

Universidad Autónoma de Baja California



Facultad de Ingeniería Ensenada



Mestría y Doctorado en Ciencias e Ingeniería

Encriptado caótico de información biométrica para el control de accesos

TESIS

que para cubrir parcialmente los requisitos necesarios para
obtener el grado de

MAESTRO EN INGENIERÍA

Presenta

Ing. Fausto Abundiz Pérez

Ensenada, Baja California, Diciembre de 2010.

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA
UNIDAD ENSENADA

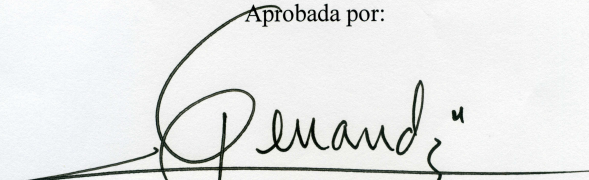
**Encriptado caótico de información biométrica para el control de
acceso**

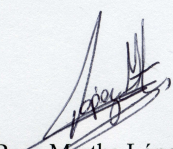
TESIS

Que para obtener el grado de maestría en ingeniería presenta:

Ing. Fausto Abundiz Pérez

Aprobada por:


Dr. César Cruz Hernández
Director de tesis


Dr. Rosa Martha López Gutiérrez
Miembro del comité


Dr. Enrique Efraín García Guerrero
Miembro del comité

Ensenada Baja California, México. Diciembre de 2010

Universidad Autónoma de Baja California

Facultad de Ingeniería - Ensenada

Mestría y Doctorado en Ciencias e Ingeniería

**Encriptado caótico de información biométrica para el control
de accesos**

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el grado de

MAESTRO EN INGENIERÍA

Presenta:

Ing. Fausto Abundiz Pérez

Aprobada por:

Dr. César Cruz Hernández
Director de Tesis

Dr. Rosa Martha López Gutiérrez
Miembro del Comité

Dr. Enrique Efrén García Guerrero
Miembro del Comité

Ensenada, Baja California, Diciembre de 2010.

Resumen de la tesis de **Fausto Abundiz Pérez**, presentada como requisito parcial para la obtención del grado de MAESTRO EN INGENIERÍA del programa de Maestría y Doctorado en Ciencias e Ingeniería (MYDCI) de la UABC. Ensenada, Baja California, México. Diciembre de 2010.

Encriptado caótico de información biométrica para el control de accesos

Resumen aprobado por:

Dr. César Cruz Hernández
Director de Tesis

El presente trabajo de tesis versa sobre el encriptado de información biométrica para su empleo en los sistemas de control de accesos. El interés de implementar un encriptado caótico a la información biométrica (en nuestro caso particular, huella dactilar) en los sistemas de control de accesos, radica en proporcionar una posible solución al problema de seguridad en algunas líneas de comunicación de los sistemas de control de accesos. Para realizar el encriptado de la información biométrica se recurre a un sistema caótico en tiempo discreto que presenta un comportamiento hipercaótico brindando así una mayor seguridad al encriptado. En la realización de las pruebas para comprobar la eficiencia del encriptado caótico, se tomaron algunas consideraciones en los sistemas biométricos de control de accesos basados en huellas dactilares, debido a esto en el trabajo se presenta una alternativa de comparación por medio del histograma de la imagen de la huella dactilar. Como resultado final del trabajo de tesis se obtuvo un cifrado caótico satisfactorio en este tipo de sistemas.

Palabras clave: Encriptado caótico, comunicaciones seguras, sistema biométrico, comparación de histogramas, sistemas de control de acceso, identificador biométrico de huellas dactilares.

Dedicatoria

A mi madre Laureana Pérez Leonidez, a mi padre Cleofas Abundis Tecorral, a mi hermano Eduardo Abundiz Pérez y a mi hermana Auria Abundis Pérez, quienes con su apoyo, comprensión, consejos y cariños, que me brindaron, provocaron en mí el entusiasmo de culminar el presente trabajo de tesis. Por dicho motivo les dedico este trabajo.

Agradecimientos

Quiero expresar mi agradecimiento:

A mi familia, Laureana Pérez Leonidez, Cleofas Abundis Tecorral, Eduardo Abundiz Pérez y Auria Abundis Pérez, quienes en el transcurso de mi vida, siempre me apoyan, aconsejan, educan, asesoran y me alientan a superarme cada día más. Por ese motivo los quiero mucho ya que ustedes son la parte más importante de mi vida.

Al Dr. César Cruz Hernández quien día con día me asesoro y motivo de manera paciente en cada uno de los procesos realizados para culminar el presente trabajo. Además de ello me alentó a proseguir con mis estudios académicos, lo cual, le agradezco sinceramente.

A mi comité de tesis: Dr. César Cruz Hernández, Dr. Rosa Martha López Gutiérrez y Dr. Enrique Efrén García Guerrero. Quienes con su apoyo, guía, sugerencias, confianza y amistad me motivaron a culminar mis estudios del grado de maestría.

A mis compañeros de cubículo de maestría, quienes me brindaron su sincera amistad y apoyo durante la realización del presente trabajo.

Al M.C. Everardo Inzunza G. por contribuir en el desarrollo del presente proyecto.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT), quienes me proporcionaron una beca, la cual, fue un gran apoyo económico para realizar mis estudios de posgrados en Facultad de Ingeniería Campus Ensenada de la Universidad Autónoma de Baja California (UABC).

Y a todos mis amigos, que me alentaron a proseguir con mis estudios de posgrado sinceramente muchas gracias por su apoyo.

Contenido

1. Introducción	1
1.1. Motivación	1
1.2. Descripción del problema de estudio	2
1.3. Objetivos	5
1.3.1. Objetivo general	5
1.3.2. Objetivos específicos	5
1.4. Propuesta de solución	5
1.5. Organización del manuscrito de tesis	6
2. Biometría	8
2.1. Introducción	8
2.2. Algunos antecedentes históricos	8
2.3. Biometría	9
2.4. Identificadores biométricos	9
2.5. Sistema biométrico	11
2.6. Estructura de un sistema biométrico	12
2.7. Aplicaciones de los sistemas biométricos	14
2.8. Ataques a puntos vulnerables	15
2.9. Conclusión	18
3. Criptografía	19
3.1. Introducción	19
3.2. Criptografía	19
3.3. Algunos antecedentes históricos	20
3.4. Clasificación de la criptografía	21
3.5. Algoritmos criptográficos	23
3.5.1. Algoritmo Blowfish	23
3.5.2. Algoritmo AES	23
3.5.3. Algoritmo IDEA	24
3.5.4. Algoritmo DES	26
3.5.5. Algoritmo RSA	27
3.5.6. Algoritmo DSA	28
3.6. Alternativas de encriptado	30
3.7. Aplicaciones de criptografía	30

3.8. Conclusión	31
4. Caos	32
4.1. Introducción	32
4.2. ¿Qué es el caos?	32
4.3. Algunos antecedentes históricos	33
4.4. Sistemas caóticos	35
4.5. Aplicaciones del caos	40
4.6. Conclusión	40
5. Sistema biométrico con base en huellas dactilares	41
5.1. Introducción	41
5.2. Sistema biométrico de huellas dactilares	41
5.2.1. Identificador biométrico de huellas dactilares	41
5.2.2. Lectura de la huella dactilar	42
5.2.3. Escenario de distribución del sistema biométrico	43
5.3. Alternativa de comparación	48
5.4. Alternativa de encriptado	49
5.4.1. Sistema caótico como encriptador	51
5.5. Ubicación del encriptado	52
5.6. Conclusiones	53
6. Pruebas y resultados	55
6.1. Introducción	55
6.2. Problema de verificación de un usuario	55
6.2.1. Definición del problema de verificación de un usuario	55
6.2.2. Pruebas realizadas para la posible solución del problema	56
6.3. Problema de identificación de múltiples usuarios	84
6.3.1. Definición del problema	85
6.3.2. Prueba realizada para la posible solución del problema	86
6.4. Conclusiones	107
7. Encriptado, desencriptado y comparación de imágenes	108
7.1. Introducción	108
7.2. Encriptado de imágenes	108
7.3. Desencriptado de imágenes	111
7.4. Comparación de las imágenes	112
7.5. Programas finales del sistema biométrico	114
7.5.1. Programa de registro	115
7.5.2. Programa de verificación	118
7.5.3. Programa de identificación	120
7.6. Conclusiones	123

