mathcal{S})$ .
12. Si  $\theta(\tilde{x}|_S, x^\psi|_S) \leq \theta(\bar{x}|\mathcal{S}, \bar{x}^\varphi|\mathcal{S})$ , realizar la asignación  $\omega = \psi$ ; de otro modo, realizar la asignación  $\omega = \varphi$ .
13. Obtener  $(x^\omega|_S)^S$  que es precisamente el vector  $x^\omega$ , por el siguiente teorema:

De acuerdo con la definición utilizada en el paso 1 de la fase de aprendizaje. La expresión  $(x^\mu|_S)^S = x^\mu$  se cumple para  $\forall \mu \in \{1, 2, \dots, p\}$ .

Si  $\tilde{x}$  es uno de los patrones del conjunto fundamental, el Teorema utilizado en el paso 13 garantiza que la recuperación es correcta.

### 3.6 Clasificador Gamma

Primero se explicaran algunas operaciones y herramientas utilizadas en el clasificador Gamma que son: los operadores Alfa y Beta, los cuales son fundamentales para la implementación del clasificador Gamma y que fueron explicados en el algoritmo de las Máquinas de Soporte Vectorial Alfa-Beta SVM, el operador  $\mathcal{U}_\beta$  utilizado por el clasificador Gamma, la operación de módulo ya que es utilizada por el operador Gamma, el algoritmo del código Johnson-Möbius modificado, ya que los patrones a usar para la clasificación deben estar codificados con este método. Finalmente se describe el clasificador Gamma utilizado en la propuesta de este trabajo de tesis.

#### 3.6.1 Operador $\mathcal{U}_\beta$

En esta sección se presenta el operador  $\mathcal{U}_\beta$ , definido en [63], y que también es utilizado por el clasificador Gamma.

Sea el conjunto  $A = \{0,1\}$ , un número  $n \in \mathbb{Z}$  y  $x \in A^n$  un vector binario de dimensión n, con la  $i$ -ésima componente representada por  $x_i$ . Se define el operador  $\mathcal{U}_\beta(x)$  de la siguiente manera:  $\mathcal{U}_\beta(x)$  tiene como argumento de entrada un vector binario n-dimensional x y la salida es un número entero no negativo que se calcula de la siguiente manera:

$$\mathcal{U}_\beta(x) = \sum_{i=1}^n \beta(x_i, x_i)$$

#### 3.6.2 Operación módulo

El concepto de módulo, al igual que el operador de la sección anterior, son relevantes en el desarrollo del clasificador Gamma.

Hay situaciones en las que, al utilizar la operación de división que se acostumbra, resulta de más interés el residuo (entero) de dicha operación que el resultado (fraccional) mismo. Par solucionar este problema existe el operador módulo, denotado por *mod*.



Sea  $a$  un número entero y  $m$  un número entero positivo. Se denota por  $a \bmod m$  al residuo como resultado de la división de  $a$  por  $m$ .

De otra manera, se puede entender que  $a \bmod m$  es el número entero  $r$  tal que  $a = qm + r$  y  $0 \leq r < m$ .

### 3.6.3 Código binario Johnson-Möbius modificado

Dado que los vectores con que trabaja el clasificador Gamma, son codificados usando el código Johnson-Möbius modificado, en esta sección se describe este método, presentado en [98].

El algoritmo para la generación del código Johnson-Möbius se explica a continuación:

- 1) Sea un conjunto de números reales

$$\{r_1, r_2, \dots, r_i, \dots, r_n\}$$

donde  $n$  es un número entero positivo fijo.

- 2) Si uno de los números del conjunto (por ejemplo  $r_i$ ) es negativo, se crea un nuevo conjunto transformado a través de la resta del valor  $r_i$ , a cada uno de los  $n$  números

$$\{t_1, t_2, \dots, t_i, \dots, t_n\}$$

donde  $t_j = r_j - r_i \forall j \in \{1, 2, \dots, n\}$  y particularmente  $t_i = 0$ . Cabe mencionar que si existe más de un negativo, se trabaja con el menor.

- 3) Se escoge un número fijo  $d$  de decimales y trunca cada uno de los número del conjunto transformado (los cuales son no negativos) con precisamente  $d$  decimales.

- 4) Realizar un escalamiento de  $10d$  en el conjunto del paso 3, para obtener un conjunto de  $n$  enteros no negativos

$$\{e_1, e_2, \dots, e_i, \dots, e_m, \dots, e_n\}$$

donde  $e_m$  es el número mayor.

- 5) El código Johnson-Möbius modificado para cada  $j = 1, 2, \dots, n$  se obtiene al generar  $(e_m - e_j)$  ceros concatenados por la derecha con  $e_j$  unos.

### 3.6.4 Clasificador Gamma

El clasificador Gamma es un clasificador de patrones de alto desempeño, cuyo algoritmo original fue presentado en [63], y que utiliza el operador Gamma de similitud ( $\gamma$ ). El clasificador Gamma también hace uso de los operadores Alfa y Beta, el operador  $\mathcal{U}_\beta$  y los patrones codificados con el código Johnson-Möbius modificado, que fueron descritos en secciones previas. Aquí se introduce primero la definición del operador Gamma de similitud y posteriormente se presenta el algoritmo del clasificador Gamma modificado, que fue presentado en [64].

Antes de presentar el clasificador Gamma es necesario introducir el operador Gamma de similitud, ya que es de gran importancia para la formulación del algoritmo. El operador Gamma de similitud está basado en las operaciones Alfa y Beta de las memorias asociativas Alfa-Beta; este operador indica si dos vectores son parecidos o no, dado un grado de disimilitud  $\theta$ ; que indica la tolerancia para que dos vectores, al compararlos, sean considerados similares, no obstante que son diferentes [63]. A continuación, se define el operador Gamma de similitud.

Sea el conjunto  $A = \{0,1\}$ , un número  $n \in \mathbb{Z}^+$ ,  $x \in A^n$  y  $y \in A^n$  dos vectores binarios n-dimensionales, con la i-ésima componente representada por  $x_i$  y  $y_i$  respectivamente, y además,  $\theta$  un número entero no negativo. Se define el operador Gamma de similitud  $\gamma(x, y, \theta)$  de la siguiente manera:  $\gamma(x, y, \theta)$  tiene como argumentos de entrada dos vectores binarios n-dimensionales  $x$  y  $y$ , y un número entero no negativo  $\theta$ , y la salida es un número binario que se calcula así:

$$\gamma(x, y, \theta) = \begin{cases} 1 & \text{si } n - \mathcal{U}_\beta[\alpha(x, y) \bmod 2] \leq \theta \\ 0 & \text{en otro caso} \end{cases}$$

Sea el conjunto fundamental del clasificador Gamma el conjunto de patrones asociados a una clase, de la forma  $\{(x^\mu, y^\mu) | \mu = 1, 2, \dots, p\}$ ; donde  $x^\mu$  es un patrón y  $y^\mu$  es su clase correspondiente. Además, para este conjunto fundamental se cumplen las siguientes afirmaciones:

1)  $x^i \neq x^j \forall i, j \in \{1, 2, \dots, p\}$  tal que  $i \neq j$

Que implica que no hay patrones repetidos.

2)  $x^i = x^j \implies y^i = y^j \forall i, j \in \{1, 2, \dots, p\}$

Que para un patrón dado, este no puede tener más de una clase asociada.

3)  $y^i \neq y^j \implies x^i \neq x^j \forall i, j \in \{1, 2, \dots, p\}$

Que significa que clases diferentes tienen asociados patrones diferentes.

Dicho de otra manera, el conjunto fundamental debe inducir una relación entre el conjunto de patrones y el conjunto de clases, de tal manera que dicha relación cumpla con las características de una función.

El algoritmo del clasificador gamma se explica a continuación. Sea el conjunto fundamental del clasificador Gamma que cumple con las afirmaciones anteriores. Al presentarse un patrón a clasificar  $\tilde{x}$ , donde  $\tilde{x}$  es un vector real n-dimensional  $\tilde{x} \in \mathbb{R}^n$ , con  $n \in \mathbb{Z}^+$ , se realiza lo siguiente.

1. Codificar cada componente de cada patrón del conjunto fundamental con el código Johnson-Möbius modificado, restando el valor menor a todos los valores por cada componente, con lo cual se obtiene un valor  $e_m = \bigvee_{i=1}^p x_j^i$  por cada componente. Con lo anterior se desplaza el rango de cada componente para que vaya de 0 a  $e_m$ . Transformando así, la componente  $x_j^i$  en un vector binario de dimensión  $e_m(j)$ .
2. Codificar cada componente del patrón a clasificar con el código Johnson-Möbius modificado, utilizando las mismas condiciones utilizadas para codificar las componentes de

los patrones fundamentales. En caso de que alguna componente del patrón a clasificar sea mayor al  $e_m$  correspondiente ( $\tilde{x}_\xi > e_m(\xi)$ ), hay que igualar esa componente a  $e_m(\xi)$  y almacenar su valor anterior en la variable  $mgamma_\xi$ . Por otro lado, si alguna componente da un valor negativo una vez desplazada, hay que igualar esa componente a 0 y asignar el valor  $e_m(\xi) + |\tilde{x}_\xi|$  a  $mgamma_\xi$ .

3. Calcular el parámetro de paro  $p$  y el parámetro de pausa  $p_0$ . Dependiendo del problema a tratar, algunas posibilidades sugeridas para estos parámetros son las siguientes:

- a)  $p = \bigwedge_{j=1}^n (V_{i=1}^p x_j^i)$ .

- b)  $p = \frac{1}{n} \sum_{j=1}^n (V_{i=1}^p x_j^i)$ .

- c)  $p = \bigvee_{j=1}^n (V_{i=1}^p x_j^i)$ .

- d)  $p_0 = \bigwedge_{j=1}^n (V_{i=1}^p x_j^i)$ , sobre todo si  $p = \bigvee_{j=1}^n (V_{i=1}^p x_j^i)$ .

- e)  $p_0 > p$ , cuando se desea asignar forzosamente una clase conocida a los patrones desconocidos.

4. Determinar el umbral de pausa  $u$ . Considerando que el valor de este umbral depende fuertemente de las características del problema y las propiedades del conjunto fundamental, se ofrecen las siguientes sugerencias como valores iniciales:

- a)  $u = 0$ .

- b)  $u = n$ .

5. Determinar los pesos de cada dimensión  $w_i \in \mathbb{R}^+ | i = 1, 2, \dots, n$ . A falta de afinar estos pesos a las características del problema y las propiedades del conjunto fundamental, se sugieren los siguientes rangos como valores iniciales empírico:

- a) Dentro del rango  $[1.5, 2]$  a las dimensiones que sean puntualmente separables para todas las clases.

- b) Dentro del rango  $[1, 1.5]$  a las dimensiones que sean puntualmente separables para algunas clases o bien, que sean puntualmente segmentables para todas las clases.

- c) Dentro del rango  $[0.8, 1.2]$  a las dimensiones que sean puntualmente segmentables para todas o algunas clases.

- d) Dentro del rango  $(0, 0.5]$  a las dimensiones que sean puntualmente no separables.

6. Realizar una transformación de índices en los patrones del conjunto fundamental, de manera que el índice único que tenía un patrón originalmente en el conjunto fundamental, por ejemplo  $x^\mu$ , se convierta en dos índices: uno para la clase (por ejemplo la clase  $i$ ) y otro para el orden que le corresponde a ese patrón dentro de esa clase (por ejemplo  $\omega$ ). Bajo

estas condiciones ejemplificadas, la notación para el patrón  $x^\mu$  será ahora, con la transformación,  $x^{i\omega}$ . Lo anterior se realiza para todos los patrones del conjunto fundamental.

7. Inicializar  $\theta$  en 0.
8. Realizar la operación  $\gamma_g(x_j^\mu, \tilde{x}_j, \theta)$  para cada componente de cada uno de los patrones fundamentales y del patrón a clasificar, considerándose  $m\gamma_{\xi}$  como la dimensión del patrón binario  $\tilde{x}_\xi$ , en caso necesario.
9. Calcular la suma ponderada inicial  $c_i^0$  de los resultados obtenidos en el paso 8, para cada patrón fundamental  $\mu = 1, 2, \dots, p$ :

$$c_i^0 = \sum_{j=1}^n \omega_j * \gamma_g(x_j^\mu, \tilde{x}_j, \theta)$$

10. Si existe un máximo único, cuyo valor se además igual a  $n$ , asignar al patrón a clasificar la clase correspondiente a ese máximo:

$$\tilde{y} = y^j \text{ tal que } \bigvee_{\mu=1}^p c_\mu^0 = c_j^0 = n$$

De lo contrario, continuar.

11. Realizar la operación  $\gamma_g(x_j^{i\omega}, \tilde{x}_j, \theta)$  para cada clase y para cada componente de cada uno de los patrones fundamentales que corresponden a esa clase, y del patrón a clasificar, considerándose  $m\gamma_{\xi}$  como la dimensión del patrón binario  $\tilde{x}_\xi$  si es necesario.
12. Calcular la suma ponderada  $c_i$  de los resultados obtenidos en el paso 11, para cada clase  $i = 1, 2, \dots, m$ :

$$c_i = \frac{\sum_{\omega=1}^{k_i} \sum_{j=1}^n \omega_j * \gamma_g(x_j^{i\omega}, \tilde{x}_j, \theta)}{k_i}$$

13. Si existe más de un máximo entre las sumas ponderadas por clase, incrementar  $\theta$  en 1 y repetir los pasos 11 y 12 hasta que:
  - a) Exista un máximo único;
  - b) O se cumpla con la condición de pausa:  $\theta = p_0$ ;
  - c) O se cumpla con la condición de paro:  $\theta \geq p$ .
14. Si se cumple con la condición de pausa  $\theta = p_0$ , se compara el valor máximo de las sumas ponderadas con el umbral de pausa.

a) Si  $\bigvee_{i=1}^m c_i \leq u$  entonces se asigna la clase desconocida al patrón a clasificar:

$$C_{\tilde{x}} = C_0$$

b) Si  $\bigvee_{i=1}^m c_i > u$  entonces se continúa en el paso 11.

15. Si existe un máximo único, asignar al patrón a clasificar la clase correspondiente a ese máximo:

$$\tilde{y} = y^j \text{ tal que } \bigvee_{i=1}^m c_i = c_j$$

16. En caso contrario: si  $\lambda$  es el índice más pequeño de la clase que corresponde a uno de los máximos, asignar al patrón a clasificar la clase  $\tilde{y} = y^\lambda$ .

### 3.7 MVC

El Modelo Vista Controlador (MVC) es una arquitectura de software que separa los datos y la lógica de negocio de la interfaz de usuario, en tres componentes. Esta arquitectura normalmente se ve aplicada en aplicaciones web y móviles, donde la vista representa a las páginas HTML en las aplicaciones web y a las pantallas creadas en XML en las aplicaciones móviles, el modelo representa las entidades utilizadas y la manipulación de datos, y por último el controlador que es la lógica de negocio de las aplicaciones.

La forma en que interactúan las tres componentes se da de la siguiente forma: las vistas acceden a la lógica de negocio por medio del controlador para ejecutar operaciones de manipulación de datos por medio de entidades definidas en el modelo. La Figura 16 muestra un diagrama de lo antes explicado:

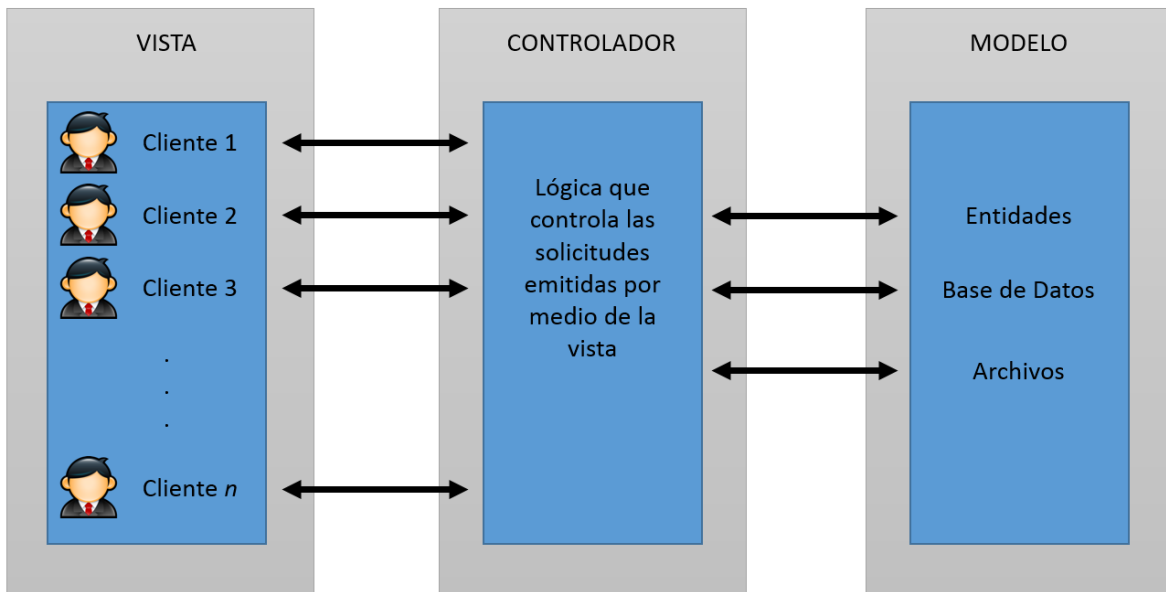


Figura 16 Diagrama general de la arquitectura MVC

En la actualidad se han desarrollado un conjunto de frameworks que se adaptan a esta arquitectura y otros que sólo están orientadas a alguna de sus componentes, en las aplicaciones web. Entre los frameworks open-source que implementan esta tecnología se encuentran:

- Spring MVC
- JSF
- Struts
- Hibernate
- Primefaces

### **3.8 Base de Datos NoSQL**

Como su nombre lo indica, NoSQL es una familia de Sistemas Manejadores de Bases de Datos (DBMS) que no usan el lenguaje SQL, y muchas de las veces no implican el uso del modelo relacional. El almacenamiento también es manejado de forma diferente ya que no utiliza esquemas fijos, y permiten un escalamiento horizontal. Este tipo de bases de datos están optimizadas para la ejecución de operaciones de lectura y escritura de datos, y son capaces de trabajar con grandes cantidades de datos, haciéndolas ideales para resolver ciertos tipos de problemas [99].