8. Conclusiones generales y trabajo futuro	124
8.1. Conclusiones generales	124
8.2. Trabajos futuros	125
A. Programa en MatLab - Problema 1	126
A.1. Resultado numérico	126
A.2. Resultado gráfico	128
B. Programa en MatLab - Problema 2	130
B.1. Resultado numérico	130
B.2. Resultado gráfico	133
C. Programas en MatLab de encriptado, desencriptado y comparación de imágenes	135
C.1. Código de encriptado	135
C.2. Código de desencriptado	136
C.3. Código de comparación de las imágenes	136
D. Programas finales del sistema biométrico	139
D.1. Programa de registro de usuarios	139
D.2. Programa de verificación de usuario	142
D.3. Programa de identificación de usuario	145
E. Programas complementarios	149
E.1. Conversión de imagen a vector de datos	149
E.2. Sistema caótico	150
E.3. Conversión del vector de datos a imagen	151
E.4. Comparación de imágenes	152

Lista de Figuras

1.1.	Esquema de control de acceso remoto.	3
1.2.	Esquema de control de acceso remoto con etapa de encriptado.	4
1.3.	Proceso general de propuesta de encriptado y desencriptado.	5
2.1.	Diagrama a bloques de la etapa de inscripción en los sistemas biométricos.	13
2.2.	Diagrama a bloques de la etapa de verificación en los sistemas biométricos.	14
2.3.	Diagrama a bloques de la etapa de identificación en los sistemas biométricos.	14
2.4.	Aplicaciones de acceso electrónico que utilizan sistemas biométricos.	15
2.5.	Puntos vulnerables de ataques en los sistemas biométricos.	16
3.1.	Diagrama a bloques del cifrado de información utilizando criptografía de clave simétrica. .	21
3.2.	Diagrama a bloques del cifrado de información utilizando criptografía de clave asimétrica.	22
3.3.	Matriz de estado, bytes de entrada y bytes de salida.	24
3.4.	Proceso de encriptado del algoritmo IDEA.	25
3.5.	Proceso de encriptado del algoritmo DES	27
4.1.	Atractor de Lorenz	34
4.2.	Osilaciones no periódicas de Rössler.	37
4.3.	Osilaciones no periódicas de Lorenz.	37
4.4.	Sensibilidad a las condiciones iniciales de Rössler.	38
4.5.	Sensibilidad a las condiciones iniciales de Lorenz.	38
4.6.	Atractores extraños proyectados en el espacio tridimensional.	39
4.7.	Curva de Koch	40
5.1.	Lector de huellas dactilares Microsoft DG2-00004.	43
5.2.	Diagrama a bloques de componentes físicos básicos de un sistema biométrico con huella dactilar.	43
5.3.	Diagrama a bloques de un sistema biométrico Viridi 300.	44
5.4.	Diagrama a bloques del sistema biométrico STUU5.	46
5.5.	Diagrama de distribución del escenario seleccionado del sistema biométrico.	47
5.6.	Imagen y su histograma	48
5.7.	Diagrama a bloques que muestra como un histograma puede pertenecer a numerosas imágenes.	48
5.8.	Diagramas a bloques del proceso de encriptado y desencriptado por medio de encriptado caótico aditivo.	49
5.9.	Gráficas temporales	51

5.10. Atractores hipercaóticos del mapeo de Rössler.	52
5.11. Diagrama a bloques del sistema biométrico con las etapas de encriptado y desencriptado .	53
6.1. Diagrama ilustrativo para el problema 1	56
6.2. Huella dactilar de la base de datos	57
6.3. Diagrama esquemático para el problema 1	58
6.4. Imágenes del primer conjunto de cinco muestras	59
6.5. Histogramas del primer conjunto de cinco muestras de personas.	59
6.6. Sección ampliada de la gráfica de los histogramas del primer conjunto	60
6.7. Comparación de histograma, persona 1 del conjunto 1 versus usuario BD	61
6.8. Comparación de histograma persona 2 del conjunto 1 versus usuario BD	62
6.9. Comparación de histograma persona 3 del conjunto 1 versus usuario BD	63
6.10. Comparación de histograma persona 4 del conjunto 1 versus usuario BD	64
6.11. Comparación de histograma persona 5 del conjunto 1 versus usuario BD	65
6.12. Imágenes del segundo conjunto de cinco huellas	66
6.13. Histogramas de las muestras de cinco personas del segundo conjunto	67
6.14. Sección ampliada de la gráfica de los histogramas del segundo conjunto	67
6.15. Comparación de histograma persona 1 del conjunto 2 versus usuario BD	69
6.16. Comparación de histograma persona 2 del conjunto 2 versus usuario BD	70
6.17. Comparación de histograma persona 3 del conjunto 2 versus usuario BD	71
6.18. Comparación de histograma persona 4 del conjunto 2 versus usuario BD	72
6.19. Comparación de histograma persona 5 del conjunto 2 versus usuario BD	73
6.20. Resultados gráficos del programa en MatLab para el conjunto 2	74
6.21. Imágenes de las muestras de cinco personas del tercer conjunto.	75
6.22. Histogramas de las muestras de cinco personas del tercer conjunto.	75
6.23. Sección ampliada de la gráfica de los histogramas del tercer conjunto	76
6.24. Comparación de histograma persona 1 del conjunto 3 versus usuario BD	77
6.25. Comparación de histograma persona 2 del conjunto 3 versus usuario BD	78
6.26. Comparación de histograma persona 3 del conjunto 3 versus usuario BD	79
6.27. Comparación de histograma persona 4 del conjunto 3 versus usuario BD	80
6.28. Comparación de histograma persona 5 del conjunto 3 versus usuario BD	81
6.29. Resultados gráficos del programa en MatLab para el conjunto 3	82
6.30. Imágenes de las muestras renombradas de las personas de las etapas de calibración y verificación.	83
6.31. Resultados gráficos obtenidos de la comparación de 15 histogramas	84
6.32. Diagrama ilustrativo para el problema 2	85
6.33. Muestras de la base de datos del problema 2	86
6.34. Histograma del usuario 1 registrado en la base de datos del problema 2	86
6.35. Histograma del usuario 2 registrado en la base de datos del problema 2	87
6.36. Histograma del usuario 3 registrado en la base de datos del problema 2	87
6.37. Histograma del usuario 4 registrado en la base de datos del problema 2	87
6.38. Histograma del usuario 5 registrado en la base de datos del problema 2	88
6.39. Histograma obtenido en la comparación con la misma imagen	88
6.40. Diagrama esquemático para el problema 2	89

6.41. Muestras de personas de ingreso del problema 2	90
6.42. Histogramas de las muestras de las personas del 1 al 10.	91
6.43. Histogramas de las muestras de las personas del 11 al 20.	91
6.44. Resultado de comparación de la persona 1 de ingreso versus usuario 1 registrado	93
6.45. Resultado de comparación de la persona 2 de ingreso versus usuario 2 registrado	94
6.46. Resultado de comparación de la persona 3 de ingreso versus usuario 3 registrado	95
6.47. Resultado de comparación de la persona 4 de ingreso versus usuario 4 registrado	96
6.48. Resultado de comparación de la persona 5 de ingreso versus usuario 5 registrado	97
6.49. Resultado de comparación del programa en MatLab para la persona 1 de ingreso	98
6.50. Resultado de comparación del programa en MatLab para la persona 3 de ingreso	99
6.51. Resultado de comparación del programa en MatLab para la persona 5 de ingreso	100
6.52. Resultado de comparación del programa en MatLab para la persona 2 de ingreso	102
6.53. Resultado de comparación del programa en MatLab para la persona 4 de ingreso	103
6.54. Resultado de comparación del programa en MatLab para la persona 15 de ingreso	104
6.55. Resultado de comparación del programa en MatLab para la persona 17 de ingreso	105
6.56. Resultado de comparación del programa en MatLab para la persona 20 de ingreso	106
7.1. Diagrama de bloques del proceso de encriptado de imágenes de huellas dactilares.	109
7.2. Imagen de la huella dactilar capturada por el sensor que se encriptará.	109
7.3. Resultados proporcionados por el programa del Anexo C.1	109
7.4. Datos encriptados y estado caótico	110
7.5. Imagen de la huella dactilar encriptada.	110
7.6. Diagrama de bloques del proceso de desencriptado de imágenes de huellas dactilares.	111
7.7. Imagen de la huella dactilar recuperada.	111
7.8. Imagen de la huella dactilar recuperada con condiciones iniciales distintas.	112
7.9. Diagrama del proceso de comparación de imágenes de huellas dactilares.	113
7.10. Histogramas utilizados en el programa del anexo C.3	113
7.11. Histograma resultante de la comparación	114
7.12. Decisión de la comparación del programa del anexo C.3.	114
7.13. Ventana del programa al iniciar el proceso de registro.	115
7.14. Ventana que se despliega al pulsar el botón “Lectura de imagen”.	116
7.15. Ventana del programa de registro al finalizar el proceso.	117
7.16. Ventana que se despliega al pulsar el botón “Guardar como ...”	118
7.17. Ventana al iniciar el programa de verificación	119
7.18. Ventana del programa de verificación al finalizar el proceso.	120
7.19. Ventana al iniciar el programa de identificación.	121
7.20. Ventana del programa de identificación al finalizar el proceso.	122