Aunque existen diferentes tipos de bases de datos NoSQL, la forma ms común es la que permite almacenar los datos mediante el modelo de clave-valor codificados en algún formato estándar que puede ser XML o JSON. Este modelo permite almacenar entidades con diferente número de atributos y diferentes tipos de datos, a diferencia de las bases de datos convencionales.

Otra de las diferencias de las bases de datos convencionales y las bases de datos NoSQL, es que estas últimas no garantizan completamente ACID (Atomicidad, Coherencia, Aislamiento y Durabilidad). Sin embargo, esto es intercambiado por la alta velocidad en lectura de datos.

En resumen, este tipo de DBMS están orientadas a problemas muy específicos, donde se cumplan los siguientes aspectos:

- Se requiera trabajar con grandes cantidades de información.
- Se requiera un acceso rápido a los datos.
- Se necesita trabajar con tablas de columnas variables.
- No se realicen operaciones de escritura y eliminación frecuentes.
- No se requieran operaciones transaccionales.

## Capítulo 4

### 4. Solución Propuesta

En el presente capítulo se detalla la arquitectura, métodos y algoritmos para el reconocimiento de personas mediante un sistema multibiométrico. En la primera sección se describen los métodos de procesamiento digital de imágenes aplicados a imágenes de iris y huellas dactilares, así como la generación de los patrones en cada caso, remarcando las consideraciones tomadas para la aplicación de los mismos. En la siguiente sección se abordan los algoritmos de clasificación propuestos y su aplicación sobre los patrones generados en la sección anterior. En secciones posteriores se abordarán los temas referentes a las tecnologías que se utilizaron para la implementación del sistema sobre una plataforma en la nube.

#### 4.1 Procesamiento digital y generación de patrones a partir de imágenes de Iris y Huella Dactilar

En esta etapa del proceso de reconocimiento multibiométrico (Iris y Huella Dactilar) para la identificación y verificación de personas, se eligieron ciertos métodos de procesamiento digital de imágenes y técnicas para la generación de patrones para Iris y Huella Dactilar que han demostrado ser de utilidad. Las consideraciones tomadas en la elección de estos métodos es que ayudan significativamente con su aplicación en un sistema de reconocimiento a la tasa de falsos positivos (TFP) y la tasa de verdaderos negativos (TVN).

Para el iris, los métodos de procesamiento digital y técnica para la generación de patrones son:

- Método de Sobel
- Binarización mediante Histéresis
- Transformada de Hough
- Normalización del Iris
- Laplaciano del Gaussiano
- Generación del código de Daughman

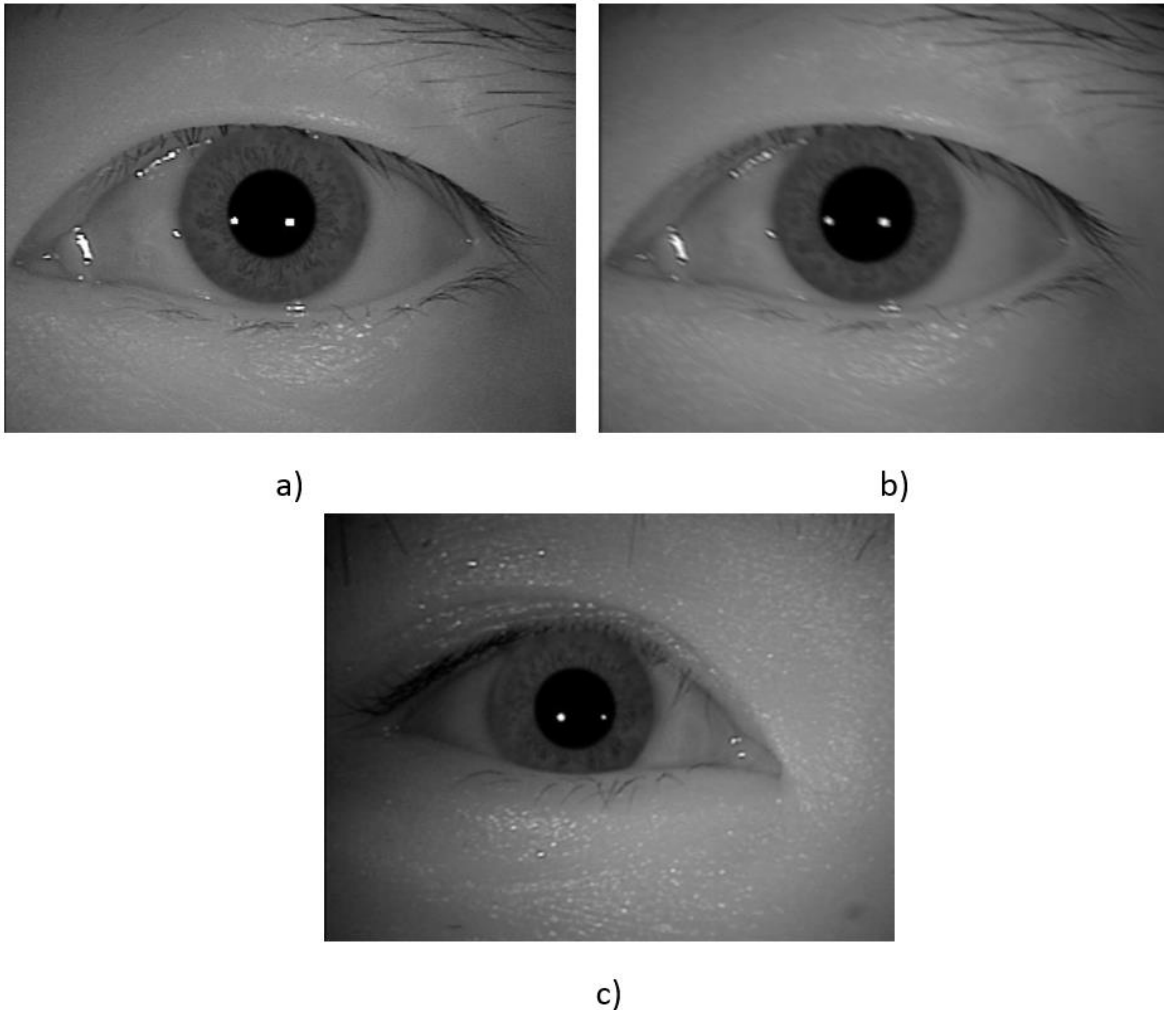
Para la huella dactilar, los métodos de procesamiento digital y técnica para la generación de patrones son:

- Ecuilización del Histograma
- Transformada de Fourier
- Segmentación basada en el método de Sobel
- Binarización Adaptativa
- Adelgazamiento por Guo-Hall
- Extracción de Minucias

##### 4.1.1 Iris

En esta sección se explica a detalle la aplicación de los métodos de procesamiento digital de imágenes propuestos para imágenes de iris. Para la explicación de los mismos, se apoyará de algunas imágenes pertenecientes al banco de datos de imágenes de iris propuesto (CASIA-Iris-Syn

Versión 4). En la Figura 17 se muestran 3 imágenes de Iris que se utilizarán a lo largo de esta sección para detallar la aplicación de todos los métodos.



*Figura 17 Imágenes digitales de Iris pertenecientes a la base de datos CAISA-Iris-Syn.*

El primer método aplicado es el de Sobel, cuya aplicación principal es la detección de bordes en las imágenes, en este caso lo que nos interesa es que el método logre detectar los bordes en imágenes de iris, en donde los bordes pertenecientes a las fronteras pupilar y límbica están incluidos puesto que es lo que se quiere lograr con la aplicación de este método. La Figura 18 muestra la aplicación del algoritmo del método de Sobel aplicado sobre las imágenes mostradas en la Figura 17.



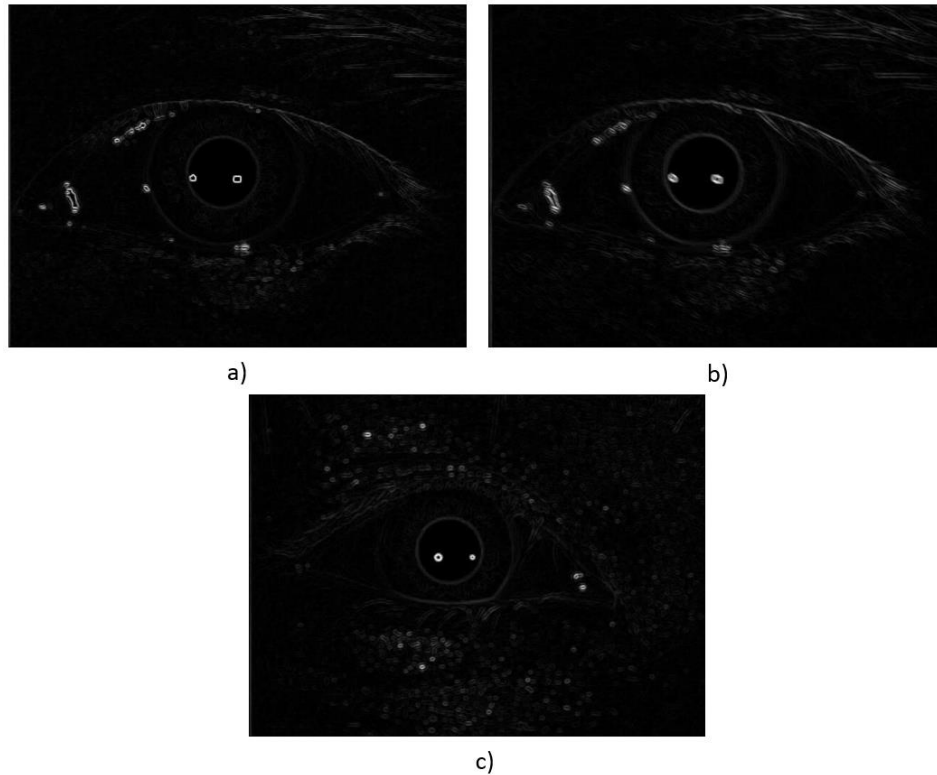


Figura 18 Método de Sobel aplicado a las imágenes de la Figura 17.

El siguiente método aplicado es la Binarización mediante Histéresis, el cual es aplicado sobre imágenes que hayan pasado por un algoritmo que permita la detección de bordes, en este caso el uso del método de Sobel, y el cual es útil para eliminar discontinuidades en los bordes que se encuentren en la imagen proporcionada. Con este algoritmo lo que se busca es generar una imagen binarizada en la que estén resaltados los bordes pertenecientes a la frontera pupilar y límbica. Las Figuras 19 y 20 muestran el resultado de aplicar la binarización mediante histéresis cuyos parámetros proporcionados para su ejecución fueron: 5 para el umbral bajo y 25 para el más alto que fueron los valores con los que se obtuvieron mejores resultados.

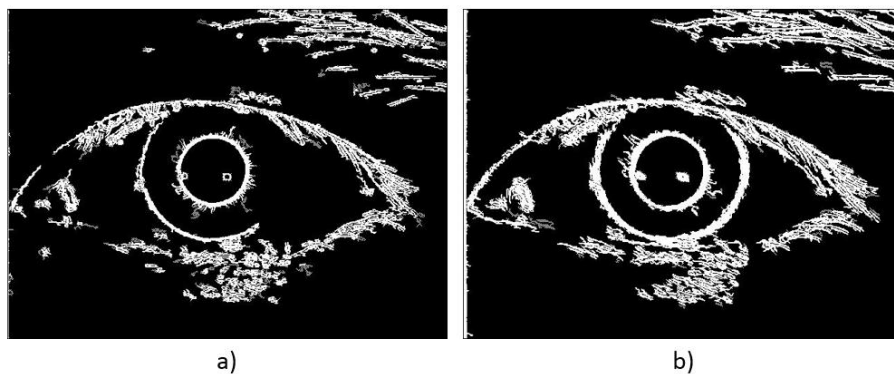
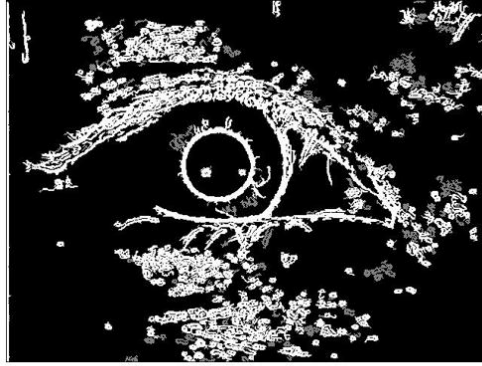


Figura 19 Aplicación de la binarización mediante histéresis sobre las imágenes a) y b) de la Figura 18.



c)

Figura 20 Aplicación de la binarización mediante histéresis sobre la imagen c) de la Figura 18.

Con esto ha llegado el momento de detectar las fronteras límbica y pupilar por medio de las imágenes obtenidas de la binarización mediante histéresis. Para esto se aplicará la Transformada de Hough para la detección de círculos en imágenes binarias. La Transformada de Hough recibe 3 argumentos: dos argumentos que le permiten delimitar el área de búsqueda de círculos en la imagen y el otro es el tamaño aproximado del radio del círculo que se quiere detectar, en este caso los círculos correspondientes a las fronteras pupilar y límbica. El algoritmo proporcionará el valor del radio  $r$  del círculo y los valores  $x_0$  y  $y_0$  pertenecientes al centro del círculo con base en el acumulador utilizado por el algoritmo de la transformada de Hough. Sin embargo, este proceso aún es manual ya que para que fuese automático y lo más óptimo posible aplicándolo de esta forma para cada imagen de iris, habría que indicar el radio más próximo a la frontera pupilar o frontera límbica, según sea el caso. Por lo que para hacerlo automático se modificó el algoritmo de la transformada de Hough añadiendo otro parámetro  $p$ , de modo que en lugar de analizar un solo radio, sean analizados un conjunto de radios en el rango de  $[r - p, p + r]$ , siendo ambos parámetros del algoritmo. Esto permitirá que el acumulador utilizado por el algoritmo de la transformada trabaje sobre el todo el conjunto de radios, dando como resultado un radio  $r$  y los valores  $x_0$  y  $y_0$  considerados más óptimos para el algoritmo. Las Figuras 21 y 22 muestran la detección de las fronteras pupilar y límbica mediante círculos dibujados en las imágenes originales cuyos parámetros fueron obtenidos del algoritmo propuesto de la Transformada de Hough.

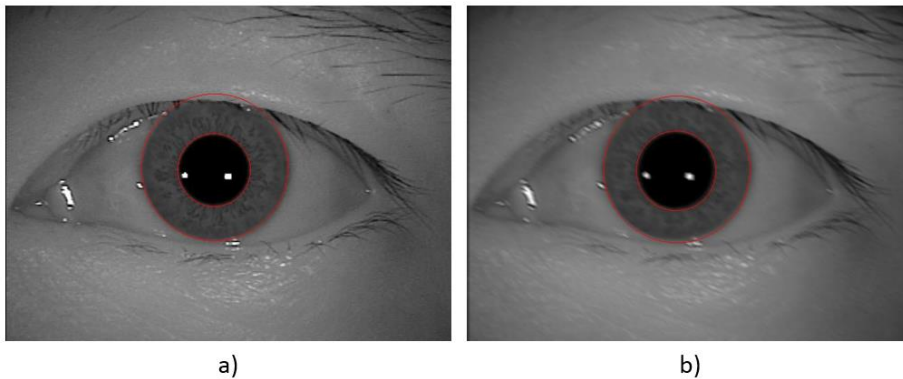
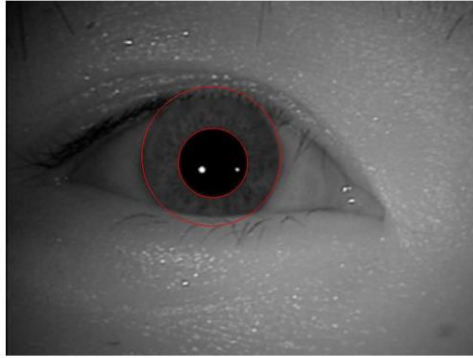


Figura 21 Frontera pupilar y frontera límbica detectadas en las imágenes a) y b) de la Figura 17 por medio de la Transformada de Hough.



c)

*Figura 22 Frontera pupilar y frontera límbica detectadas en la imagen c) de la Figura 17 por medio de la Transformada de Hough.*

Al aplicar la transformada de Hough para la detección de círculos para la obtención de los radios de la frontera pupilar y frontera límbica, denotados por  $r_p$  y  $r_l$ , respectivamente. Así, como el centro de los mismos en las posiciones  $(x_{p_0}, y_{p_0})$  para la frontera pupilar y  $(x_{l_0}, y_{l_0})$  para la frontera límbica.

Lo siguiente es normalizar el iris haciendo uso de los parámetros encontrados en el proceso anterior. La normalización consiste en realizar un cambio de coordenadas polares a rectangulares, que informalmente es desenvolver el iris para que quede en forma de un rectángulo. El algoritmo de la normalización hace uso de los radios y centros de la frontera pupilar y frontera límbica encontrados en la imagen analizada. La Figura 23 muestra el resultado de aplicar la normalización del iris en las imágenes originales haciendo uso de los parámetros encontrados en el proceso anterior.



a)



b)



c)

*Figura 23 Imágenes de iris normalizados pertenecientes a las imágenes de la figura 17.*

El siguiente paso, previo a la generación del código del iris, es la aplicación de un filtro Laplaciano del Gaussiano que permitirá resaltar las características presentes en las imágenes de iris

normalizadas, lo cual proporcionará información muy útil para poder diferenciar el un iris de otro por medio de su codificación. Los parámetros proporcionados al algoritmo son 8 y 1.5, siendo el primero el tamaño del Gaussiano utilizado por el filtro del Laplaciano y el segundo un parámetro utilizado para aplicar el Laplaciano y del cual dependerá en gran medida que tan resaltadas quedarán las características del iris y también del tamaño del código del iris que será generado. La Figura 24 muestra la aplicación de este filtro sobre las imágenes obtenidas con la normalización de iris.

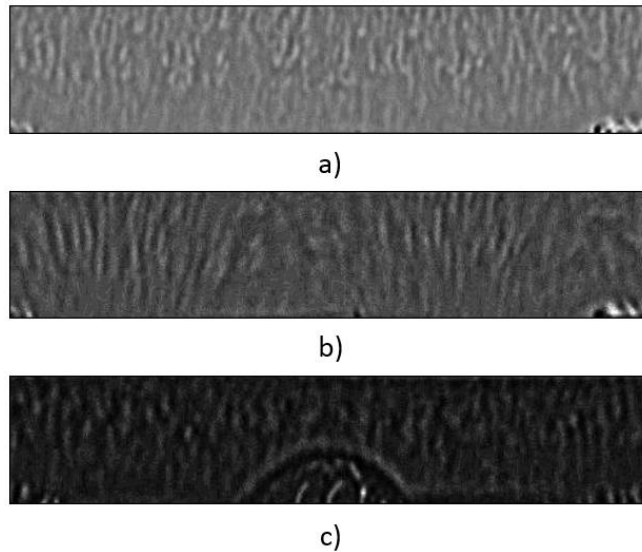


Figura 24 Aplicación del filtro Laplaciano del Gaussiano en las imágenes de iris normalizados de la Figura 23.

Finalmente, para la generación del código del iris, se hace uso de imágenes que han sido filtradas previamente por medio del filtro Laplaciano del Gaussiano para que las características del iris sean más notables y por lo tanto el código de iris que se genere posea la mayor cantidad posible para que sea diferenciada de otros códigos de iris. Para la generación del código del iris se debe proporcionar al algoritmo de generación de código un parámetro que determinara el tamaño del código de iris, se considera que el valor de este parámetro debe ser cercano al tamaño del parámetro utilizado para calcular el Gaussiano en el filtro del Laplaciano del Gaussiano. La Figura 25 muestra los códigos de iris generados utilizando un valor de 8 como parámetro para el algoritmo de codificación.

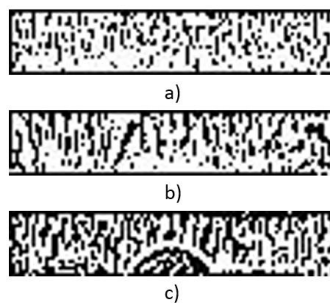


Figura 25 Códigos de iris generados a partir de las imágenes de la Figura 24.

### 4.1.2 Huella Dactilar

En esta sección se explica a detalle la aplicación de los métodos de procesamiento digital de imágenes propuestos para imágenes de huella dactilar. Para la explicación de los mismos, se apoyará de algunas imágenes pertenecientes al banco de datos de imágenes de huella dactilar propuesto (CASIA-Fingerprint Versión 5). En la Figura 26 se muestran 3 imágenes de huella dactilar que se utilizarán a lo largo de esta sección para detallar la aplicación de todos los métodos.



*Figura 26 Imágenes digitales de Huella Dactilar pertenecientes a la base de datos CAISA-Fingerprint.*

Para la mejora de la imagen se utilizó el algoritmo de la Transformada Discreta de Fourier (TDF), el cual permite mejorar el contraste de las imágenes de huella dactilar, de tal forma que la distinción entre crestas y valles sea más evidente. El uso de la TDF es fundamental en el modelo propuesto para el reconocimiento de huellas dactilares ya que su implementación permite reducir significativamente la tasa de verdaderos negativos y la tasa de falsos positivos. Lo suele ser un problema común en el desarrollo de sistemas biométricos. En la Figura 27 se muestra el resultado de aplicar el algoritmo de la TDF. En dichas imágenes se puede apreciar el evidente potencial que tiene la TDF para mejorar el contraste en imágenes de huella dactilar.



*Figura 27 Transformada de Fourier aplicada a las imágenes de la Figura F1.*

En conjunto con la TDF se aplicó otro algoritmo cuya función principal es la misma que la de la DTF que es la mejora del contraste en imágenes, no necesariamente de imágenes de huella dactilar. Este algoritmo es el de ecualización del histograma, cuyo funcionamiento consiste en obtener el histograma de la imagen y lo estira para que ocupe todo el rango de valores permitido en una imagen en niveles de gris que es de 0 a 255. Dicha aplicación se observa en la Figura 28 posterior a la TDF.



*Figura 28 Ecualización del Histograma aplicado a las imágenes de la Figura 27.*

La siguiente etapa del proceso es el de la segmentación, cuya etapa es muy importante dado que permite detectar las zonas donde existe huella dactilar, descartando las zonas donde no existe. Esto es muy importante debido a que la presencia de objetos o formas que no pertenecen a la huella dactilar pueden provocar ruido en las etapas posteriores del reconocimiento de huellas dactilares, impidiendo el correcto reconocimiento de las huellas dactilares. En la Figura 29 se muestra el resultado de la segmentación de imágenes de huellas dactilares aplicado sobre las imágenes originales donde se visualiza solo la presencia de características pertenecientes a la huella dactilar presente en la imagen.



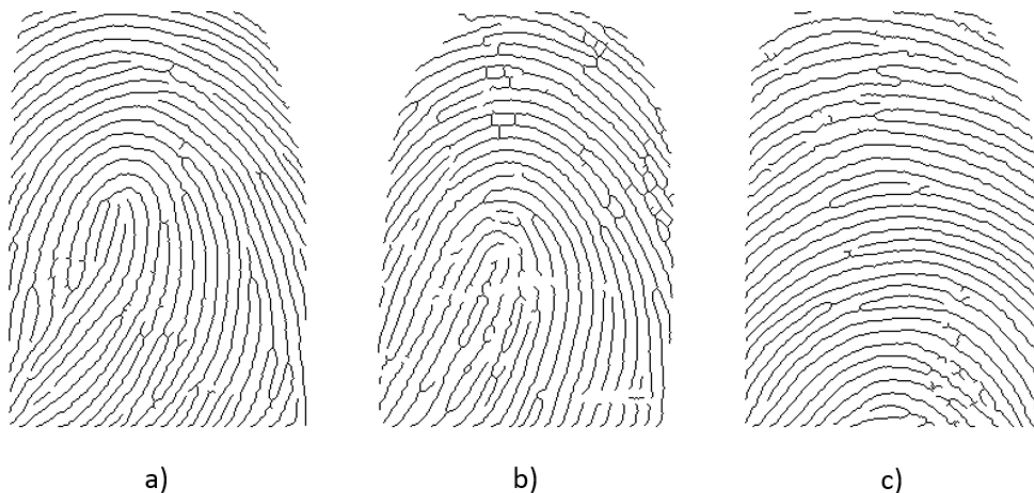
*Figura 29 Imagen que refleja la segmentación aplicada sobre las imágenes originales de la Figura F1, donde se puso en blanco las zonas que no pertenecen a la huella dactilar.*

El siguiente paso es la binarización por medio de un algoritmo adaptativo aplicado sobre las imágenes mejoradas mediante la DTF y la ecualización del histograma, tomando en cuenta la información proporcionada por la segmentación, con la finalidad de analizar solo las zonas donde existe huella dactilar y descartando las demás y dejándolas con un valor de 255 para denotar la no presencia de información. En la Figura 30 se muestra el resultado de aplicar la binarización adaptativa sobre imágenes previamente mejoradas.



*Figura 30 Aplicación de la binarización adaptativa a imágenes previamente mejoradas con la TDF y ecualización del histograma mostradas en la Figura 29.*

El adelgazamiento consiste en preservar la estructura de la imagen binarizada, pero reduciendo a un pixel de grosor cada cordillera que se encuentre en la imagen. Para el adelgazamiento de la imagen se utilizó el método de Guo-Hall el cual analiza la imagen previamente binarizada, y aplica reglas de forma iterativa sobre la imagen donde se detectó la presencia de huella dactilar para ir reduciendo poco a poco el ancho de las crestas de la huella dactilar hasta que ninguna regla pueda aplicarse y por ende las iteraciones terminen. El resultado de utilizar este método se puede visualizar en la Figura 31.



*Figura 31 Esqueleto generado por medio del método Guo-Hall aplicado a las imágenes de la Figura 30.*

Para la detección de las minucias se realizó un algoritmo sencillo que consistió en analizar cada pixel de la imagen adelgazada y verificar el número de ramificaciones encontradas en cada caso. Si el número de ramificaciones era igual a 1, se consideró como minucia de tipo terminación, si se encontró que tenía 3 ramificaciones se consideró como minucia de tipo bifurcación. Con la finalidad de eliminar minucias falsas se implementó un algoritmo para dicha tarea que se encarga de analizar las minucias encontradas por medio de ventanas cuadradas alrededor de las mismas, y por medio de reglas tomar la decisión de si una minucia es falsa. En la Figura 32 se muestra el resultado de obtener las minucias en imágenes previamente adelgazadas. Donde se marcan con puntos rojos las minucias de tipo bifurcación y con azul las minucias de tipo terminación encontradas en cada caso.

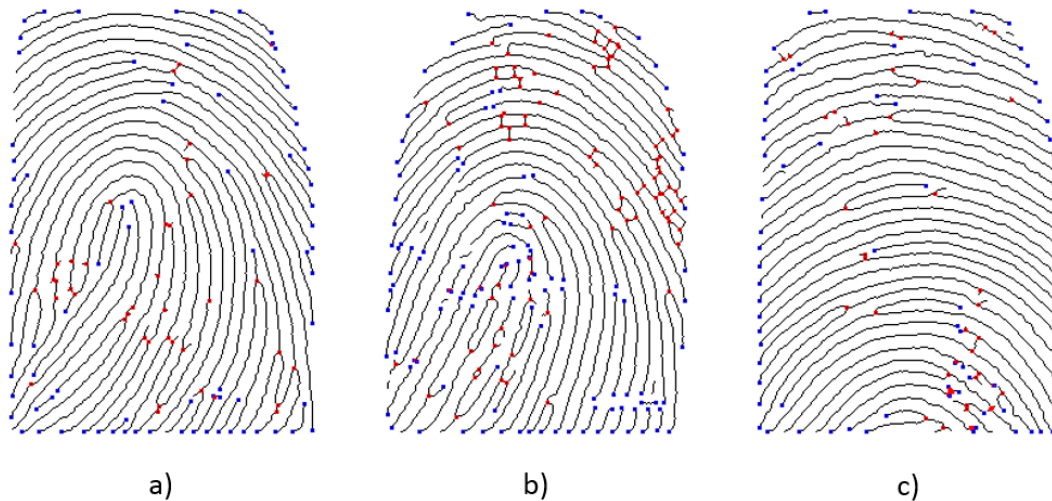


Figura 32 Imágenes de huellas dactilares en las que se detectaron minucias de tipo bifurcación marcadas con puntos rojos y de tipo terminación marcadas con puntos azules.

## 4.2 Algoritmos de Clasificación para reconocimiento de Iris y Huella Dactilar

En este trabajo de tesis se utilizaron dos algoritmos de clasificación distintos para el reconocimiento de iris y huella dactilar. Dichos algoritmos corresponden a las Memorias Alfa-Beta SVM para el iris y el clasificador Gamma para las huellas dactilares.

El hecho de utilizar diferentes algoritmos, es debido a que el rendimiento experimental obtenido para cada conjunto de patrones, es el mejor en ambos casos, para cada uno de los modelos. Es decir, tanto las memorias Alfa-Beta SVM fueron mejores para el reconocimiento de Retina, como el Clasificador Gamma para las huellas.

A continuación, se detalla la metodología utilizada para el uso de estos algoritmos en las tareas de reconocimiento biométrico.

### 4.2.1 Iris

En el caso de la detección de retina por medio del iris, se utilizó el algoritmo de clasificación Alfa-Beta SVM debido a su robustez ante el ruido en imágenes.



Las máquinas de soporte vectorial Alfa-Beta es un modelo de clasificación binaria que permiten separar linealmente conjuntos de datos de manera altamente eficiente, las cuales hacen uso del concepto de vectores de soporte para formar una memoria autoasociativa que permita recuperar patrones [55].

La idea de este tipo de memorias es eliminar información repetida en el conjunto fundamental; en este caso, dicha información corresponde a la parte que hay en común entre todos los patrones del conjunto de entrenamiento. De esta manera, se genera un patrón con la información repetida, el cual se ha llamado vector de soporte [56].

En la fase de recuperación, cuando se presenta un patrón desconocido, es decir, que no está en el conjunto fundamental, primeramente, se elimina la información correspondiente al vector de soporte y posteriormente se busca aquel vector del conjunto fundamental que tenga menos diferencias. Una vez encontrado éste patrón, se vuelve a colocar la información que eliminada (repetida) y se devuelve como resultado de la recuperación [57].

Lo anterior facilita que los códigos generados para el iris sean patrones ideales para la aplicación de este clasificador. Partiendo de la imagen de la figura 33, se describe el uso del clasificador Alfa-Beta SVM con el uso de códigos generados para iris.



*Figura 33 Código de iris generado a partir del preprocesamiento de una imagen de un ojo.*

Como se observa, el código del iris solo está representado por dos valores (0 que representa el color negro y 1 que representa el color blanco) que es justo lo que se necesita para que el clasificador basado en las máquinas Alfa-Beta SVM funcione sin necesidad de modificarlo.

Como los códigos de iris obtenidos son imágenes, para su uso en el clasificador se realizó una conversión de los mismos a un código representado por una cadena de números (0's y 1's). En la Figura 34 se ejemplifica la conversión de una imagen de código de iris en una cadena de números que resultará en el patrón que represente a dicha imagen en el clasificador.

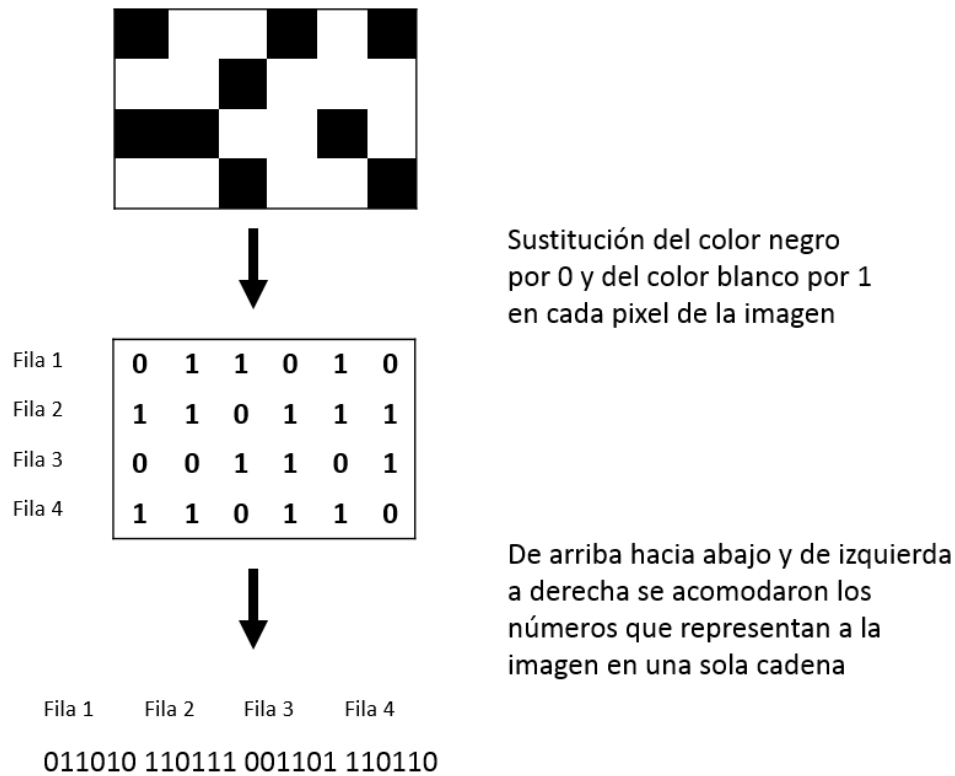


Figura 34 Ejemplo de conversión de la imagen de un código de iris a un patrón numérico.

Una vez realizado el preprocesamiento de todas las imágenes del banco de datos de iris para la generación de los códigos de iris y su correspondiente conversión para la obtención de patrones. El siguiente paso fue realizar la fase de entrenamiento del clasificador para obtener los vectores soporte  $S$  y  $\hat{S}$  definidos en 3.5.4.

Un ejemplo de patrón de soporte se muestra continuación:

Sean los siguientes patrones pertenecientes al conjunto fundamental:

X1: {0, 1, 0, 1, 0, 1, 1, 1, 1}

X2: {0, 1, 0, 1, 1, 1, 1, 1, 1}

X3: {0, 1, 1, 1, 1, 1, 0, 1, 1}

Y1: {0, 1, 1, 1, 0, 1, 1, 1, 1}

Y2: {0, 1, 0, 1, 1, 1, 0, 1, 1}

Los siguientes patrones son la versión negada de los patrones del conjunto fundamental:

$\widehat{X1}$ : {1, 0, 1, 0, 1, 0, 0, 0, 0}

$\widehat{X2}$ : {1, 0, 1, 0, 0, 0, 0, 0, 0}

$\widehat{X3}$ : {1, 0, 0, 0, 0, 0, 1, 0, 0}

$\widehat{Y1}$ : {1, 0, 0, 0, 1, 0, 0, 0, 0}

$\widehat{Y2}$ : {1, 0, 1, 0, 0, 0, 1, 0, 0}

Con lo anterior se pueden calcular los vectores de soporte  $S$  y  $\hat{S}$  que se obtienen mediante la fase de aprendizaje del clasificador Alfa-Beta SVM explicada en 3.5.4, los cuales son:

$$S: \{0, 1, 0, 1, 0, 1, 0, 1, 1\}$$

$$\hat{S}: \{1, 0, 0, 0, 0, 0, 0, 0, 0\}$$

Posteriormente, se generan los vectores restringidos y vectores restringidos negados de los vectores de conjunto fundamental y del conjunto fundamental negado respectivamente según el proceso de la fase de aprendizaje, quedando de la siguiente manera:

$$X1: \{0, 0, 0, 1\}$$

$$X2: \{0, 0, 1, 1\}$$

$$X3: \{0, 1, 1, 0\}$$

$$Y1: \{0, 1, 0, 1\}$$

$$Y2: \{0, 0, 1, 0\}$$

$$\hat{X}1: \{0, 1, 0, 1, 0, 0, 0, 0\}$$

$$\hat{X}2: \{0, 1, 0, 0, 0, 0, 0, 0\}$$

$$\hat{X}3: \{0, 0, 0, 0, 0, 1, 0, 0\}$$

$$\hat{Y}1: \{0, 0, 0, 1, 0, 0, 0, 0\}$$

$$\hat{Y}2: \{0, 1, 0, 0, 0, 1, 0, 0\}$$

Con lo anterior se tiene todo lo necesario para poder realizar la fase de recuperación. Para reconocer un patrón fuera del conjunto fundamental, se genera el vector restringido y el vector restringido negado del mismo y posteriormente se busca el vector con mayor similitud del conjunto fundamental mediante el cálculo de phi ( $\varphi$ ) y omega ( $\psi$ ). Por ejemplo:

Sea el patrón desconocido D:

$$D: \{0, 1, 1, 1, 1, 1, 0, 1, 0\}$$

Los vectores restringidos de para D son:

$$D: \{0, 1, 1, 0\}$$

$$\hat{D}: \{0, 0, 0, 0, 0, 1, 0, 1\}$$

Posteriormente se obtienen los valores de phi ( $\varphi$ ) y omega ( $\psi$ ) que son:

$$\varphi = 1$$

$$\psi = 0$$

Donde phi ( $\varphi$ ) y omega ( $\psi$ ) representan a la clase de un patrón dentro del conjunto de entrenamiento, y cuyos valores son comparados entre sí, lo cual sirve para determinar qué resultado devolver, siendo el mínimo de estos.