Lista de Tablas

2.1. Aplicaciones forenses, gubernamentales y comerciales de sistemas biométricos basados en huellas dactilares.	15
5.1. Comparación entre identificadores biométricos. La información está simbolizada como: Alto (A), Medio (M) o Bajo (B).	42
6.1. Resultados de comparación del primer conjunto para el primer problema.	60
6.2. Resultados de comparación del segundo conjunto para el primer problema.	68
6.3. Resultados de comparación del tercer conjunto para el primer problema.	76
6.4. Resultados de comparación de los 3 conjuntos para el primer problema.	83
6.5. Resultados del segundo problema	92
6.6. Resultados de la comparación de histogramas para múltiples usuarios.	101

Capítulo 1

Introducción

1.1. Motivación

En la actualidad, se realizan múltiples y diversas operaciones o actividades que se encuentran comunicadas en red; por ejemplo, las transacciones bancarias que se realizan en internet, recepción y envío de correos electrónicos, tener acceso a un foro de la red, realizar compras en línea, tener acceso a un edificio que restringe su ingreso a personas no autorizadas. Cuando se mencionan estas actividades, siempre se presentan interrogantes relacionadas al tema de la seguridad e integridad al realizar las actividades mencionadas. Algunas de estas interrogantes podrían ser las siguientes: ¿son realmente seguras?, ¿qué tan confiable son?, ¿alguien podría suplantarme para realizar estas actividades u operaciones?.

A causa de las interrogantes mencionadas, se comenzó a construir sistemas automatizados de reconocimiento de identidad, para brindar confianza a los individuos que utilicen los servicios mencionados. Dentro de los sistemas de reconocimiento automatizados, se encuentran los que utilizan información *biométrica*, siendo éstos los que se encuentran en gran auge de uso en la actualidad y futuro.

En la actualidad hay diversos artículos escritos haciendo referencia al tema de la biometría, algunos ejemplos son las referencias [JRP04], [JR04], [WM] y [GAKD00]. También hay libros dedicados al tema de la implementación de la biometría como alternativa de identificación, algunos ejemplos de estos son las referencias [DZ04], [BOL04], [SR00] y [JFR03].

Actualmente los estudios realizados con respecto al tema de la biometría son para que el sistema realice más rápido su trabajo, ver por ejemplo la referencia [FLSCC06], o que la adquisición del identificador biométrico sea mejorada de una forma más eficiente, ver por ejemplo la referencia [HWJ98], entre otros aspectos. Aún cuando hay pocos artículos que atienden el tema de la seguridad en las líneas de comunicación de estos sistemas, una alternativa propuesta a este problema se reporta por ejemplo en las referencias [EMPU08], [VTDL03], [MCPMC06] y [UPJP04]. Motivados con la idea de aportar o contribuir en una alternativa para mantener la integridad y seguridad de la información biométrica utilizada

en los sistemas de control automático de acceso, en este trabajo de tesis se presenta la alternativa de *encriptado*.

En particular, este trabajo de tesis se centra principalmente en proponer como alternativa de encriptado para proporcionar seguridad a la información biométrica, la implementación de un *encriptado caótico* en los sistemas de control automático de acceso que utilizan la huella dactilar como identificador biométrico.

1.2. Descripción del problema de estudio

Mantener la privacidad de la información transmitida entre un emisor y un receptor, es una práctica muy demandada en la actualidad, pero lleva muchos años siendo aplicada y es bien conocido que sus primeras aplicaciones fueron bélicas, esto se debe a que la información que se transmite entre un regimiento a otro en el campo de batalla es crucial para obtener una victoria o derrota. Uno de los casos más conocidos, fue la utilización de una máquina de encriptado, durante la Segunda Guerra Mundial, llamada *Enigma*. Así como el caso mencionado, se podrían encontrar algunos otros aludidos en algunos libros de historia de la criptografía, que mencionen guerras disputadas entre pueblos, naciones, etc.

Actualmente, el tema de la *confidencialidad* de la información no solo es un tema que le interesa al ejército, sino que también tiene una gran importancia para el ciudadano común, esto se debe en gran parte al incremento de criminalidad que se sufre en las comunidades, siendo este tema de interés mundial, el cual, a su vez es un factor de importancia para implantar desconfianza en los individuos, haciendo que tengan temor de que si su información está realmente segura o algún extraño podría tener acceso a ella y ser usada de una manera inadecuada.

En la actualidad una de las medidas más utilizada para restringir el acceso de alguna pertenencia o datos de interés de un individuo, es utilizar sistemas que controlen el acceso de manera virtual o física. Ejemplos donde se restringa el acceso de manera virtual son, el acceso a las cuentas de correos electrónicos, sistema operativo de computadoras personales, acceso a las bases de datos, los perfiles de comunidades de internet, u otros servicios que se ofrezcan en la internet. Ejemplos de medios físicos, se mencionan el acceso a un laboratorio, cruces de fronteras entre países vecinos, acceso a edificios gubernamentales, a instalaciones industriales, etc. ver figura 1.1.

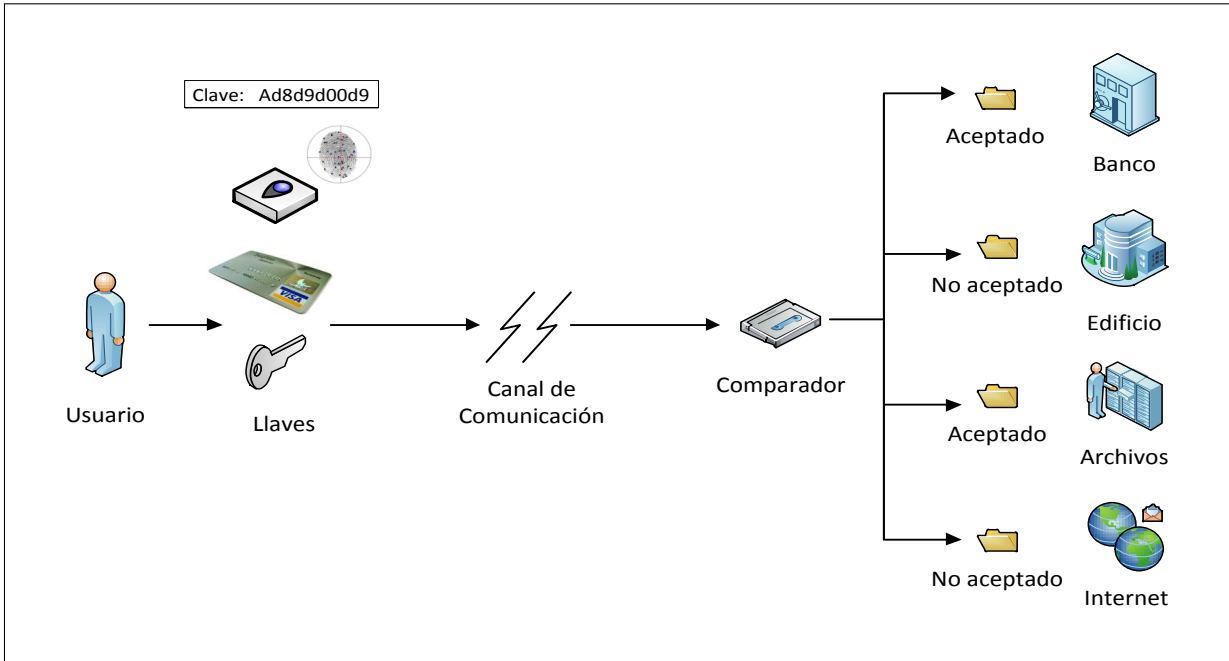


Figura 1.1: Esquema de control de acceso remoto.

Los **sistemas de control de accesos** son prácticamente sistemas que no permiten la utilización de un servicio o el acceso al mismo de *personas no autorizadas*. Este control se realiza por medio de una **llave**, la cual, puede ser una *tarjeta inteligente, clave numérica, llave física, entre otras*. Sin embargo, uno de los inconvenientes de utilizar alguna llave física o claves, es la facilidad con la que se puede extraviar, prestar o suplantar. Caso que no ocurre al utilizar un identificador biométrico, ésto debido a que solo se puede perder por medio de algún accidente físico que sufra la persona. En la actualidad, lo que se está utilizando más, son los sistemas que utilizan alguna información biométrica como llave para poder disponer del servicio, esto se debe a la gran ventaja que se tiene al utilizar algún identificador biométrico, esto se debe en gran parte a las características con las que disponen, las cuales, se mocionarán a detalle en el segundo capítulo de esta tesis.

Un gran inconveniente que presenta a menudo los sistemas biométricos de control de accesos, es que para abarcar un gran espacio de control, las etapas de estos sistemas se tendrán que ubicar de manera remota. Por ejemplo, el caso de una empresa que tiene varios edificios para el desempeño de sus actividades, los cuales, tienen aéreas restringidas a personas no autorizadas. Entonces tendrá que instalar un sistema de control de acceso que abarque todas estas áreas restringidas, por lo que tendrá que colocar una base de datos, que con la ayuda de algunos dispositivos electrónicos controlaran el acceso automático a cada una de las áreas restringidas.

Tomando en consideración la situación planteada, se puede notar que la distancia entre los edificios, es un factor que contribuye a que el sistema sea propenso a sufrir algún ataque en sus líneas de comunicación, entre los dispositivos electrónicos y su base de datos. Por tal

motivo, se tiene la necesidad de brindarle una protección a estas líneas de comunicación, siendo una opción la de *encriptar con caos la información biométrica que se transmite por los canales de comunicación*. Actualmente se utilizan métodos convencionales de encriptado, los cuales, están basados en sofisticados algoritmos matemáticos, ver figura 1.2.

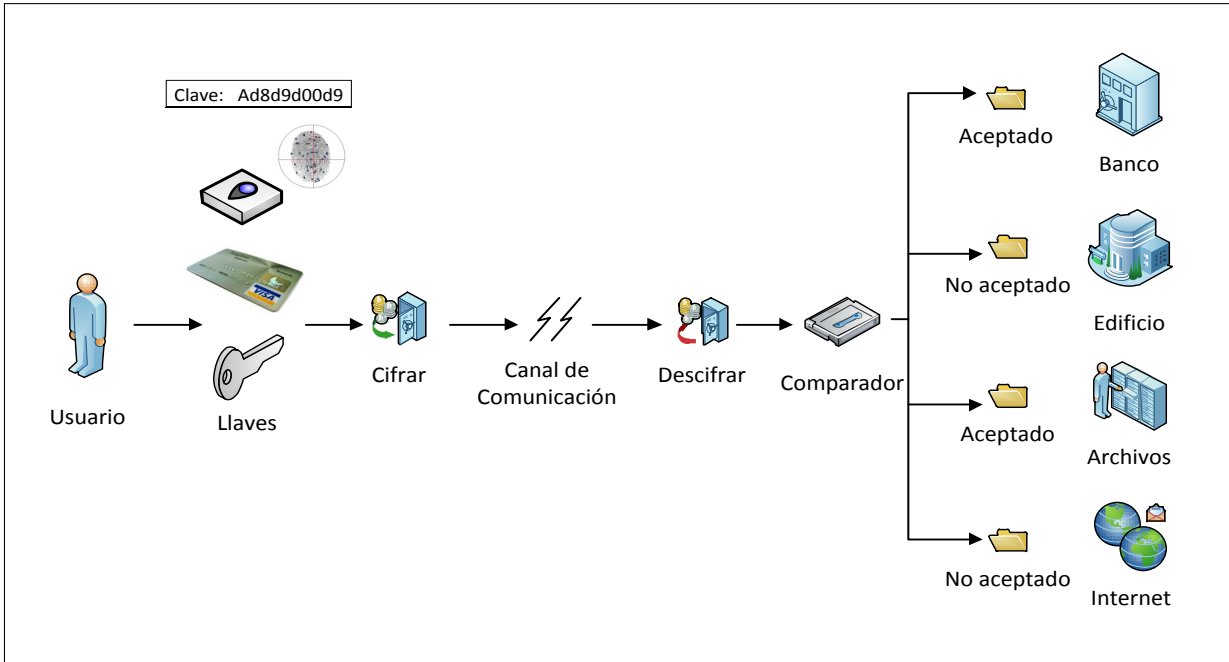


Figura 1.2: Esquema de control de acceso remoto con etapa de encriptado.

Hacer uso de un algoritmo de encriptado por periodos de tiempo muy largos, hace que el algoritmo sea más propenso a que un **criptoanalista** disponga de más tiempo para obtener la clave con que trabaja el algoritmo, el cual, tiene el propósito de encontrar alguna forma para descifrar el contenido de la información cifrada. Dicho esto, se puede entender que hacer uso de métodos de encriptado convencionales, los cuales, tiene bastante tiempo siendo usados, hace que la información utilizada en los sistemas biométricos sea propensa a ser extraída por algún intruso.

Actualmente, en los sistemas de control de acceso que utilizan información biométrica como llave, se les está poniendo una seguridad a la información biométrica, haciendo uso de métodos de encriptados convencionales. Motivados por el hecho de brindarle una seguridad eficiente a la información biométrica utilizada en estos sistemas de control de accesos. En este trabajo de tesis, se propone utilizar un **sistema caótico** para cifrar la información biométrica, con el propósito de contribuir a la seguridad de la misma.

1.3. Objetivos

1.3.1. Objetivo general

Aplicar el cifrado caótico a los sistemas de información biométrica para el control de accesos.

1.3.2. Objetivos específicos

- Seleccionar un identificador biométrico.
- Abordar el problema de verificación de un usuario para el control de accesos.
- Abordar el problema de identificación de múltiples usuarios para el control de accesos.
- Seleccionar un sistema caótico como medio encriptador.
- Diseñar un esquema de cifrado y descifrado, empleando el sistema caótico seleccionado para sistemas automáticos de control de accesos.
- Validar y probar, el sistema de encriptado así como descifrado de la información biométrica.

1.4. Propuesta de solución

La solución que se propone en este trabajo de tesis es utilizar un sistema caótico, para encriptar la información biométrica, esto con el motivo de brindarle una mayor seguridad, a algunas líneas de comunicación en el sistema biométrico, así como una seguridad extra en la base de datos, en la cual, se resguardará para su posterior utilización (ver figura 1.3).

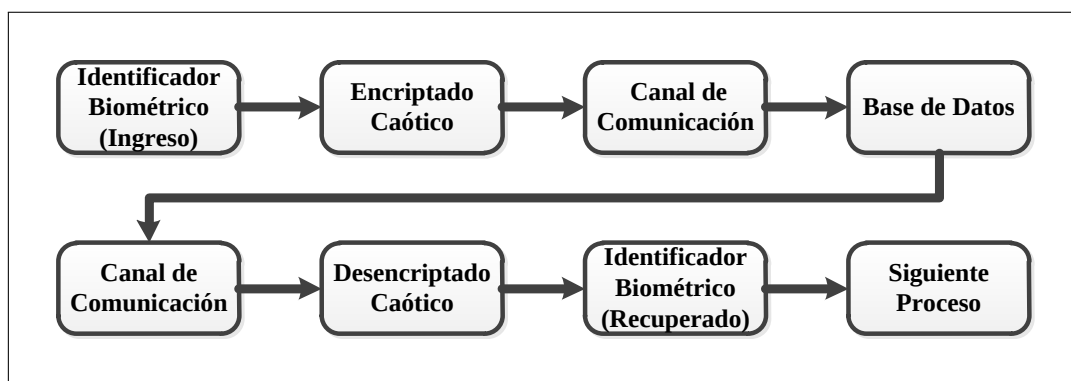


Figura 1.3: Proceso general de propuesta de encriptado y descifrado.

1.5. Organización del manuscrito de tesis

El manuscrito de la tesis está constituido principalmente por siete capítulos, de los cuales, el primero es el que está leyendo en este momento el lector, el cual, contiene la motivación de la realización de la tesis, la descripción del problema, los objetivos planteados, los objetivos específicos, la solución propuesta y la organización del manuscrito. El contenido de los siguientes capítulos se menciona a continuación.