En este caso como  $\psi$  es menor o igual que  $\varphi$ , la clase devuelta es X representada por  $\psi$ .

#### 4.2.2 Huella Dactilar

Para el caso de las huellas dactilares, se utilizó el clasificador Gamma mediante la siguiente metodología [63, 64].

Dos ejemplos de patrones utilizados son los siguientes:

P: {11 250 -0.7853981633974483 T, 11 21 0 B}

Q: {110 19 1.5707963267948966 T, 10 141 0.7853981633974483 B}

Donde se observa que cada patrón tiene dos componentes que representan a las minucias de la huella dactilar, y donde cada componente está dividida en 4 secciones. La primera representa su posición en la imagen en el eje de las  $x$ , la segunda sección representa su posición en eje de las  $y$ , la siguiente al ángulo que tiene la minucia y por último una letra que representa al tipo de minucia (T para terminación y B para bifurcación).

Previo a la implementación de los patrones en el clasificador Gamma, es necesario realizar una alineación del patrón de entrada con todos los patrones del conjunto de entrenamiento. Dicha alineación se realiza con respecto al ángulo y a la posición. La primera para alinear las minucias del patrón de entrada con cada uno de los patrones con los que se va a comparar en el clasificador, tratando de alinear los patrones en el mismo ángulo (alineación de un patrón respecto al otro) y la segunda consiste en alinear el patrón de entrada con los que se va a comparar con respecto a su posición en el plano para hacer que las minucias estén alineadas en el plano. Una vez alineados los patrones, solo se consideraron las 12 minucias más cercanas al centro de la huella.

Previo a aplicar el algoritmo de clasificación, es necesario realizar una conversión a los datos de los patrones el cual consistió en:

1. Eliminar el dato del ángulo, dado que su función principal era la de alinear el patrón de entrada con los del conjunto de entrenamiento.
2. En el caso del tipo de minucia, a la T se le asignó un valor 0 y a la B un valor 1.

Tomando los ejemplos anteriores, después de realizar la conversión mencionada anteriormente quedarían de la siguiente manera:

P: {11 250 0, 11 21 1}

Q: {110 19 0, 10 141 1}

Posteriormente, se realizó una normalización tomando en cuenta la longitud máxima de los datos correspondientes al eje de las  $x$  y las  $y$ , añadiendo ceros a la izquierda para completar la máxima longitud encontrada. Al aplicar la normalización, los patrones resultantes tomando en cuenta los patrones de ejemplo son:

P: {011 250 0, 11 021 1}

Q: {110 019 0, 10 141 1}

Posteriormente se realizó una concatenación de en cada componente de los patrones para poder aplicar el clasificador gamma, quedando los patrones a utilizar como se muestran a continuación:

P: {0112500, 110211}

Q: {1100190, 101411}

Para aplicar el clasificador gamma, se realiza una fase de entrenamiento donde los patrones son codificados con el código Johnson-Möbius modificado. En este caso no es necesario realizar la etapa de la normalización debido a que previamente fue realizada de otra forma para obtener solo valores enteros positivos.

De acuerdo a los pasos mencionados en la sección 3.6.4 se procedió a realizar la fase del reconocimiento para determinar a qué clase pertenece un patrón de entrada.

### 4.3 Aplicación Web

Para que el desarrollo del sistema mantenga una comunicación entre el servidor y los clientes, se hace uso del patrón MVC (Model-View-Controller) sobre la plataforma J2EE de Java, lo cual permite distribuir de manera eficiente la codificación del sistema e implementar de manera fácil tecnologías que toman como base este estándar en el desarrollo de aplicaciones web [82, 97].

Además de las ventajas ya mencionadas del patrón MVC sobre aplicaciones web, es muy útil su aplicación sobre una arquitectura de cómputo en la nube utilizando una IaaS (Infraestructure as a Service). Entre las ventajas de implementar una aplicación web sobre una arquitectura basada en cómputo en la nube, se encuentran:

- Alta disponibilidad: Dado que el cómputo en la nube hace uso de la virtualización para la generación de equipos, el sistema no se encuentra en lugar físico dentro de las zonas geográficas marcadas por el proveedor de servicios, por otra parte, se puede realizar un balanceo de carga que permita el buen funcionamiento del sistema y su disponibilidad aún en caso de fallos.
- Consumo bajo demanda: Solo es necesario pagar por lo que se utiliza, por lo que el sistema puede estar compartido con otros sistemas, permitiendo así mejorar la eficiencia en el consumo de los recursos.
- Acceso inmediato: Es muy útil en proyectos donde no se cuenta con un *datacenter* y se requiere el acceso inmediato a los recursos de cómputo en cualquier momento.
- Escalable: Es sistema puede ser escalado vertical y horizontalmente de manera transparente y sin afectar el funcionamiento del sistema, como el aumento virtual de núcleos del procesador y la memoria RAM.

En el caso de los sistemas de reconocimiento de patrones, el que el sistema sea tolerante a fallos es de suma importancia, en este caso cuando el reconocimiento de una persona es crucial. La alta disponibilidad también es vital para para este trabajo de tesis y dada la naturaleza del sistema que es la de identificar y verificar la identidad de personas, existe un alto consumo de recursos, por lo que

se requiere que la aplicación pueda ser escalada en cualquier momento sin la necesidad de adecuar los algoritmos implementados en el sistema.

A continuación, se describe cada una de las capas del modelo MVC implementadas sobre una arquitectura de cómputo en la nube de Amazon AWS.

#### 4.3.1 Modelo

Esta capa representa las instancias que están almacenadas en bases de datos, las cuales son accedidas de forma eficiente para el sistema de reconocimiento de personas implementado.

En el sistema propuesto, con base en la arquitectura de *cloud computing* de Amazon AWS y específicamente de su servicio EC2, se realizó la instalación de dos motores de bases de datos dentro de una misma microinstancia que son: MySQL y MongoDB.

MySQL es un gestor de bases de datos relacional de Oracle, bajo una licencia GNU/GPL. Entre las ventajas de MySQL se encuentran:

- Es rápido y eficiente cuando la base de datos está diseñada correctamente de acuerdo a los estándares de generación de bases de datos relacionales.
- En aplicaciones web hay baja concurrencia con la modificación de datos y por el contrario es intensiva con la lectura de los datos.

La función principal de este gestor en el sistema es la de almacenar la información de las personas que no es útil para los algoritmos de clasificación pero sí para conocer más detalle de ellas al momento de obtener los resultados de una identificación o verificación, como son: nombre, apellidos, edad, dirección, entre otros.

Por otro lado, MongoDB es un gestor de bases de datos NoSQL o no relacional, bajo una licencia open-source y escrito en lenguaje C++. Las principales ventajas de MongoDB son:

- Almacenamiento orientado a documentos, muy útil cuando se utilizan esquemas dinámicos del estilo de JSON o XML.
- Se puede escalar horizontalmente y de forma transparente, aún con el gestor de base de datos corriendo.
- Ejecución de consultas utilizando el lenguaje JavaScript para que estas sean enviadas directamente a la base de datos para su posterior ejecución.

Dado que cada banco de datos puede tener diferentes tipos de patrones, diferentes rasgos y posiblemente patrones con información distinta; un DBMS (Database Management System) NoSQL es el caso útil para utilizarlo, debido a que no se realizan operaciones constantes de eliminación y donde los algoritmos de clasificación sean rápidos y por lo tanto requieren de una baja latencia por parte del DBMS.

El acceso al modelo se realiza a través de una capa de servicio que es implementada para conectarse a ambas bases de datos antes mencionadas desde el lenguaje Java. El patrón de diseño DAO (Data

Access Object) permite la abstracción de consultas de lectura y escritura hacia las bases de datos, para que se realicen de manera transparente al controlador.

#### **4.3.2 Vista**

Básicamente la vista representa la interfaz gráfica de la aplicación, que cuenta con los componentes necesarios para que el usuario interactúe con el sistema de una forma amigable. El uso del framework PrimeFaces de JBoss está diseñado para el desarrollo de interfaces web, el cual cuenta con una gran cantidad de componentes que facilitan el diseño de la capa de la vista e implementa de manera nativa y transparente las tecnologías de AJAX y JQuery.

PrimeFaces, permite que el sistema de reconocimiento de personas realice una sincronización entre el proceso de reconocimiento de personas y el renderizado del código HTML de los clientes referente al resultado obtenido de dicho proceso de reconocimiento. Los componentes utilizados para capturar una nueva huella biométrica, la de los resultados obtenidos e información de la persona reconocida se enlazan directamente con el controlador facilitando las tareas de escritura y lectura en el servicio de Amazon S3.

#### **4.3.3 Controlador**

Esta capa es la que aloja la mayor parte del negocio debido a que es la encargada de enlazar las dos capas anteriormente mencionadas. JSF es un framework basado completamente en el modelo MVC, que nos permite extraer las acciones realizadas por el usuario en la capa de la Vista para poder ejecutar los métodos que requieren acceso a los datos que se encuentran en el Modelo y a los algoritmos alojados en la capa de Amazon EC2.

En el caso del sistema de reconocimiento de personas, en esta capa residen los algoritmos de procesamiento digital de imágenes y reconocimiento de patrones. Sin embargo, estos algoritmos no interactúan directamente con la vista, sino que se hace uso de un componente propio de JSF conocido como *ManagedBean*. La función de esta clase es capturar los eventos de los componentes implementados en la vista, específicamente cuando el usuario desea reconocer una nueva huella biométrica para proceder con el reconocimiento de la misma, y proporcionar un método que permita la ejecución de los algoritmos para dicha tarea de reconocimiento.

En adición, la clase *ManagedBean* de JSF permite mantener, en caso de ser necesario, los datos relacionados con las operaciones que se realicen en memoria durante la sesión HTTP, para poder configurar el sistema y ejecutar el proceso de reconocimiento de manera eficiente.

### **4.4 Aplicación Móvil**

Con el fin de que el sistema pueda ser accedido no solo desde el internet a través de una aplicación web, se realizó el desarrollo de dos aplicaciones móviles para los sistemas iOS y Android.

Se realizó una aplicación móvil nativa para cada dispositivo con el fin de aprovechar de manera eficaz y eficiente los recursos de los dispositivos móviles. Lo anterior consistió en implementar el preprocesamiento de las imágenes y la extracción de las plantillas biométricas en el dispositivo móvil para agilizar el envío de los datos de la plantilla (archivo de texto) desde el dispositivo móvil

hacia el servidor que es quien tiene la base de datos y realiza la tarea del reconocimiento. Posteriormente, el servidor envía el resultado hacia el dispositivo móvil para mostrarlo en pantalla.

Cabe mencionar que el tiempo que tarda el preprocesamiento de las imágenes depende del dispositivo que se utilice, sin embargo, los algoritmos utilizados para el preprocesamiento fueron optimizados tomando en consideración lo anterior.

Al igual que la aplicación móvil es importante mencionar que se aplicó el patrón MVC para aplicaciones móviles, en donde la vista son las pantallas de la aplicación móvil que es con quien interactúa el usuario, el controlador al igual que las aplicaciones móviles es quien tiene la lógica del sistema y en este caso se hace uso del modelo solo para mantener segura la aplicación móvil mediante un login ya que la base de datos principal y que es la que se utiliza para el reconocimiento reside en el servidor web implementado en la nube.

#### **4.5 Arquitectura de las aplicaciones para el sistema multibiométrico**

En la Figura 35 se muestra la arquitectura general de la aplicación web y móvil que se implementó sobre Amazon AWS, haciendo uso de los servicios EC2 para la virtualización del Servidor Glassfish y MongoDB, y el servicio S3 de Amazon para el almacenamiento masivo de datos utilizados en el reconocimiento. Se puede observar la comunicación HTTP mediante la capa de la vista que hace uso de AJAX a través de PrimeFaces hacia la capa del Controlador basado en JSF y que se encuentra en el servidor, donde residen los algoritmos para efectuar la tarea del reconocimiento. De igual manera se muestra la interacción existente entre la capa del controlador con la información almacenada en las bases de datos perteneciente a la capa del modelo.



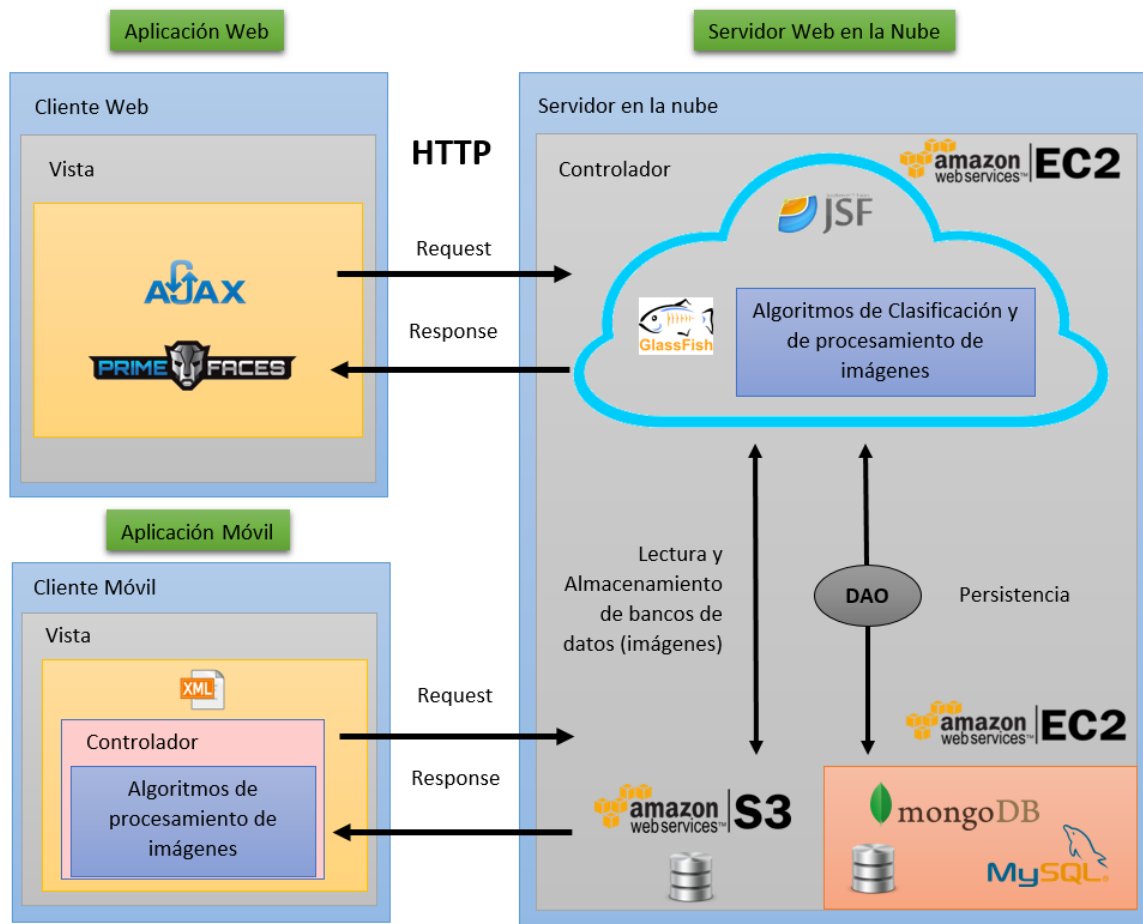


Figura 35 Arquitectura del servidor Web en la nube y las aplicaciones móviles para el sistema multibiométrico para el reconocimiento de personas

#### 4.6 Proceso completo

A continuación, se muestra el proceso completo del modelo propuesto para el sistema multibiométrico para el reconocimiento de personas (Iris y Huella Dactilar). En las Figuras 36-39 se muestran los diagramas que contienen el proceso completo.

En los primeros 3 diagramas se muestran subprocesos con funciones específicas utilizados y especificados en el diagrama 4 que incluye el proceso completo del reconocimiento de personas.

En el diagrama 1 de la Figura 36 se muestra el subproceso de registro de plantillas biométricas en base de datos para su posterior uso en la tarea del reconocimiento de un patrón.

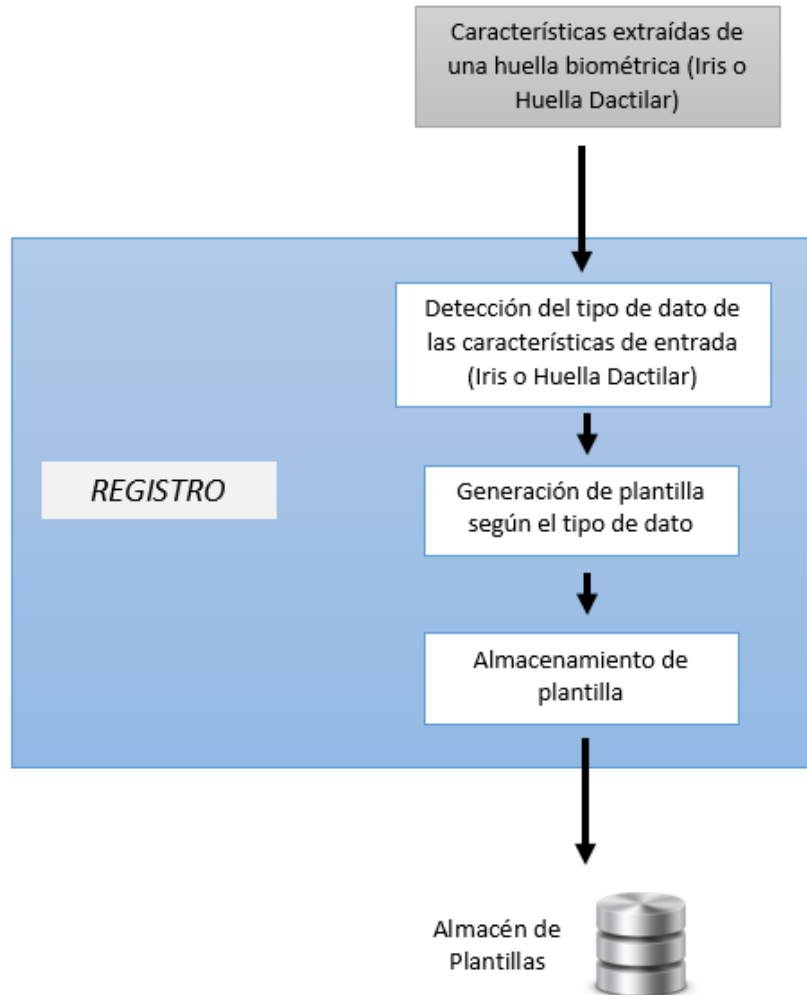


Figura 36 Diagrama 1 del subproceso de registro de plantillas biométricas

En el diagrama 2 de la Figura 37 se muestra el subproceso que permite el reconocimiento de las características extraídas a partir de una huella biométrica de Iris i Huella Dactilar y proporcionar un resultado de dicho reconocimiento.



Figura 37 Diagrama 2 del subproceso de reconocimiento de características extraídas de una huella biométrica (Iris o Huella Dactilar)

En el diagrama 3 de la Figura 38 se muestra el subproceso que realiza la toma de una decisión para proporcionar un resultado final en el sistema multibiométrico cuyas entradas de dicho subproceso son los resultados individuales del reconocimiento del Iris y la Huella Dactilar.

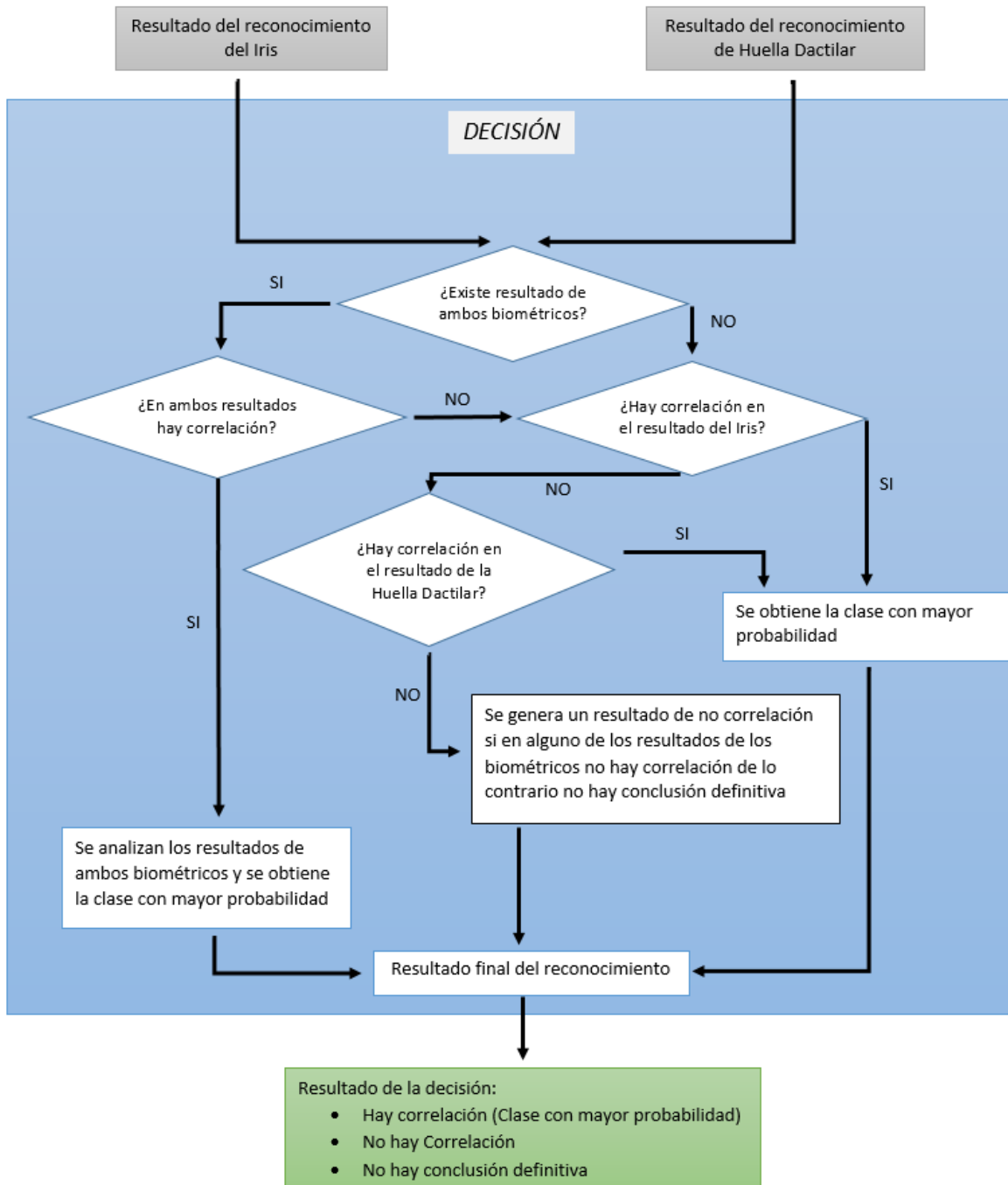


Figura 38 Diagrama 3 del subproceso de la toma de decisión para proporcionar un resultado al sistema multibiométrico

Finalmente, en el diagrama 4 de la Figura 39 se muestra el proceso completo del modelo propuesto para el sistema multibiométrico de reconocimiento de personas en el cual se indica el uso de los subprocesos mostrados en los diagramas 1, 2 y 3.

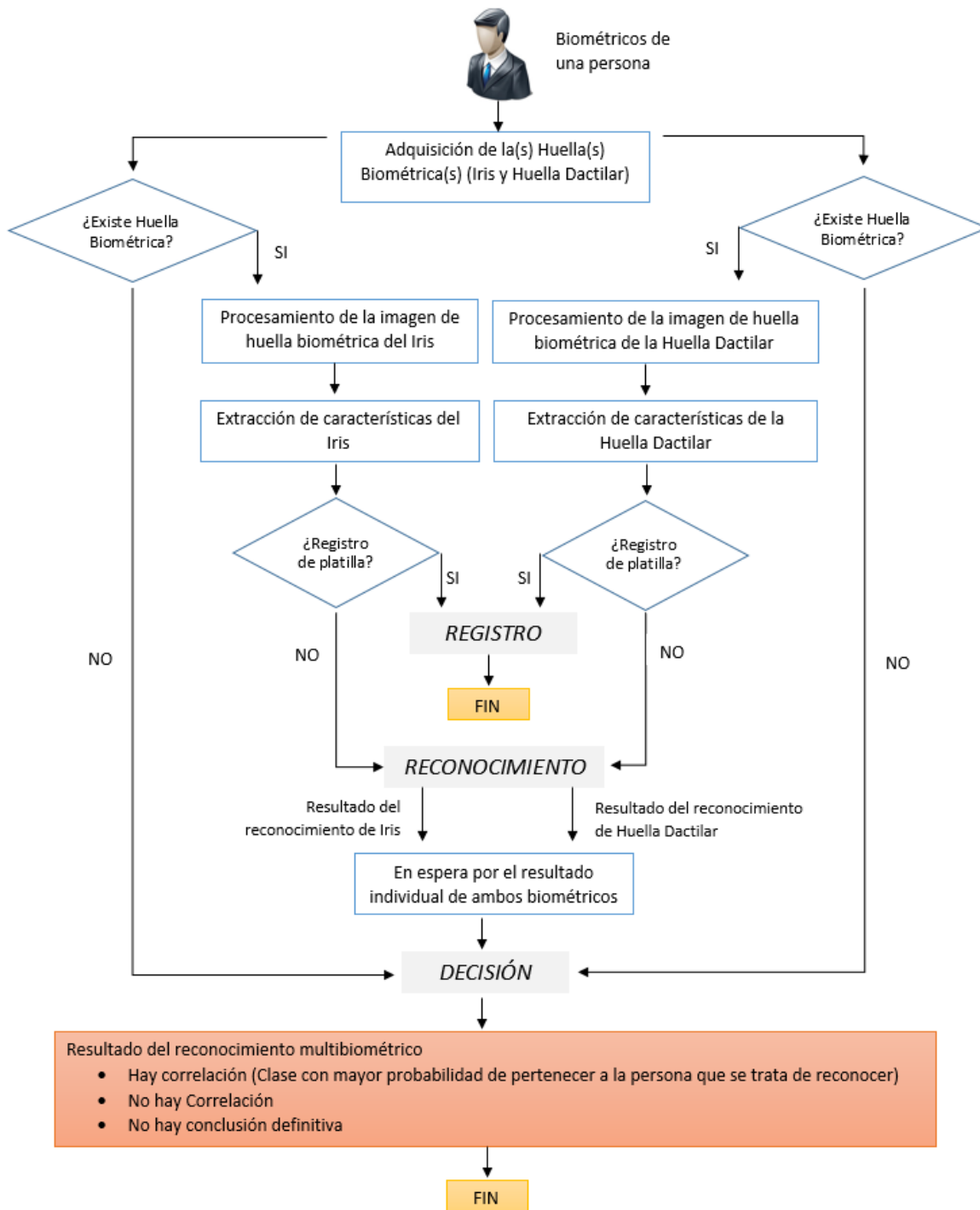


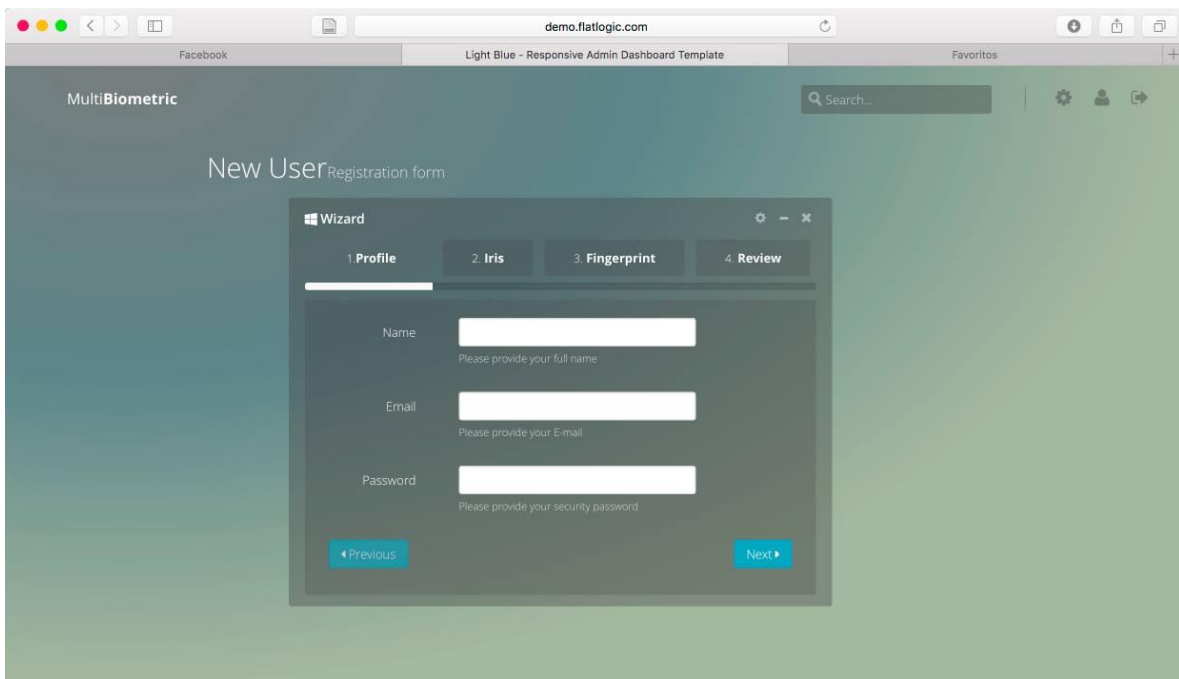
Figura 39 Diagrama 4 del proceso completo del modelo propuesto para el sistema multibiométrico para el reconocimiento de personas

## 4.7 Sistema multibiométrico

En esta sección se mostrarán unas pantallas que muestran el funcionamiento de la aplicación web y de la aplicación desarrollada para los dispositivos Android y iOS. Las aplicaciones desarrolladas hacen uso de la arquitectura y el proceso mencionado anteriormente.

En las Figuras 40, 41, 42 y 43 se muestran pantallas del sistema web donde se observa el registro de los usuarios en el sistema para que la información almacenada sea posteriormente utilizada.

En la Figura 40 se muestra se hace un registro de información muy básica de la persona que solo consistió del nombre, correo y una contraseña.

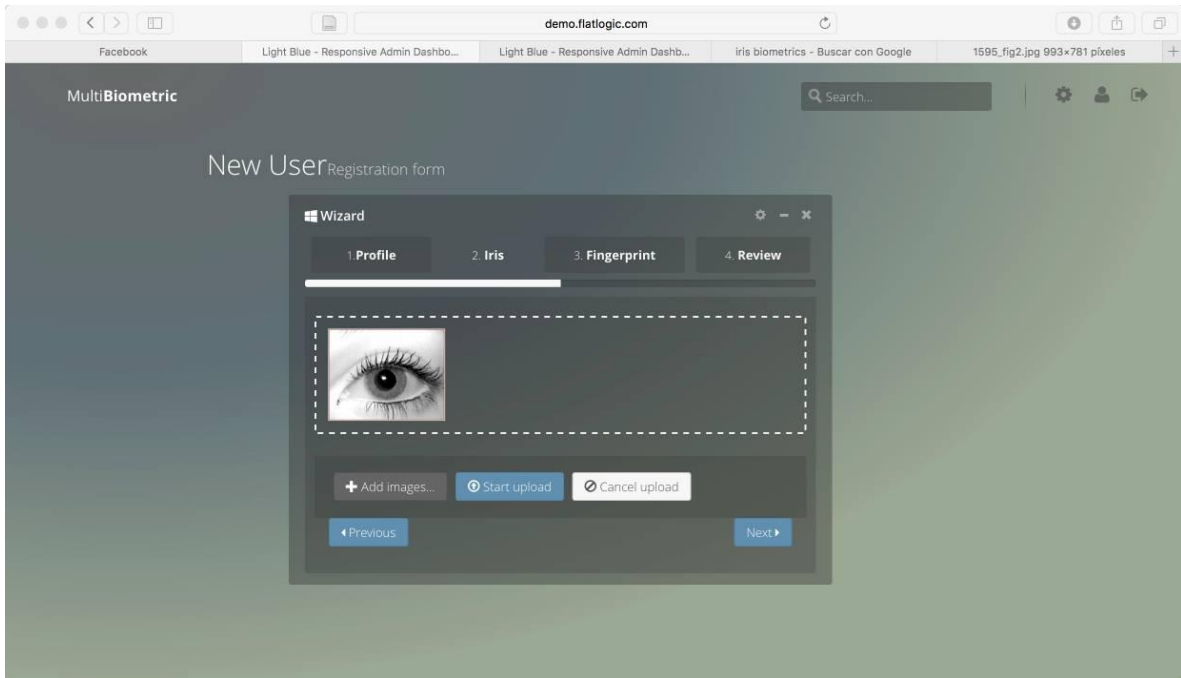


*Figura 40 Pantalla de registro de información básica de un usuario*

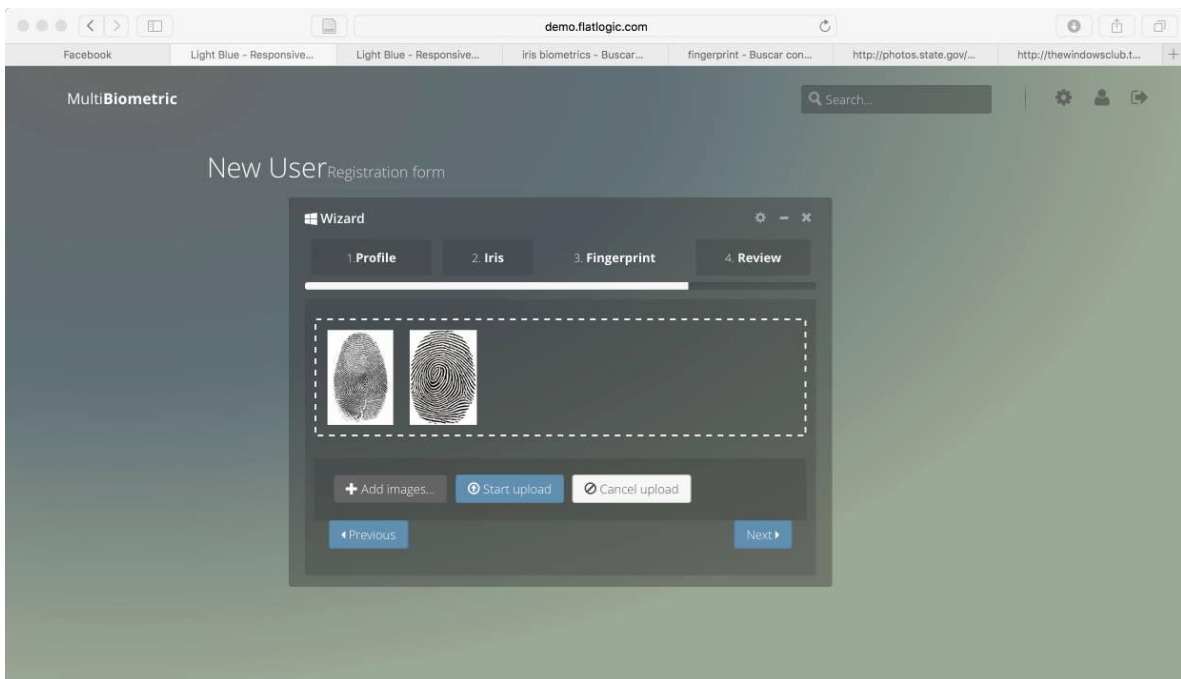
En las Figuras 41 y 42 se muestra el registro de las imágenes de iris y huella dactilar pertenecientes a la persona que se está registrando que posteriormente serán utilizadas para poder identificarla o verificada según sea el caso.