El **capítulo dos**, contiene información que hará que el lector se introduzca un poco en el tema de la biometría para la mejor comprensión de la tesis, comenzando con una introducción, después de dará una definición de biometría, breves antecedentes sobre el tema, cual es, el significado de un sistema biométrico, estructura de los sistemas biométricos, algunas aplicaciones para los sistemas biométricos, sus puntos vulnerables y algunas soluciones aplicadas.

En el **capítulo tres**, se introduce al lector de una manera sencilla en el tema del encriptado, comenzando el capítulo con una introducción, para después mostrar una definición de criptografía, siguiendo con algunos antecedentes históricos, clasificaciones de la criptografía, algoritmos criptográficos utilizados, alternativas de encriptado y por último, algunas de sus aplicaciones.

El propósito del **capítulo cuatro** al igual que en los capítulos anteriores es introducir al lector en el tema de los sistemas caóticos, el cual, comenzará con una introducción para después exponer una definición del concepto de caos, posteriormente se presentan breves antecedentes históricos, una breve explicación de lo que son los sistemas caóticos, algunas de sus aplicaciones y por último se concluye el capítulo.

En el **capítulo quinto**, se presentan algunos aspectos que se tomaron en consideración para la realización de la tesis, este capítulo comienza con una introducción, después presenta algunos aspectos que se tomaron en consideración con respecto a los sistemas biométricos, por ejemplo tipo de identificador biométrico, lector utilizado, entre otros. Después se presenta la alternativa de comparación utilizada en el trabajo de tesis, posteriormente se presenta la alternativa de encriptado, la ubicación del encriptado y por último, se presentan algunas conclusiones con respecto a este capítulo.

En el **capítulo sexto**, se presentan resultados obtenidos en las pruebas realizadas en este trabajo de tesis, comenzando el capítulo con una breve introducción, después se expone el problema de verificación de un usuario así como las pruebas y resultados obtenidos en la búsqueda de su solución, posteriormente se presentan las pruebas realizadas del problema de identificación de múltiples usuarios, los resultados de las pruebas realizadas, los resultados obtenidos y por último se presentan algunas conclusiones de las pruebas realizadas .

En el **capítulo séptimo**, se presentan las pruebas realizadas en el proceso de encriptado, desencriptado y la comparación del identificador biométrico después de los dos procesos

anteriores. Posteriormente se presentan como trabajo final unos programas realizados en MatLab para las etapas de registro, identificación y verificación, la última sección de este capítulo se presentan las conclusiones de las pruebas realizadas.

En el *capítulo octavo* y último capítulo, se presentan las conclusiones generales arrojadas por el trabajo de tesis. Se mencionan los posibles trabajos futuros que se pueden realizar en esta dirección.

Capítulo 2

Biometría

2.1. Introducción

El objetivo de este capítulo es introducir al lector de este trabajo de tesis, al tema de la biometría de una manera breve y sencilla, por lo cual, se proporcionan los conceptos y fundamentos suficientes para que al lector no familiarizado con el tópico de los sistemas biométricos, adquiera la información necesaria para una mejor comprensión del alcance de este trabajo de tesis.

2.2. Algunos antecedentes históricos

Actualmente, el tema de la biometría está adquiriendo gran importancia en cuestión de los sistemas de seguridad. Sin embargo, muchas personas ignoran que la identificación de individuos por medio de una característica biométrica, es una práctica realizada de un largo tiempo atrás. Uno de los ejemplos que se puede comprender de una manera sencilla, es cuando reconocemos a una persona que pasa por la calle, solo observando su rostro o escuchando su voz, esto aun cuando lo hacemos de manera inconsciente, se podría considerar como un ***reconocimiento biométrico***. Hay evidencia arqueológica de que la huella dactilar era usada como identificador biométrico desde 7 000 a 6 000 años A. C. por los antiguos asirios y chinos, algunas de las evidencias son que en algunas arcillas se encontraron impresas imágenes de huellas dactilares.

Un hecho importante en cuanto a la identificación por medio de las características de las huellas dactilares, ocurrió a mediados de los 1800 donde se realizaron algunos estudios para establecer dos principios, que aún hoy en día se cumplen: no hay huellas dactilares que sean iguales en sus patrones de crestas entre dedos distintos de un mismo individuo y los patrones de las crestas de la huella dactilar son constantes durante toda la vida. Para mayor información, el lector interesado puede consultar la referencia [JBP99].

Otro aspecto de importancia en la identificación haciendo uso de la biometría aconteció a mediados del siglo XIX, donde Alphonse Bertillon Jefe de la División de Identificación Cri-

minal del Departamento de Policía de París, desarrolló e implementó la idea de un sistema de identificación de criminales por medio de una serie de mediciones del cuerpo humano. Pero esta idea fue sustituida debido a que se comenzó a ver que utilizar las características de la huella era más confiable que utilizar las medidas del cuerpo humano.

Sin embargo, el reconocimiento de una persona en esa época, se realizaba de forma manual, esto se refiere, a que un individuo revisaba minuciosamente la huella dactilar para identificarla por medio de sus características. Posteriormente, se crearon métodos que incrementaron la eficiencia de este proceso, haciendo la búsqueda más organizada y rápida, esto a su vez incrementó la demanda de utilizar este método de reconocimiento, por tal motivo, se comenzó a emplear la tecnología de la época con el propósito de adquirir la lectura de la huella por medio de dispositivos electrónicos, e implementar el reconocimiento automático por medio de procesos computarizados, dando como resultado el **sistema de reconocimiento automático de huellas** (AFIS, por sus siglas en inglés Automatic Fingerprint Identification Systems). Para mayor información, el lector interesado puede consultar la referencia [MMJP03].

En los años 90's se comenzaron a introducir en el mercado dispositivos de captura de huellas dactilares, su precio no era elevado y realizaban la captura más rápida, estos factores ayudaron aun más a expandir la utilización de reconocimiento por medio de huellas dactilares. En la actualidad, el reconocimiento de individuos por medio de sus huellas dactilares tiene diversas aplicaciones. Por ejemplo, lo que interesa en este trabajo de tesis, el **control automático de acceso**.

2.3. Biometría

Antes de mencionar a que se refiere la palabra biometría en este documento, se recurre a la definición del diccionario de la *Real Academia Española* [ESP92], el cual, la define de la siguiente manera “(De -bio y -metría). Estudio mensurativo o estadístico de los fenómenos o procesos biológicos.” Otra de las definiciones se puede encontrar en el diccionario que lleva por título en inglés, *The New Oxford American Dictionary* [JA01], el cual, define a la biometría de la siguiente manera, “The application of statistical analysis to biological data”. Entonces, en este documento, se entenderá como **biometría** al estudio estadístico de algunas características biológicas del ser humano. Actualmente en el área tecnológica, la biometría se aplica en la identificación o verificación de individuos de una manera automatizada por medio de las características biológicas de un individuo.

2.4. Identificadores biométricos

Se llama **identificador biométrico** a la característica biológica de una persona, que permite la identificación de la misma. Actualmente los identificadores biométricos que se utilizan en los sistemas biométricos, están clasificados en dos grupos: **los identificadores**

físicos y conductuales que describimos a continuación:

Identificadores físicos. Estos identificadores biométricos como su nombre lo dice, son partes físicas del cuerpo humano, utilizados para la identificación de una persona, ejemplos de estos identificadores son: huellas dactilares, geometría de la mano, iris, termografía facial y cornea.

Identificadores conductuales. Estos identificadores biométricos son características que describen una conducta propia de un individuo, como ejemplo de estos identificadores se pueden mencionar los siguientes: firma rubrica, forma de teclear, reconocimiento de voz y forma de caminar.

Aún cuando los rasgos de conductas o partes físicas de una persona puedan utilizarse para la identificación personal, no todos pueden ser considerados identificadores biométricos, para que sean considerados así, deben reunir las siguientes propiedades:

Universalidad. Esto se refiere a que todas las personas cuenten con esta característica.

Única. Esto se refiere a que debe ser único para cada persona y que no puede ser igual a otra.

Permanente. Esto se refiere a que la característica debe ser suficientemente invariante en un período largo de tiempo.

Cuantificación. Esto se refiere a que la característica biométrica pueda ser medida cuantitativamente.

Desempeño. Esto se refiere a la velocidad y precisión alcanzable para el reconocimiento, así como el funcionamiento, los factores ambientales que afectan a la precisión como a la velocidad.

Aceptabilidad. Esto se refiere a que tanto se acepta la implementación del identificador biométrico en la sociedad.