Y por último en la Figura 43 se muestra una pantalla de que el registro fue satisfactorio y que por lo tanto la información capturada puede empezar a utilizarse.

Cabe mencionar que al momento de realizar el registro, las imágenes de huella dactilar e iris son procesadas y se obtienen los patrones en cada caso, con la finalidad de que desde que ya se tenga lo necesario para una verificación o identificación.



*Figura 41 Pantalla en la que se capturan las imágenes de iris del usuario que se está registrando*



*Figura 42 Pantalla en la que se capturan las imágenes de huella dactilar del usuario que se está registrando*

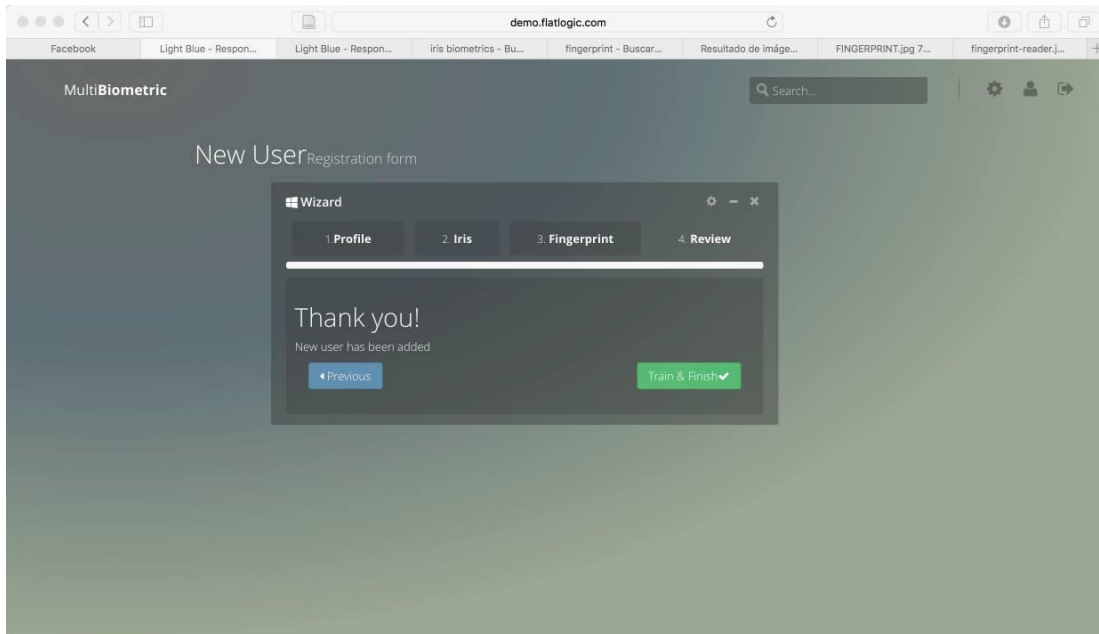


Figura 43 Pantalla que muestra un registro exitoso

A continuación, en las Figuras 44, 45 y 46 se muestran las pantallas de las aplicaciones móviles en las cuales se observa el proceso de verificación de usuarios que están registrados en el sistema.

En la Figura 45 se muestra una pantalla de inicio de sesión para posteriormente acceder a un menú de opciones donde puede realizar una verificación de sí mismo mediante la huella dactilar o el iris.

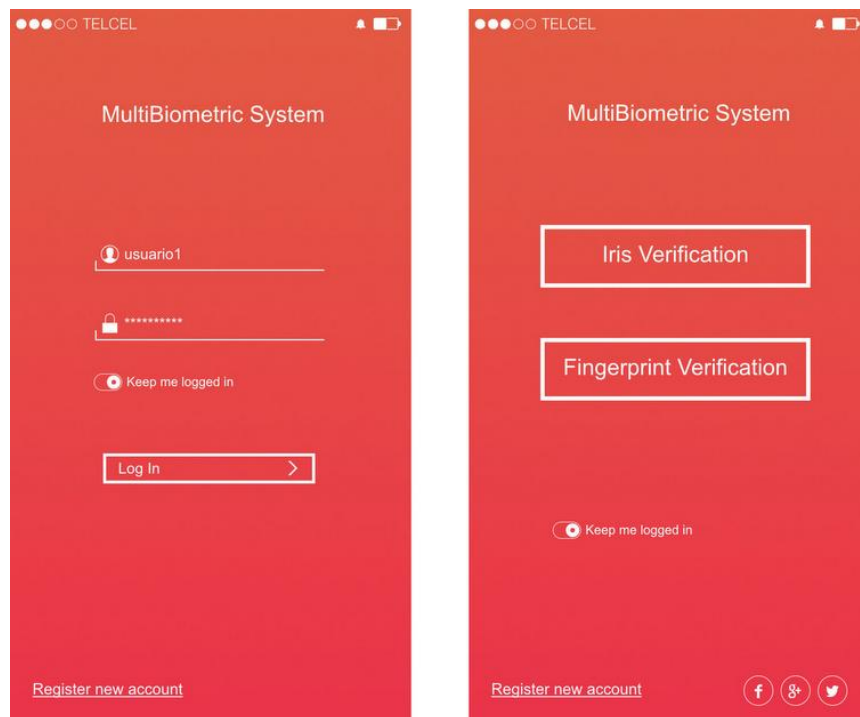


Figura 44 Pantalla de inicio de sesión y menú de opciones para la verificación



En las Figuras 45 y 46 se muestran los procesos de verificación mediante iris o huella dactilar de un usuario, donde se carga la imagen del iris o huella dactilar que se va a verificar, las cuales serán procesadas para obtener los patrones y posteriormente compararse con las que se tienen registradas en el sistema y finalizar con un porcentaje que representa que tanto se parece la imagen de huella dactilar o iris proporcionados contra las registradas previamente en el sistema.

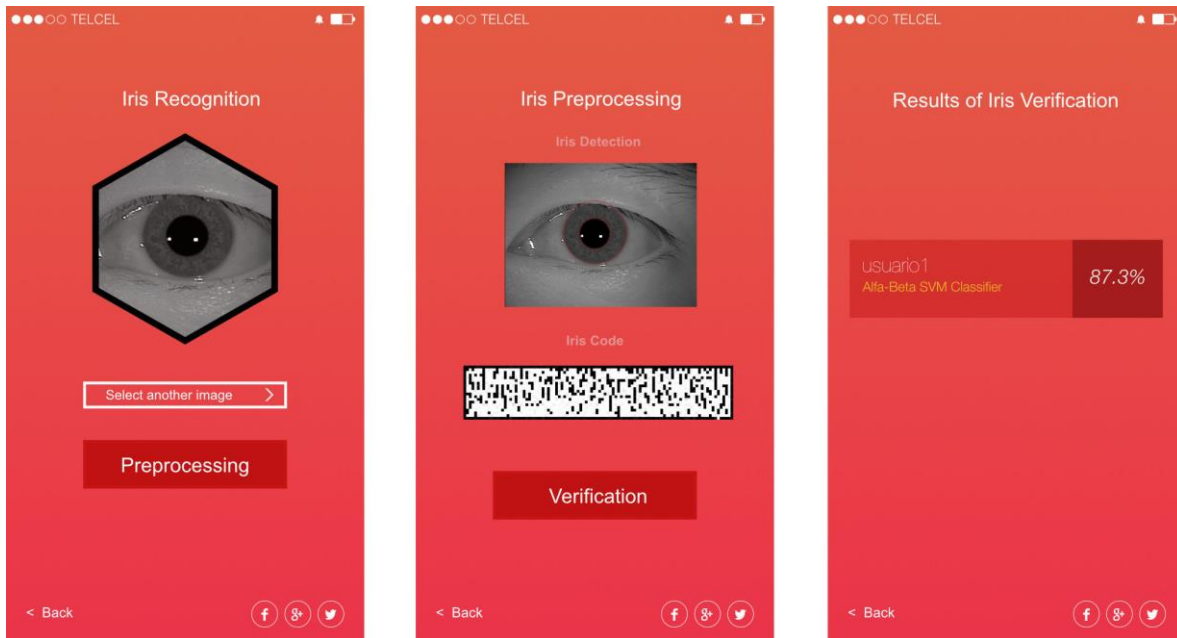


Figura 45 Pantalla que muestra el proceso de verificación por medio del iris en un dispositivo móvil

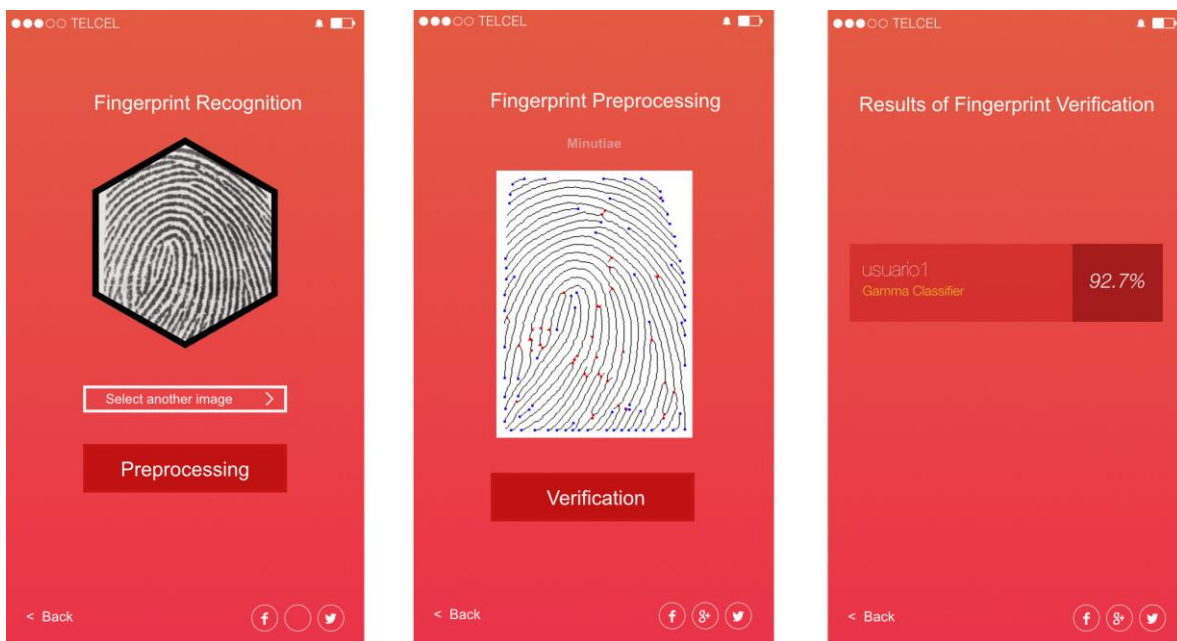


Figura 46 Pantalla que muestra el proceso de verificación por medio de la huella dactilar en un dispositivo móvil

## Capítulo 5

### 5. Resultados y Discusión

En este capítulo se describen los experimentos realizados y se analizan los resultados obtenidos. Para ello, primero se presentan los bancos de datos utilizados durante la fase experimental, así como la plataforma tecnológica empleada para desarrollar el entorno de pruebas.

#### 5.1 Bancos de Datos

El contenido de esta sección basado en los bancos de prueba utilizados para tareas de procesamiento digital de imágenes y clasificación. Los bancos de datos utilizados son las de CASIA para huellas dactilares e iris, específicamente hablando las versiones utilizadas fueron CASIA-Iris-Syn Versión 4 para el reconocimiento de iris y CASIA-Fingerprint Versión 5 para huellas dactilares. La primera consta de 10,000 imágenes sintéticas pertenecientes a 1,000 clases diferentes, a partir de dicho banco de datos se formó uno nuevo que está conformado por un subconjunto de imágenes de CASIA-Iris-Syn conformado por 2000 imágenes pertenecientes a 250 clases diferentes (CASIA-Iris-Syn-Sub). La segunda base de datos para huellas dactilares contiene 20,000 pertenecientes a 500 individuos donde se tienen 40 muestras por individuo pertenecientes a 8 de 10 dedos de las manos. Para fines prácticos se consideró a cada conjunto del mismo dedo como una clase teniendo así un total de 4,000 clases diferentes en el banco de datos, y al igual que en el banco de datos de la iris aquí también se creó un nuevo banco de datos conformado por 1250 imágenes pertenecientes a 250 clases diferentes (CASIA-Fingerprint-Sub).

Finalmente para las pruebas finales del sistema multibiométrico, se realizó la creación de un nuevo banco de datos a partir de los bancos creados (CASIA-Iris-Syn-Sub y CASIA-Fingerprint-Sub). Creando parejas entre las clases de ambos bancos de datos, a dicho banco se le denominó CASIA-Iris-Fingerprint.

#### 5.2 Entorno de Prueba

El entorno donde se implementó el sistema multibiométrico presentado en esta tesis, es una micro-instancia del servicio de Amazon EC2 con las siguientes características:

- Tipo: t1.micro
- Sistema Operativo: Ubuntu Amazon Custom
- Almacenamiento: 10GB
- vCPU: 1
- Memoria RAM: 2GB

Los algoritmos implementados en WEKA se ejecutaron con las configuraciones por defecto y en una computadora de escritorio con las siguientes características:

- Modelo: Genérico
- Sistema Operativo: Windows 7 x64
- Software: Weka 3.6
- Procesador: Intel i7-4770K CPU @ 3.50GHz

- Memoria RAM: 16GB
- Almacenamiento: 120GB SSD

Asimismo, se utilizó el método de validación K-Fold Cross-Validation con  $k=10$  estratificado, debido a que este valor es utilizado en gran parte de la literatura y artículos publicados en revistas JCR como un estándar en la medida de rendimiento para sistemas de clasificación.

### 5.3 Resultados experimentales

Durante la fase experimental, se probaron tres configuraciones diferentes de la propuesta presentada en esta tesis. Primero, se evaluó el desempeño de los métodos para la identificación biométrica de iris y huella dactilar por separado. Posteriormente, se evaluó el desempeño del sistema completo, involucrando ambos métodos biométricos en un sistema multibiométrico integrado.

En el ámbito del estudio de biométricos (incluyendo el caso de los sistemas multibiométricos) se consideran dos grandes escenarios como los más frecuentes [4]. A saber, estos escenarios son la identificación y la verificación. Como se había mencionado en el capítulo 1, la identificación corresponde a la situación en la que no se conoce a la persona y lo que el sistema indicará es precisamente de qué persona se trata. Por otro lado, la verificación consiste en confirmar la identidad de una persona, al comparar su identificador propuesto con las características biométricas asociadas. Dicho de otra manera, es posible caracterizar las entradas y salidas de un sistema biométrico en cada uno de estos escenarios según la Tabla 4

*Tabla 4 Escenarios de operación de un sistema biométrico.*

Escenario	Entradas	Salida
Identificación	Huella biométrica	Identidad, o clase desconocida
Verificación	Identidad propuesta, huella biométrica	Sí concuerda con la identidad propuesta, o No concuerda

En este sentido, si el sistema biométrico tiene conocimiento de  $p$  personas, en el caso de la identificación dará como respuesta una de esas  $p$  identidades, o bien la clase desconocida ( $p+1$  posibles respuestas) Por el contrario, en la verificación sólo hay dos posibles respuestas: se acepta la identidad propuesta (resultado positivo) o se rechaza la identidad propuesta (resultado negativo).

Dado lo anterior, se decidió probar el sistema bajo el escenario de verificación, considerando que esto facilitará evaluar y comparar los resultados obtenidos. En este escenario, cabe señalar que los posibles resultados quedan circunscritos a una matriz de confusión de dimensión 2 por 2, en donde existen 4 casos posibles:

- Verdadero positivo (VP): El clasificador asigna la clase correcta al patrón, por lo que la persona es aceptada y verificada correctamente.
- Falso positivo (FP): El clasificador asigna una clase incorrecta al patrón, por lo que la persona es aceptada con una identidad incorrecta, por lo que es aceptada y verificada incorrectamente.

- Verdadero negativo (VN): El clasificador detecta que el patrón de una persona desconocida no pertenece a ninguna clase conocida y asigna la clase desconocida, por lo que la persona es rechazada correctamente.
- Falso negativo (FN): El clasificador detecta que el patrón de una persona conocida no pertenece a ninguna clase conocida y asigna la clase desconocida, por lo que la persona es rechazada incorrectamente.

Para evaluar el desempeño del sistema propuesto al trabajar únicamente con información de iris, se utilizó el banco de datos descrito anteriormente e identificado como CASIA-Iris-Syn-Sub, mismo que está compuesto por 8 muestras de 250 individuos, dando un total de 2000 imágenes de iris.

Como se explicó en el capítulo 4, el método propuesto utiliza las Alfa-Bet SVMs como clasificador para tratar las imágenes de iris. En particular, se le añadió una etapa de preprocesamiento al clasificador para darle la capacidad de identificar patrones correspondientes a la misma imagen, cuando esta ha sido rotada. Para ello, se le añadió la capacidad de realizar un corrimiento a izquierda o a la derecha, con rotación, al patrón binario con el que se opera la Alfa-Beta SVM.

Por ejemplo, supóngase que los patrones de entrada que representa a un par de iris son:

1010101                      1001010

Entonces la rotación simplemente consiste en eliminar las  $n$  componentes de cada fila del patrón de entrada, ya sea del final o del inicio y colocarlas en el lado opuesto en su fila respectiva. En el siguiente caso se realizó una rotación al patrón eliminando las primeras 3 componentes del inicio y colocándolas después al final:

0101101                      1010100

Asimismo, se le dotó a las Alfa-Beta SVM de la capacidad de asignar una clase desconocida, cuando el valor máximo de similitud calculado (internamente) por el clasificador no supera un umbral establecido.

Dado que el método de validación utilizado ( $K$ -Fold Cross Validation con  $K=10$ ) asigna los patrones a los diferentes folds de manera aleatoria, el experimento se ejecutó 10 veces. Los resultados obtenidos se muestran en la Tabla 5, donde VP, FP, VN y FN se refieren a verdadero positivo, falso positivo, verdadero negativo y falso negativo, respectivamente.

Como se puede apreciar, el promedio de falsos positivos es de 8.64%, mientras que el promedio de falsos negativos es de 2.32%. Esto nos da un promedio de resultados correctos (clasificación correcta, VP + VN) de 89.04%. Este resultado es bueno, aunque no sobresaliente.

Con el fin de evaluar el desempeño del sistema propuesto cuando sólo utiliza datos referentes a la huella dactilar, se tomaron 5 muestras de 250 individuos del banco de datos CASIA-Fingerprint-Sub, para un total de 1250 imágenes. Aunque este banco de datos contiene 50 muestras por individuo, en estos experimentos sólo se utilizaron las imágenes correspondientes al dedo índice.

Como se mencionó anteriormente, el método propuesto hace uso del clasificador Gamma para tratar los patrones generados a partir de imágenes de huellas dactilares. En este caso, se aprovechó la

capacidad del clasificador Gamma para lidiar con una clase desconocida cuando la persona a ser verificada (o identificada si se quisiera) no ha sido registrada en el sistema.

*Tabla 5 Resultados experimentales de 10 experimentos con datos de iris, con posibles clases desconocidas (verdaderos negativos)*

<b>Experimento</b>	<b>VP</b>	<b>FP</b>	<b>VN</b>	<b>FN</b>
1	167	23	55	5
2	155	21	68	6
3	156	21	66	7
4	164	22	58	6
5	166	23	56	5
6	158	22	63	7
7	169	22	55	4
8	152	19	72	7
9	168	21	56	5
10	158	22	64	6
<b>Promedio</b>	161.3	21.6	61.3	5.8

De manera similar a lo sucedido con la iris, el experimento se ejecutó 10 veces para tomar en cuenta los efectos aleatorios del K-Fold Cross Validation. Los resultados obtenidos se muestran en la Tabla 6, donde VP, FP, VN y FN se refieren a verdadero positivo, falso positivo, verdadero negativo y falso negativo, respectivamente.

*Tabla 6 Resultados experimentales de 10 experimentos con datos de huellas dactilares, con posibles clases desconocidas (verdaderos negativos)*

<b>Experimento</b>	<b>VP</b>	<b>FP</b>	<b>VN</b>	<b>FN</b>
1	185	15	47	3
2	174	13	59	4
3	173	12	63	2
4	178	14	54	4
5	182	13	50	5
6	171	16	60	3
7	173	15	60	2
8	178	13	57	2
9	184	13	49	4
10	186	13	46	5
<b>Promedio</b>	178.4	13.7	54.5	3.4

En este caso, los resultados son ligeramente mejores que con iris. Ahora, el promedio de resultados correctos (clasificación correcta, VP + VN) es de 93.16%, mientras que los promedios de falsos negativos y de falsos positivos son de 1.36% y 5.48%, respectivamente.

Ahora bien, ¿qué sucede cuando se utiliza el sistema propuesto ya integrado? Con el fin de aprovechar mejor las bondades de ambos reconocedores inmersos en la propuesta (Alfa-Beta SVMs y clasificador Gamma), así como tratar el caso de respuestas no coincidentes, se modificó la respuesta entregada por ambos clasificadores para admitir como salida las tres clases que representan a las mejores opciones, en lugar de que dicha salida sea sólo una clase.

De esta manera, a cada clase entregada por uno de los dos clasificadores se le asignó un valor correspondiente al porcentaje obtenido mediante las validaciones aplicadas en cada clasificador que hacen uso de un umbral para determinar qué tan parecido es un patrón de entrada con los que se tienen registrados. Los porcentajes obtenidos se manejaron como números reales con 8 dígitos de precisión.

Para determinar el resultado entregado por el sistema multibiométrico, se consideraron dos casos:

- Si ambos clasificadores dan como mejor clase (i.e. mayor porcentaje) a la misma clase, esta será la respuesta. Este es el caso en que ambos clasificadores coinciden en su respuesta.
- En caso de ambigüedad (la mejor clase entregada por cada clasificador no coinciden entre sí), se consideran aquellas clases presentes en ambas respuestas, se suman sus porcentajes respectivos y se selecciona aquella con el mayor valor.

Por ejemplo, si cada clasificador diera los siguientes resultados:

Iris (Alfa-Beta SVM)	Huella dactilar (clasificador Gamma)
C6 0.89123685	C6 0.91458621
C33 0.77126587	C43 0.82214595
C32 0.72596482	C52 0.75215698

entonces la clase devuelta sería C6, puesto que ambos clasificadores coinciden en que es esta la mejor clase según sus porcentajes obtenidos.

Sin embargo, si los resultados individuales fueran como sigue:

Iris (Alfa-Beta SVM)	Huella dactilar (clasificador Gamma)
C65 0.85236842	C4 0.94465321
C4 0.82546879	C42 0.81468842
C42 0.71356871	C74 0.72468745

entonces la clase devuelta sería C4, puesto que habiendo ambigüedad hay dos clases presentadas por ambos clasificadores (C4 y C42), pero la suma de los porcentajes de C4 es de 1.770122 mientras que la de C42 es de 1.52825713, siendo seleccionada C4 por tener un mayor valor.

Para este experimento se consideraron dos casos. En el primero no se garantiza que el conjunto de entrenamiento tenga información de todas las clases en todas las corridas de K-Fold Cross Validation. Este caso simula la situación en la que algunos individuos que operan con el sistema son totalmente desconocidos, y se logra al usar la versión no estratificada de K-Fold Cross Validation, como ha sucedido en los experimentos anteriores. Los resultados obtenidos se muestran en la Tabla 7, donde VP, FP, VN y FN se refieren a verdadero positivo, falso positivo, verdadero negativo y falso negativo, respectivamente.

*Tabla 7 Resultados experimentales de 10 experimentos con datos de iris y huellas dactilares, con posibles clases desconocidas (verdaderos negativos)*

<b>Experimento</b>	<b>VP</b>	<b>FP</b>	<b>VN</b>	<b>FN</b>
1	196	14	36	4
2	188	16	41	5

3	192	14	41	3
4	187	11	48	4
5	189	15	44	2
6	196	12	39	3
7	194	16	39	1
8	190	11	44	5
9	191	13	43	3
10	194	11	44	1
<b>Promedio</b>	191.7	13.3	41.9	3.1

Ahora el promedio de falsos positivos es de 5.32%, mientras que el de falsos negativos es de 1.24% y el promedio de resultados correctos (clasificación correcta) es de 93.44%. Como se puede ver, se tiene una ligera mejora con respecto al primer experimento (con sólo datos de iris) y una mejora apenas perceptible con respecto al segundo experimento (con sólo datos de huellas dactilares).

El segundo caso de este experimento se garantiza que todos los folds generados por K-Fold Cross Validation cuenten con información de todas las clases, pero no se garantiza que siempre se cuente con información de ambos biométricos. Este caso simula la situación en donde el sistema cuenta con información de todos los individuos que hacen uso del mismo, pero no necesariamente cuenta con información completa (i.e. todos los biométricos) para todos los individuos. Los resultados obtenidos se muestran en la Tabla 8, donde VP, FP, VN y FN se refieren a verdadero positivo, falso positivo, verdadero negativo y falso negativo, respectivamente.

*Tabla 8 Resultados experimentales de 10 experimentos con datos de iris y huellas dactilares, con posibles clases desconocidas (verdaderos negativos)*

<b>Experimento</b>	<b>VP</b>	<b>FP</b>	<b>VN</b>	<b>FN</b>
1	190	60	-	-
2	191	59	-	-
3	196	54	-	-
4	198	52	-	-
5	189	61	-	-
6	191	59	-	-
7	192	58	-	-
8	191	59	-	-
9	198	52	-	-
10	192	58	-	-
<b>Promedio</b>	192.8	57.2	-	-

Nótese que en este caso, no existen resultados para los verdaderos negativos o los falsos negativos, puesto que se garantiza que en todas las instancias, el individuo a ser verificado (o identificado, si así se quisiera) es conocido por el sistema, al menos parcialmente. En este caso, el promedio de verdaderos positivos (resultados correctos) es de 77.12%, con un promedio de falsos positivos (error) de 22.88%.

Estos desempeños indican claramente que, para el buen funcionamiento del sistema propuesto, es preferible contar con información completa (i.e. todos los biométricos) de todas las personas conocidas y arriesgarse a encontrar personas desconocidas, que contar con información incompleta de todas las personas que usarán el sistema.

## Capítulo 6

### 6. Conclusiones y Trabajo a Futuro

En este capítulo se presentan las conclusiones a partir de las pruebas con los diferentes bancos de datos analizados en el capítulo anterior. Así mismo, se proponen ideas que sirvan como trabajo a futuro para ser considerados nuevos trabajos de investigación.

#### 6.1 Conclusiones

Las conclusiones a las que se llegó son las siguientes:

Se realizó una investigación del estado del arte sobre la clasificación actual para el reconocimiento de personas con el uso de iris y huella dactilar, concluyendo en que lo más utilizado es la distancia de Hamming para el reconocimiento de iris y la comparación de minucias con las huellas dactilares, y aunque las Máquinas Asociativas Alfa-Beta SVM y el Clasificador Gamma aunque no han sido utilizados en la etapa de clasificación, han demostrado tener buenos resultados.

Se desarrolló un sistema multibiométrico para el reconocimiento de personas basado en los métodos de procesamiento digital de imágenes más utilizados en el estado del arte con algunas modificaciones para la obtención de mejores resultados y el uso de las Máquinas Asociativas Alfa-Beta SVM, obteniendo resultados competitivos con base en la mediación de la tasa de falsos positivos y la tasa de verdaderos negativos.

Aunque el algoritmo de las Máquinas Asociativas Alfa-Beta SVM y el Clasificador Gamma no han sido utilizados en el reconocimiento de personas, se realizaron adecuaciones en ambos para utilizarlo en la clasificación de patrones de este tipo.

Se llegó a la conclusión de que la escalabilidad es una de las ventajas más significativas del uso de una plataforma de cómputo en la nube.

Se mostró un caso práctico y necesario para el uso de las bases de datos NoSQL.

Se implementó el sistema multibiométrico en la nube de Amazon sobre los servicios EC2 y S3; brindando una arquitectura para aprovechar las ventajas de la misma.

Se implementó una aplicación móvil que permite el procesamiento de imágenes de iris y huella dactilar para la extracción de características.

Los resultados obtenidos reflejan un sistema con alto grado de confiabilidad para pruebas de reconocimiento de personas.

El modelo propuesto para el reconocimiento de personas, permite la inclusión de más biométricos del tipo fisiológico para intentar mejorar los resultados en la identificación o verificación de personas.

Permitir que el sistema funcione aun cuando se tenga solo un biométrico para el reconocimiento.



## **6.2 Trabajo a Futuro**

Probar el algoritmo con bancos de datos que sean más grandes, que contengan al menos 1 millón de instancias.

Modificar ciertos procesos del modelo de reconocimiento de personas para ser implementado sobre cómputo paralelo haciendo uso de una herramienta como CUDA.

Utilizar otros algoritmos de clasificación para la fase de identificación o verificación del sistema propuesto.

Utilizar otros métodos de procesamiento digital de imágenes en la fase de mejora de las imágenes y extracción de características en el sistema propuesto.

Implementar el sistema sobre otras plataformas de cómputo en la nube.

Incluir en el sistema de reconocimiento nuevos biométricos fisiológicos o incluso conductuales que permitan obtener mejores resultados en el reconocimiento de las personas.

Probar el modelo de reconocimiento de personas con algunos bancos de datos proporcionados por instituciones que manejen alguno o ambos biométricos utilizados en este trabajo.

## Referencias

- [1] Shoniregun, Charles A., and Stephen Crosier. *Securing Biometrics Applications*. London; New York: Springer, 2008.
- [2] Wayman, James, ed. *Biometric Systems: Technology, Design, and Performance Evaluation*. London: Springer, 2005: 1-20.
- [3] Jain, Anil, Ruud Bolle, and Sharath Pankanti. *Personal Identification in Networked Society*. Kluwer Academic Publisher, 1999.
- [4] Ashbourn, Julian. *Biometrics in the New World*. Cham: Springer International Publishing, 2014: Pages 2-23.
- [5] Hong, L., Yifei Wan, and A. Jain. "Fingerprint Image Enhancement: Algorithm and Performance Evaluation." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (August 1998): 777–89.
- [6] Leung, W. F., S. H. Leung, W. H. Lau, and Andrew Luk. "Fingerprint Recognition Using Neural Network." In *Neural Networks for Signal Processing [1991]*, Proceedings of the 1991 IEEE Workshop, 226–35. IEEE, 1991.
- [7] Afsar, F. A., M. Arif, and M. Hussain. "Fingerprint Identification and Verification System Using Minutiae Matching." In *National Conference on Emerging Technologies*, 141–46, 2004.
- [8] Lee, Henry C., and R. E. Gaensslen. *Advances in Fingerprint Technology*. Elsevier, 1991.
- [9] Chatterjee, Atanu, Shuvankar Mandal, G. M. Atiqur Rahaman, and Abu Shamim Mohammad Arif. "Fingerprint Identification and Verification System by Minutiae Extraction Using Artificial Neural Network." *JCIT* 1 (2010): 12–16.
- [10] Jain, A., L. Hong, and R. Bolle. "On-Line Fingerprint Verification." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19, no. 4 (April 1997): 302–314.
- [11] Ratha, Nalini K., Shaoyun Chen, and Anil K. Jain. "Adaptive Flow Orientation-Based Feature Extraction in Fingerprint Images." *Pattern Recognition* 28, no. 11 (November 1995): 1657–72.
- [12] Tan, Xuejun, and Bir Bhanu. "Fingerprint Matching by Genetic Algorithms." *Pattern Recognition* 39, no. 3 (March 2006): 465–77.
- [13] Kaur, Manjjeet, Mukhwinder Singh, Akshay Girdhar, and Parvinder S. Sandhu. "Fingerprint Verification System Using Minutiae Extraction Technique." *World Academy of Science, Engineering and Technology* 46 (2008): 497–502.
- [14] Ezhilarasan, M., D. Suresh Kumar, S. Santhanakrishnan, S. Dhanabalan, and A. Vinod. "Person Identification Using Fingerprint by Hybridizing Core Point and Minutiae

- Features.” *International Journal on Computer Science and Engineering* 2, no. 9 (2010): 3075–78.
- [15] Bahgat, G. A., A. H. Khalil, N. S. Abdel Kader, and S. Mashali. “Fast and Accurate Algorithm for Core Point Detection in Fingerprint Images.” *Egyptian Informatics Journal* 14 (March 2013): 15–25.
- [16] Daugman, John G. “High Confidence Visual Recognition of Persons by a Test of Statistical Independence.” *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 15, no. 11 (1993): 1148–61.
- [17] Daugman, J. “How Iris Recognition Works.” *IEEE Transactions on Circuits and Systems for Video Technology* 14, no. 1 (January 2004): 21–30.
- [18] Daugman, John. “Probing the Uniqueness and Randomness of IrisCodes: Results from 200 Billion Iris Pair Comparisons.” *Proceedings of the IEEE* 94, no. 11 (2006): 1927–35.
- [19] Daugman, J. “New Methods in Iris Recognition.” *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 37, no. 5 (October 2007): 1167–75.
- [20] Kong, A.W.K., D. Zhang, and M.S. Kamel. “An Analysis of IrisCode.” *IEEE Transactions on Image Processing* 19, no. 2 (February 2010): 522–32.
- [21] Nabti, Makram, and Ahmed Bouridane. “An Effective and Fast Iris Recognition System Based on a Combined Multiscale Feature Extraction Technique.” *Pattern Recognition, Part Special issue: Feature Generation and Machine Learning for Robust Multimodal Biometrics*, 41, no. 3 (March 2008): 868–79.
- [22] Pillai, J.K., V.M. Patel, R. Chellappa, and N.K. Ratha. “Secure and Robust Iris Recognition Using Random Projections and Sparse Representations.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, no. 9 (September 2011): 1877–93.
- [23] Kang, Jin-Suk. “Mobile Iris Recognition Systems: An Emerging Biometric Technology.” *Procedia Computer Science, ICCS 2010*, 1, no. 1 (May 2010): 475–84.
- [24] Singh, Naveen, Dilip Gandhi, and Krishna Pal Singh. “Iris Recognition System Using a Canny Edge Detection and a Circular Hough Transform.” *International Journal of Advances in Engineering & Technology* 1, no. 2 (May 2011): 221–28.
- [25] Santiago Montero, R., O. Vázquez Bautista, M. Ornelas Rodríguez, A. Zamarrón Ramírez, and J. M. Carpio Valadez. “Reconocimiento de Iris Aplicando El Clasificador Asociativo de Patrones Multiclase.” In *1er. Congreso Internacional de Robótica Y Computación International Conference on Robotics and Computing*, 171–76, 2013.
- [26] Lawrence, S., C.L. Giles, Ah Chung Tsoi, and A.D. Back. “Face Recognition: A Convolutional Neural-Network Approach.” *IEEE Transactions on Neural Networks* 8, no. 1 (January 1997): 98–113.