Fiabilidad. Este parámetro se refiere a la facilidad de engañar al sistema implementando el identificador biométrico.

Si el lector desea profundizar más en esta información, puede consultar por ejemplo las referencias [JBP99], [MMJP03], [JFR03] y [MJ04].

Nota: Uno de los identificadores biométricos que cumple satisfactoriamente con los atributos mencionados y el cual, es muy utilizado actualmente en los sistemas biométricos, es *la huella dactilar*. Debido a esto, se tomó la decisión de utilizar este identificador en el proyecto de tesis.

2.5. Sistema biométrico

En general, un sistema biométrico es un sistema de reconocimiento de patrones, el cual, determina la autenticidad de una característica fisiológica o conductual, llamada identificador biométrico de un individuo, con el propósito de usarlo para su posterior reconocimiento del individuo. Dependiendo de la aplicación de un sistema biométrico, puede llamarse ***sistema de identificación*** o ***sistema de verificación***, para mayor información el lector interesado consulte la referencia [JRP04]:

Sistema de Identificación. Es un sistema que reconoce a un individuo realizando comparaciones con la muestra (o patrón) biométrica de entrada al sistema biométrico, versus las muestras previas almacenadas en la base de datos. Esto con el motivo de establecer la *identidad del individuo* que previamente se registró en la base de datos. En estos sistemas se establece la identidad del individuo sin necesidad de afirmar una identidad. Se podría decir que la pregunta que se realiza en este sistema es ***¿quién soy yo?***.

Sistema de Verificación. Es un sistema que autentifica la identidad de una persona al comparar la muestra (o patrón) biométrica de ingreso al sistema biométrico, versus la muestra del individuo que previamente fue almacenada en la base de datos del sistema. En este sistema se acepta o rechaza al sujeto que se presenta afirmando una identidad. Se podría decir que la pregunta que se realiza en este sistema es ***¿soy yo quién afirmo ser?***.

La clasificación anterior de los sistemas biométricos no es la única, los sistemas biométricos también se pueden clasificar de acuerdo a los siguientes atributos:

- Sistema en línea o fuera de línea.
- Reconocimiento positivo o negativo.
- Cooperativo o no cooperativo.
- Modo de utilización consciente o inconsciente.
- Utilización habitual o no habitual.
- Supervisado o no supervisado.
- Ambiente controlado o no controlado.
- Aplicación pública o privada.
- Abierto o cerrado.

Las aplicaciones comerciales de los sistemas biométricos tienen popularmente los siguientes atributos: son cooperativos, se utilizan conscientemente, se utilizan habitualmente, de registro atendido, reconocimiento no atendido y ambiente no controlado. Si el lector desea mayor información puede consultar el libro de la referencia [MMJP03] de las páginas 4 a la 7.

Actualmente en el mercado hay diversas empresas que hacen uso de identificadores biométricos para la identificación o verificación automatizada de individuos, algunas de estas empresas son: ActivCard, AuthenTec, BIO-Key, Biometric Security Card, Bioscrypt, DigitalPersona, Fujitsu, Identix, Indivos, NEC, NITGen, Precise Biometrics, SecuGen, Sony, ST Microelectronics y Ultra-Scan.

2.6. Estructura de un sistema biométrico

La estructura general de un sistema biométrico está diseñada en base a cuatro módulos principales, que se describen a continuación:

Modulo del sensor. La función de este modulo es capturar el identificador biométrico de un individuo por medio de un dispositivo electrónico.

Modulo de extracción de características. La función de este modulo es procesar los datos del identificador biométrico, para obtener un conjunto de datos más representativos del identificador biométrico.

Modulo de comparación. La función de este modulo es comparar el conjunto de características de entrada al sistema biométrico, versus el conjunto de características almacenadas previamente en la base de datos.

Modulo de base de datos. La función de este modulo es almacenar los datos del identificador biométrico adquiridos en el proceso de registro.

La estructura de los sistemas biométricos que basan su decisión en la comparación de características de la huella dactilar, se encuentran compuestas fundamentalmente por dos etapas, las cuales, están constituidas principalmente por los cuatro módulos mencionados arriba.

Etapas de enrolamiento: La etapa de enrolamiento o inscripción en un sistema biométrico, es la etapa en la que se registra al usuario en la base de datos para tener autorización a un servicio. Para el proceso de enrolamiento en un sistema biométrico, se requieren principalmente realizar cuatro pasos: el primero, es la adquisición de la muestra del identificador biométrico, en este paso se adquiere una imagen de la huella por medio de un sensor, el segundo es el chequeo de calidad o preproceso de la muestra, en este paso,

la imagen adquirida pasa por algunos procesos para checar la calidad de la imagen y realizar algunos procesos para mejorar la muestra, el tercero es la extracción de características del identificador biométrico, en este proceso se extraen los patrones (o plantillas) de la huella, el cuarto es el almacenamiento de la muestra en una base de datos. Esto se puede entender mejor observando la figura 2.1.

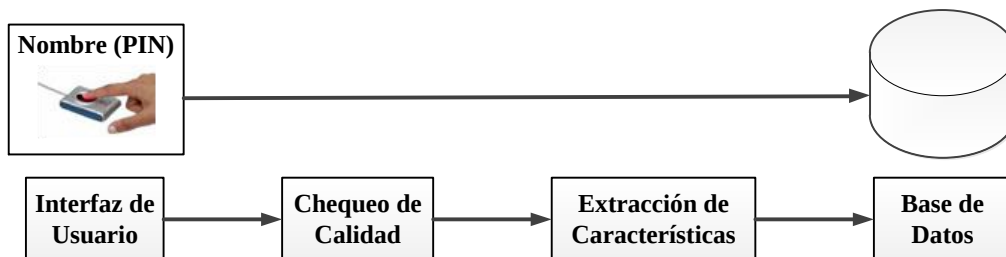


Figura 2.1: Diagrama a bloques de la etapa de inscripción en los sistemas biométricos.

Etapa de identificación o verificación: La etapa de verificación o identificación es la etapa en la que se realiza una comparación de la muestra de entrada al sistema biométrico versus la muestra almacenada en la base de datos, para determinar si la persona solicitante del servicio está inscrita en el sistema biométrico. Para el proceso de identificación o verificación en un sistema biométrico se requiere principalmente realizar cuatro pasos: el primero es la adquisición de la muestra del identificador biométrico, en este paso se adquiere una imagen de la huella por medio de un sensor, el segundo es la extracción de características del identificador biométrico en este proceso se extraen los patrones de la huella, el tercero es la comparación de la muestra de entrada versus la muestra almacenada en la base de datos, esta comparación tiene una pequeña diferencia dependiendo de su aplicación. Por ejemplo, si el sistema es para la verificación de usuarios, la comparación solamente se realiza con una huella en específico de la base de datos (ver figura 2.2), mientras que si el sistema es para la identificación de usuarios, la comparación se realiza con N huellas de la base de datos (ver figura 2.3). El cuarto paso es la toma de decisión del sistema, el cual, basa su respuesta dependiendo del resultado de la comparación.

Para mayor información el lector puede consultar por ejemplo las siguientes referencias [JRP04] y [MMJP03].

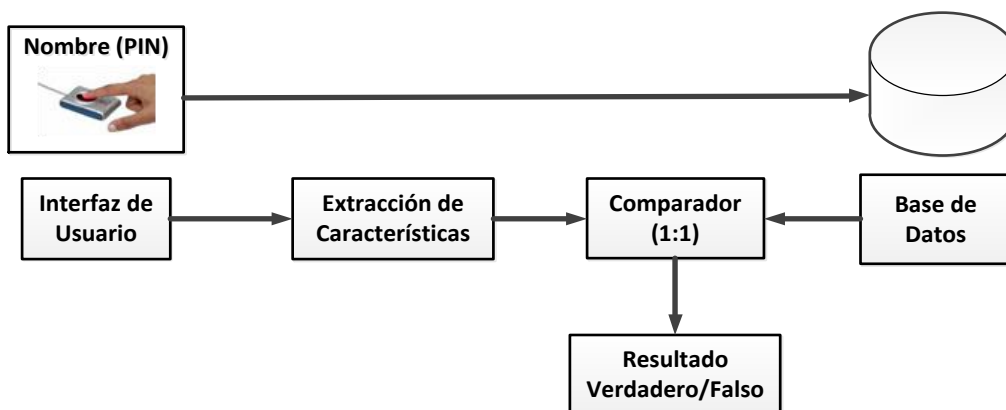


Figura 2.2: Diagrama a bloques de la etapa de verificación en los sistemas biométricos.