- [27] He, Xiaofei, Shuicheng Yan, Yuxiao Hu, P. Niyogi, and Hong-Jiang Zhang. “Face Recognition Using Laplacianfaces.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27, no. 3 (March 2005): 328–340.
- [28] Sun, Zhan-Li, Kin-Man Lam, Zhao-Yang Dong, Han Wang, Qing-Wei Gao, and Chun-Hou Zheng. “Face Recognition with Multi-Resolution Spectral Feature Images.” *PLoS ONE* 8, no. 2 (February 13, 2013): e55700.
- [29] Huang, Zhiwu, Ruiping Wang, Shiguang Shan, and Xilin Chen. “Face Recognition on Large-Scale Video in the Wild with Hybrid Euclidean-and-Riemannian Metric Learning.” *Pattern Recognition, Discriminative Feature Learning from Big Data for Visual Recognition*, 48, no. 10 (October 2015): 3113–3124.
- [30] Zhang, Wei, Youmei Zhang, Lin Ma, Jingwei Guan, and Shijie Gong. “Multimodal Learning for Facial Expression Recognition.” *Pattern Recognition, Discriminative Feature Learning from Big Data for Visual Recognition*, 48, no. 10 (October 2015): 3191–3202.
- [31] Lippmann, Richard P. “Speech Recognition by Machines and Humans.” *Speech Communication* 22, no. 1 (July 1997): 1–15.
- [32] Potamianos, Alexandros, and Petros Maragos. “Time-Frequency Distributions for Automatic Speech Recognition.” *IEEE Transactions on Speech and Audio Processing* 9, no. 3 (March 2001): 196–200.
- [33] Hinton, G., Li Deng, Dong Yu, G.E. Dahl, A. Mohamed, N. Jaitly, A. Senior, et al. “Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups.” *IEEE Signal Processing Magazine* 29, no. 6 (November 2012): 82–97.
- [34] Deng, Li, and Xiao Li. “Machine Learning Paradigms for Speech Recognition: An Overview.” *IEEE Transactions on Audio, Speech, and Language Processing* 21, no. 5 (May 2013): 1060–1089.
- [35] Ramamoorthy, Aditya, Namrata Vaswani, Santanu Chaudhury, and Subhashis Banerjee. “Recognition of Dynamic Hand Gestures.” *Pattern Recognition, Kernel and Subspace Methods for Computer Vision*, 36, no. 9 (September 2003): 2069–2081.
- [36] Yörük, Erdem, Ender Konukoglu, Bülent Sankur, and Jérôme Darbon. “Shape-Based Hand Recognition.” *IEEE Transactions on Image Processing*, 2006.
- [37] Ge, S. S., Y. Yang, and T. H. Lee. “Hand Gesture Recognition and Tracking Based on Distributed Locally Linear Embedding.” *Image and Vision Computing* 26, no. 12 (December 1, 2008): 1607–1620.
- [38] Singha, Joyeeta, and Karen Das. “Hand Gesture Recognition Based on Karhunen-Loeve Transform.” *arXiv:1306.2599 [cs]*, June 11, 2013.

- [39] Ohn-Bar, E., and M.M. Trivedi. "Hand Gesture Recognition in Real Time for Automotive Interfaces: A Multimodal Vision-Based Approach and Evaluations." *IEEE Transactions on Intelligent Transportation Systems* 15, no. 6 (December 2014): 2368–2377.
- [40] Yu, Cheng-Bo, Hua-Feng Qin, Yan-Zhe Cui, and Xiao-Qian Hu. "Finger-Vein Image Recognition Combining Modified Hausdorff Distance with Minutiae Feature Matching." *Interdisciplinary Sciences: Computational Life Sciences* 1, no. 4 (November 14, 2009): 280–289.
- [41] Chunyi, Lin, Li Mingzhong, and Sun Xiao. "A Finger Vein Recognition Algorithm Based on Gradient Correlation." *AASRI Procedia*, AASRI Conference on Computational Intelligence and Bioinformatics, 1 (2012): 40–45.
- [42] Liu, Zhi, and Shangling Song. "An Embedded Real-Time Finger-Vein Recognition System for Mobile Devices." *IEEE Transactions on Consumer Electronics* 58, no. 2 (May 2012): 522–527.
- [43] Kiranmai, V. Bhanu, K. Amruthavally, and G. Harish. "An Embedded Real-Time Finger-Vein Recognition System for Security Levels." *International Journal of Application or Innovation in Engineering & Management (IJAIEM)* 2, no. 6 (June 2013): 478–484.
- [44] Xi, Xiaoming, Gongping Yang, Yilong Yin, and Xianjing Meng. "Finger Vein Recognition with Personalized Feature Selection." *Sensors* 13, no. 9 (August 22, 2013): 11243–11259.
- [45] Baltzakis, H., and N. Papamarkos. "A New Signature Verification Technique Based on a Two-Stage Neural Network Classifier." *Engineering Applications of Artificial Intelligence* 14, no. 1 (February 2001): 95–103.
- [46] Galbally, Javier, Moises Diaz-Cabrera, Miguel A. Ferrer, Marta Gomez-Barrero, Aythami Morales, and Julian Fierrez. "On-Line Signature Recognition through the Combination of Real Dynamic Data and Synthetically Generated Static Data." *Pattern Recognition* 48, no. 9 (September 2015): 2921–2934.
- [47] Justino, Edson J. R., Flávio Bortolozzi, and Robert Sabourin. "A Comparison of SVM and HMM Classifiers in the off-Line Signature Verification." *Pattern Recognition Letters* 26, no. 9 (July 1, 2005): 1377–1385.
- [48] Chadha, Ankit, Neha Satam, and Vibha Wali. "Biometric Signature Processing & Recognition Using Radial Basis Function Network." *arXiv:1311.1694 [cs]*, November 7, 2013.
- [49] Oliveira, Alessandro Bof de, Patrícia Rodrigues da Silva, and Dante Augusto Couto Barone. "A Novel 2D Shape Signature Method Based on Complex Network Spectrum." *Pattern Recognition Letters* 63 (October 1, 2015): 43–49.
- [50] Chen, Hui, and B. Bhanu. "Human Ear Recognition in 3D." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, no. 4 (April 2007): 718–737.

- [51] Yan, Ping, and K.W. Bowyer. "Biometric Recognition Using 3D Ear Shape." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29, no. 8 (August 2007): 1297–1308.
- [52] Ross, A., and A. Abaza. "Human Ear Recognition." *Computer* 44, no. 11 (November 2011): 79–81.
- [53] Chen, Jinhui, Tetsuya Takiguchi, and Yasuo Arikawa. "A Robust SVM Classification Framework Using PSM for Multi-Class Recognition." *EURASIP Journal on Image and Video Processing* 2015, no. 1 (March 11, 2015): 1–12.
- [54] Moghey, Mudit, Ajinkya R. Ghadge, and Sagar J. Dalvi. "Human Ear Recognition Using Geometric Features." *IARJSET* 2, no. 5 (May 22, 2015): 122–125.
- [55] Leyva, López, and Luis Octavio. "Máquinas asociativas alfa-beta con soporte vectorial." Tesis de Maestría, Instituto Politécnico Nacional. Centro de Investigación en Computación, 2008.
- [56] López-Leyva, Luis O., Itzamá López-Yáñez, Rolando Flores-Carapia, and Cornelio Yáñez-Márquez. "Análisis Del Índice Metropolitano de La Calidad Del Aire de La Ciudad de México (IMECA) Mediante Las Máquinas Asociativas Alfa-Beta Con Soporte Vectorial." In *Proc. de La V Conferencia Internacional Medio Ambiente Siglo XXI, MAS XXI*. Santa Clara, Cuba, 2007.
- [57] López-Leyva, Luis O., Cornelio Yáñez-Márquez, and I. López-Yáñez. "A New Efficient Model of Support Vector Machines: Alpha-Beta SVM." In *Proc. 23rd. ISPE International Conference on CAD/CAM Robotics and Factories of the Future*. Bogotá, Colombia, 2007.
- [58] Villarreal, Solís, José Francisco, Cornelio Yáñez Márquez, and Sergio Suárez Guerra. "Reconocimiento Automático de Voz Emotiva Con Memorias Asociativas Alfa-Beta SVM." *Polibits*, no. 44 (December 2011): 19–23.
- [59] Yáñez-Márquez, Cornelio, Itzamá López-Yáñez, and Guadalupe de la Luz Sáenz Morales. "Analysis and Prediction of Air Quality Data with the Gamma Classifier." In *Progress in Pattern Recognition, Image Analysis and Applications*, 651–58. Springer Berlin Heidelberg, 2008.
- [60] Itzamá López-Yáñez, Amadeo José Argüelles-Cruz. "Pollutants Time-Series Prediction Using the Gamma Classifier." *International Journal of Computational Intelligence Systems* 4, no. 4 (2011): 680–711.
- [61] Itzamá López-Yáñez, Cornelio Yáñez-Márquez. "Application of the Gamma Classifier to Environmental Data Prediction." In *Electronics, Robotics and Automotive Mechanics Conference*, 80–84, 2008.
- [62] Itzamá López-Yáñez, Leonid Sheremetov. "Associative Model for the Forecasting of Time Series Based on the Gamma Classifier." *Lecture Notes in Computer Science* 7914 (2013): 304–13.

- [63] López Yáñez, Itzamá. “Clasificador automático de alto desempeño.” Tesis de Maestría, Instituto Politécnico Nacional - Centro de Investigación en Computación, 2007.
- [64] López Yáñez, Itzamá. “Teoría y aplicaciones del clasificador asociativo gamma.” Tesis de Doctorado, Instituto Politécnico Nacional. Centro de Investigación en Computación, 2011.
- [65] Lopez-Yanez, Itzama, Cornelio Yanez-Marquez, and Victor Manuel. “Forecasting Air Quality Data with the Gamma Classifier.” In *Pattern Recognition*, edited by Peng-Yeng Yin. InTech, 2009.
- [66] Yáñez, Itzamá López, Cornelio Yáñez Márquez, Oscar Camacho Nieto, and Amadeo José Argüelles Cruz. “Predicción de la concentración de contaminantes atmosféricos basada en un clasificador asociativo de patrones.” *Revista Facultad de Ingeniería Universidad de Antioquia*, 2011.
- [67] Ruíz E., María E., Miguel Morales S., and Yahir Hernández M. “Una Estrategia de Segmentación de Imágenes Digitales de Huellas Dactilares Latentes.” *RIEE&C* 9, no. 1 (July 2011): 1–6.
- [68] Aguilar, Gualberto, Gabriel Sánchez, Karina Toscano, Mariko Nakano, and Héctor Pérez. “Reconocimiento de Huellas Dactilares Usando Características Locales.” *Revista Facultad de Ingeniería* 0, no. 46 (December 11, 2013): 101–9.
- [69] Aguilar-Torres, G., G. Sánchez-Pérez, K. Toscano-Medina, and H. Pérez-Meana. “Fingerprint Recognition Using Local Features and Hu Moments.” *Journal of Applied Research and Technology* 10, no. 5 (October 2012): 745–54.
- [70] Guo, Zicheng, and Richard W. Hall. “Parallel Thinning with Two-Subiteration Algorithms.” *Commun. ACM* 32, no. 3 (March 1989): 359–73.
- [71] Tico, M., and P. Kuosmanen. “An Algorithm for Fingerprint Image Postprocessing.” In *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers, 2000*, 2:1735–39 vol.2, 2000.
- [72] De León Santiago, Juan Luis Díaz, and Cornelio Yáñez Márquez. *Introducción a La Morfología Matemática de Conjuntos*. Instituto Politécnico Nacional, 2003.
- [73] Berners-Lee, T., R. T. Fielding, and H. Frystyk Nielsen. *Hypertext Transfer Protocol -- HTTP/1.0*. Internet draft, IETF, 1995.
- [74] Fielding, R., J. C. Mogul, H. Frystyk, L. Masinter, and T. Berners-Lee. *Hypertext Transfer Protocol -- HTTP/1.1*. Internet draft, IETF, 1999.
- [75] Ravi, J., Getha, K. S., Anitha, T. N., and Raja K. B. “Biomodal Biometric System using Multiple Transformation Features of Fingerprint and Iris.” Proc. of Int. Colloquiums on Computer Electronics Electrical Mechanical and Civil 2011.

- [76] Abdolahi, Mohamad, Mohamadi Majid, and Mehdi, Jafari, “Multimodal Biometric system Fusion Using Fingerprint and Iris with Fuzzy Logic.” *International Journal of Soft Computing and Engineering (IJSCE)*, January 2013.
- [77] Yáñez Márquez, Cornelio. “Memorias Asociativas Basadas En Relaciones de Orden Y Operaciones Binarias.” Tesis de Doctorado, Instituto Politécnico Nacional. Centro de Investigación en Computación, 2002.
- [78] Berners-Lee, Tim, Robert Cailliau, Ari Luotonen, Henrik Frystyk Nielsen, and Arthur Secret. “The World-Wide Web.” *Communications of the ACM* 37, no. 8 (1994): 76–82.
- [79] Schulzrinne, H. “World Wide Web: Whence, Whither, What Next?” *IEEE Network* 10, no. 2 (March 1996): 10–17.
- [80] Wei, Chen, Richard Khoury, and Simon Fong. “Web 2.0 Recommendation Service by Multi-Collaborative Filtering Trust Network Algorithm.” *Information Systems Frontiers* 15, no. 4 (September 2, 2012): 533–551.
- [81] Díaz, Oscar, Gorka Puente, Javier Luis Cánovas Izquierdo, and Jesús García Molina. “Harvesting Models from Web 2.0 Databases.” *Software & Systems Modeling* 12, no. 1 (February 24, 2011): 15–34.
- [82] Hoetzlein, R.C. “Graphics Performance in Rich Internet Applications.” *IEEE Computer Graphics and Applications* 32, no. 5 (September 2012): 98–104.
- [83] Jacobs, Ian, Jeff Jaffe, and Philippe Le Hegaret. “How the Open Web Platform Is Transforming Industry.” *IEEE Internet Computing* 16, no. 6 (November 2012): 82–86.
- [84] Kim, E. N., D. P. Schissel, G. Abla, S. Flanagan, and X. Lee. “Web-Based (HTML5) Interactive Graphics for Fusion Research and Collaboration.” *Fusion Engineering and Design*, Proceedings of the 8th IAEA Technical Meeting on Control, Data Acquisition, and Remote Participation for Fusion Research, 87, no. 12 (December 2012): 2045–2051.
- [85] Gonzalez, Rafael C., and Richard E. Woods. *Digital Image Processing*. Pearson Education India, 2008: 23-118.
- [86] Bhowmik, Pankaj, Kishore Bhowmik, Mohammad Nurul Azam, and Mohammed Wahiduzzaman Rony. “Fingerprint Image Enhancement And It’s Feature Extraction For Recognition.” *International Journal of Scientific & Technology Research* 1, no. 5 (June 2012): 117–21.
- [87] Ruíz E., María E., Miguel Morales S., and Yahir Hernández M. “Una Estrategia de Segmentación de Imágenes Digitales de Huellas Dactilares Latentes.” *RIEE&C* 9, no. 1 (July 2011): 1–6.
- [88] Acevedo, Betancur, Julián Andrés, Prieto Ortiz, Flavio Augusto, Osorio Londoño, and Gustavo Adolfo. “SEGMENTATION OF COFFEE BEANS BY MEANS OF SEEDED



- REGION GROWING TECHNIQUES.” *Revista Facultad Nacional de Agronomía, Medellín* 59, no. 1 (June 2006): 3311–33.
- [89] Tico, M., and P. Kuosmanen. “An Algorithm for Fingerprint Image Postprocessing.” In *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers, 2000*, 2:1735–39 vol.2, 2000.
- [90] Fraz, M. Moazam, Abdul Basit, Paolo Remagnino, Andreas Hoppe, and S. A. Barman. “Retinal Vasculature Segmentation by Morphological Curvature, Reconstruction and Adapted Hysteresis Thresholding.” In *Emerging Technologies (ICET), 2011 7th International Conference on*, 1–6. IEEE, 2011.
- [91] Gupta, Sudha, and Viral Doshi. “Iris Recognition System Using Biometric Template Matching Technology.” *International Journal of Computer Applications* 1, no. 2 (2010): 1–4.
- [92] Chouhan, Bhawna, and Shailja Shukla. “Analysis of Statistical Feature Extraction for Iris Recognition System Using Laplacian of Gaussian Filter.” *International Journal of Applied Engineering Research, Dindigul* 1, no. 3 (2010).
- [93] Wan, Jing, Xiaofu He, and Pengfei Shi. “An Iris Image Quality Assessment Method Based on Laplacian of Gaussian Operation.” In *MVA*, 248–51. Citeseer, 2007.
- [94] De León Santiago, Juan Luis Díaz, and Cornelio Yáñez Márquez, *Memorias Morfológicas Heteroasociativas; Serie Verde* 57, 2001.
- [95] Boser, Bernhard E., Isabelle M. Guyon, and Vladimir N. Vapnik. “A Training Algorithm for Optimal Margin Classifiers.” In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 144–52. ACM, 1992.
- [96] Cortes, Corinna, and Vladimir Vapnik. “Support-Vector Networks.” *Machine Learning* 20, no. 3 (1995): 273–97.
- [97] Burges, Christopher JC. “A Tutorial on Support Vector Machines for Pattern Recognition.” *Data Mining and Knowledge Discovery* 2, no. 2 (1998): 121–67.
- [98] Flores Carapia, Rolando. “Memorias Asociativas Alfa-Beta basadas en el código Johnson-Möbius modificado.” Tesis de Maestría, Instituto Politécnico Nacional - Centro de Investigación en Computación, 2006.
- [99] Lourenço, André, Hugo Plácido Da Silva, Carlos Carreiras, Ana Priscila Alves, and Ana LN Fred. “A Web-Based Platform for Biosignal Visualization and Annotation.” *Multimedia Tools and Applications* 70, no. 1 (2014): 433–60.