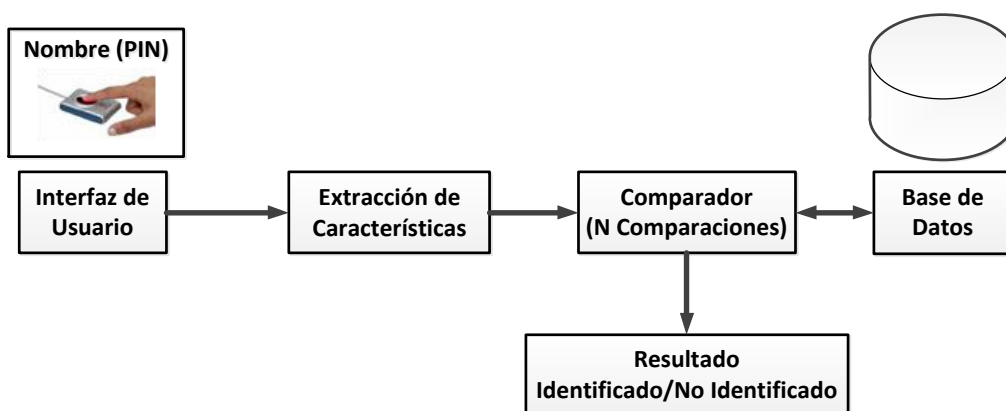


Figura 2.3: Diagrama a bloques de la etapa de identificación en los sistemas biométricos.

2.7. Aplicaciones de los sistemas biométricos

Los sistemas biométricos tienen diversas aplicaciones pero en esta sección se mencionan algunas basadas en huellas dactilares, siendo esta tecnología comúnmente asociada al *reconocimiento forense*, *identificación de criminales* y *sistemas de seguridad en prisiones*. Sin embargo, existen otras aplicaciones de los sistemas biométricos (ver tabla 2.1 y figura 2.4), los cuales, se pueden clasificar en dos grandes grupos:

1. Aplicaciones de control de acceso de algún servicio como cuentas de correos en internet, banca electrónica y comercio electrónico.
2. Aplicaciones de control de registros; por ejemplo, control de registros médicos y control de registros de emigración.

Para mayor información remitimos al lector interesado consultar la referencia [MMJP03].

Forense	Gubernamentales	Comerciales
Identificación de cadáveres Investigación criminal Identificación de terroristas Búsqueda de niños Entre otros	Tarjeta de ID nacional Correccionales Licencia de manejo Seguridad social Control de fronteras Control de pasaporte Entre otros	Inicio de sesión red Seguridad de datos electrónicos Acceso en internet ATM, tarjetas de crédito Control de acceso físico Teléfonos celulares Control de registros médicos Entre otros

Tabla 2.1: Aplicaciones forenses, gubernamentales y comerciales de sistemas biométricos basados en huellas dactilares.



Figura 2.4: Aplicaciones de acceso electrónico que utilizan sistemas biométricos.

2.8. Ataques a puntos vulnerables

Físicamente los procesos que realizan las dos etapas de los sistemas biométricos se encuentran distribuidas en diferentes dispositivos. Además, por lo general, los procesos de la etapa de registro se encuentra en áreas controladas, las cuales, son protegidas físicamente esto hace que sean menos propensas a algún ataque. Sin embargo, la etapa de verificación o identificación según sea la aplicación del sistema biométrico, no cuenta con esta seguridad por lo que, deja vulnerable los procesos que se realizan en esta etapa, los cuales, son propensos a posibles ataques. En la figura 2.5 se muestran los puntos de posibles ataques que

pueden sufrir los sistemas biométricos basados en huellas dactilares y mayormente enfocados en la etapa de verificación/identificación.

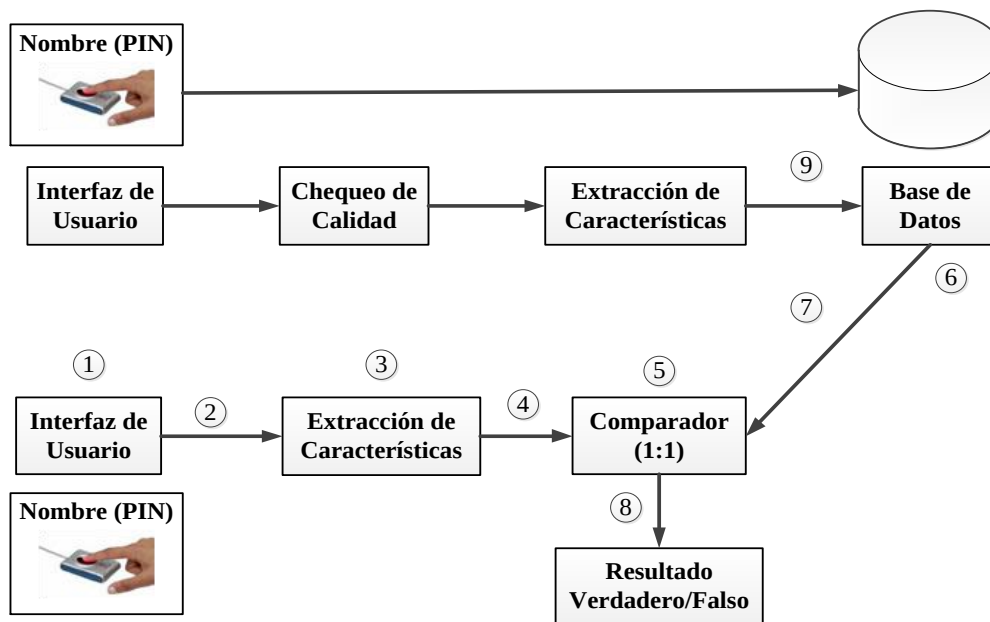


Figura 2.5: Puntos vulnerables de ataques en los sistemas biométricos de huellas dactilares.

Como se observa en la figura 2.5, se muestran numerados los puntos vulnerables de ataques que podrían sufrir los sistemas biométricos, a continuación se mencionarán de manera breve en que consiste cada uno de estos ataques.

① Ataque al elemento de censado (escáner)

Este tipo de ataque puede referirse a un ataque físico en contra del escáner de la huella dactilar. A continuación se mencionarán tres ejemplos de este tipo de ataque: Primer ejemplo, el escáner o su superficie pueden ser físicamente destruidas, esto con la intención de que ningún usuario pueda utilizar el sistema biométrico. Segundo ejemplo, al escáner se le puede presentar una falsa huella, esto se refiere a copias o impresiones sobre papel de la imagen de la huella, polímeros con la forma de la huella en tercera dimensión, moldes de polímeros gelatinosos, entre otros. Tercer ejemplo, una imagen digitalizada puede ser introducida entre el escáner y el circuito al que se encuentra conectado el escáner (convertidor A/D, microprocesador, etc.).

② Ataque al canal de comunicación entre los módulos del escáner y la extracción de características

En este tipo de ataque por lo general, se interrumpe la línea de comunicación para ob-

tener una imagen de la huella dactilar, la cual, puede utilizarse en un tiempo posterior para pasar por alto el proceso de la adquisición de la huella por medio del escáner.

③ **Ataque al modulo de extracción de características**

Este tipo de ataque se realizan por un programa malicioso (troyano), el cual, simula realizar el proceso de extracción para pasar por alto el verdadero proceso de extracción, para así, poder introducir características artificiales de la huella, las cuales, son generadas por el programa malicioso y siendo estas características las que se envían al modulo de comparación.

④ **Ataque al canal de comunicación entre los módulos de extracción de características y comparación**

En este tipo de ataque se interceptan el canal de comunicación para adquirir las características extraídas de las huellas de un usuario registrado en la base de datos. El motivo de este ataque es utilizar estas características en un tiempo futuro para pasar por alto los módulos del escáner y el de extracción de características.

⑤ **Ataque al modulo de comparación**

Este tipo de ataque se realiza por un programa malicioso (troyano), el cual, simula realizar el proceso de comparación para pasar por alto el verdadero proceso de comparación, para así, poder introducir el resultado de un punto de comparación generado artificialmente, esto podría contribuir a que un usuario registrado tenga un caso de falso rechazo o que una persona no registrada tenga un caso de falsa aceptación.

⑥ **Ataque a la base de datos**

Este tipo de ataque se realiza por un programa malicioso (troyano), el cual, simula el registro de datos en la de base de datos, para así, poder suministrar datos falsamente almacenados, los cuales, son generados artificialmente por el programa malicioso, estos datos pueden ser una plantilla registrada, el nombre del usuario, privilegios de acceso, entre otros. Este tipo de ataques puede ser realizado durante el tiempo que se efectúa la etapa de inscripción, durante el tiempo que se efectúa la etapa de verificación/identificación o podría efectuarse en la base de datos en cualquier momento.

⑦ **Ataque al canal de comunicación entre la base de datos y el modulo de comparación**

El canal de comunicación entre la base de datos y el modulo de comparación puede ser fisgoneado para robar la información grabada en la base de datos, la cual, se envía a través de este canal, esto con el motivo de utilizarla en un tiempo futuro.

⑧ **Ataque al canal de comunicación del modulo de comparación y la aplicación que requiere la verificación/identificación**

El canal de comunicación entre el modulo de comparación y la aplicación que requiere la verificación/identificación puede ser fisgoneada con el propósito de tener acceso a la respuesta resultante del modulo de comparación, esto con motivo de utilizar esta información obtenida en un tiempo futuro.

⑨ Ataque al canal de comunicación del modulo de extracción de características de la etapa de enrolamiento y la base de datos

Este canal de comunicación es propenso a ser interceptado para fisgonear, con el propósito de tener acceso a las características extraídas en el modulo anterior, con motivo de almacenarlas y poder hacer uso de estas características en un tiempo futuro.

Se puede notar que los ataques ②, ④, ⑦, ⑧ y ⑨, se podrían clasificar en el mismo tipo de ataque, debido a que estos ataques se realizan a los canales de comunicación de los sistemas biométricos, a los que nos podemos referir como ***ataques de reproducción***. También se puede notar que los ataques ①, ③, ⑤ y ⑥, son ataques que se realizan en alguno de los módulos del sistema biométrico, los cuales, comúnmente se realizan con **programas troyanos**.

Debido a la existencia de estos ataques a puntos vulnerables de los sistemas biométricos, se pueden implementar algunas acciones de protección para mitigar el riesgo de sufrir daños generados por ellos, entre estas acciones se utilizan las siguientes: datos aleatorios, retención de datos, detección en vivo, biometría múltiple, autenticación multifactor, criptografía, firmas digitales, limpieza de la red y seguridad física. Para mayor información consultar las referencias [MMJP03], [AA07], [FO06], [ORM10], [ANI08b] y [ANI08a].

2.9. Conclusión

La importancia que tienen los sistemas biométricos para el control de accesos aumenta con el pasar de los días. Actualmente como se ha mencionado, los sistemas biométrico están siendo utilizados en diversos servicios, por tal motivo, la seguridad de estos sistemas deberá mejorarse día con día, esto con el propósito de que el sistema conserve la confianza de los usuarios que utilicen el servicio. Lo cual, indica que la contribución de este trabajo de tesis en aspectos de seguridad podría influir a este propósito.

Capítulo 3

Criptografía

3.1. Introducción

El objetivo del presente capítulo, es proporcionar información necesaria relacionada al tópico de la criptografía, con el motivo de proporcionar al lector, el conocimiento básico para comprender la participación del tema en la tesis. Así, como también, podrá entender de mejor manera la contribución que podría proporcionar la acción de encriptado caótico propuesta en esta tesis, a la seguridad de los sistemas biométricos.

3.2. Criptografía

La **criptografía** es parte de la criptología, la cual, se encarga de estudiar los sistemas criptográficos siendo estos divididos en tres áreas de estudio, el criptoanálisis, la esteganografía y la criptografía. La definición de criptografía se menciona en diferentes fuentes literarias, por este motivo se proporcionan tres definiciones, que se consideran las más adecuados para este trabajo de tesis. Se comenzará con la definición presentada en el diccionario de la *Real Academia Española* [ESP92]. El cual define la criptografía de la siguiente manera:

Criptografía. Del griego $\chi\rho\upsilon\pi\tau\acute{o}\varsigma$, oculto y del sufijo *—grafía* que significa *escritura*; arte de escribir con clave secreta o de un modo enigmático.

La siguiente definición se tomó del diccionario de informática e internet [MIC04], que define a la criptografía como sigue.

“criptografía. La utilización de códigos para convertir los datos de modo que sólo un receptor específico sea capaz de leerlos utilizando una clave ... ”

La última definición se tomó del libro titulado “Handbook of applied cryptography” [MOV00], el cual, define la criptografía de la siguiente manera.

“La criptografía es el estudio de técnicas matemáticas relacionadas al aspecto de seguridad de información, tales como confidencialidad, integridad de datos, autenticación de identidad y autenticación de datos originales. ”

Después de leer las definiciones anteriores, se puede apreciar que la criptografía, se encuentra vinculada a la seguridad de la información, la cual, está encargada con cumplir las metas de confidencialidad, integridad, autenticación y no rechazo de los datos.

3.3. Algunos antecedentes históricos

En la actualidad existe mucha información en internet, en libros, en revistas, en artículos, etc. relacionada al tema de la criptografía, así como su evolución en el transcurso de la historia, una de las primeras evidencias que se podría considerar como antecedente de criptografía fue aportada por la civilización egipcia, pero su intención no era la de cifrar la información, simplemente sustituían algunos de sus jeroglíficos por otros similares que no se usaban comúnmente. También en la escritura mesopotámica, hay evidencia que en algunas ocasiones se intercambiaban algunos de sus signos por otros, pero a diferencia de los egipcios ellos si tenían la intención de cifrar la información escrita [TGC06].

Se considera que los espartanos fueron los primeros en diseñar un sistema criptográfico para cifrar la información, este método se llamaba ***escítala***, la cual, consistía en enrollar una tira de papel sobre un bastón, que tenía cierto diámetro y dimensión, lo que permitía cifrar o descifrar el mensaje [TGC06]. Otro método de encriptado fue implementado por Julio César llamado ***sustitución monoalfabética***, el cual consistía en recorrer cierto número de posiciones las letras del alfabeto esto para cambiar las letras de las palabras, dándole al escrito cierta protección para que no fuese entendido por cualquier persona, el texto era sólo legible para la persona que conocía el número de letras recorridas del alfabeto, ver por ejemplo [FD06].

Pasando por alto muchos acontecimientos históricos de la criptografía, se mencionarán sucesos de importancia referente al tema, que ocurrieron durante la Segunda Guerra Mundial, uno es lo ocurrido con el código que usó Estados Unidos durante esta guerra, en la cual, utilizó el ***dialecto indígena navajo*** para cifrar sus conversaciones, de este modo, se aseguraban de que nadie supiera que se estaba transmitiendo. Este método de encriptado fue muy efectivo, a tal grado que el código nunca fue decodificado por los japoneses [TAN03]. Otro aporte, fue realizado por los alemanes quienes inventaron una máquina llamada ***enigma***, la cual, podía cifrar y descifrar información [ALB09].

Con la invención de las computadoras se comenzó a desarrollar la ***criptografía moderna***, la cual, se implementó mediante complicados algoritmos matemáticos. Uno de los primeros métodos de encriptado moderno fue el ***DES*** (por sus siglas en inglés, Data Encryption Standard), el cual, después de algunas modificaciones sigue siendo empleado en el mundo, hasta nuestros días es el más conocido y difundido. Algunos métodos criptográficos

modernos se mencionan en las referencias [MA06], [ELG85] y [DKT07].

3.4. Clasificación de la criptografía

La criptografía se puede clasificar de diversas formas, pero en este trabajo se mencionará la clasificación de la criptografía de acuerdo al tipo de **clave** que utiliza, para el cifrado siendo esta clasificación la siguiente:

Criptografía simétrica. Este algoritmo de encriptado usa **la misma clave para cifrar y descifrar** la información (ver figura 3.1). El remitente (Alicia) usa la clave para poder cifrar la información, enviando el mensaje cifrado al destinatario (Bob) quien utilizará la misma clave para descifrar el mensaje. Utilizar la misma clave para los dos procesos (encriptado y descifrado) representa una desventaja, debido a que si algún intruso consigue la clave durante su envío al destinatario para que la conozca, el intruso podrá hacer uso de ella para realizar el proceso de cifrado o descifrado de la información en el momento que lo desee, siendo esta una oportunidad que le permitirá realizar algunos ataques a la información que se envían entre el remitente y el destinatario.

Algunos algoritmos de clave simétrica son: 3DES, AES, Blowfish e IDEA.

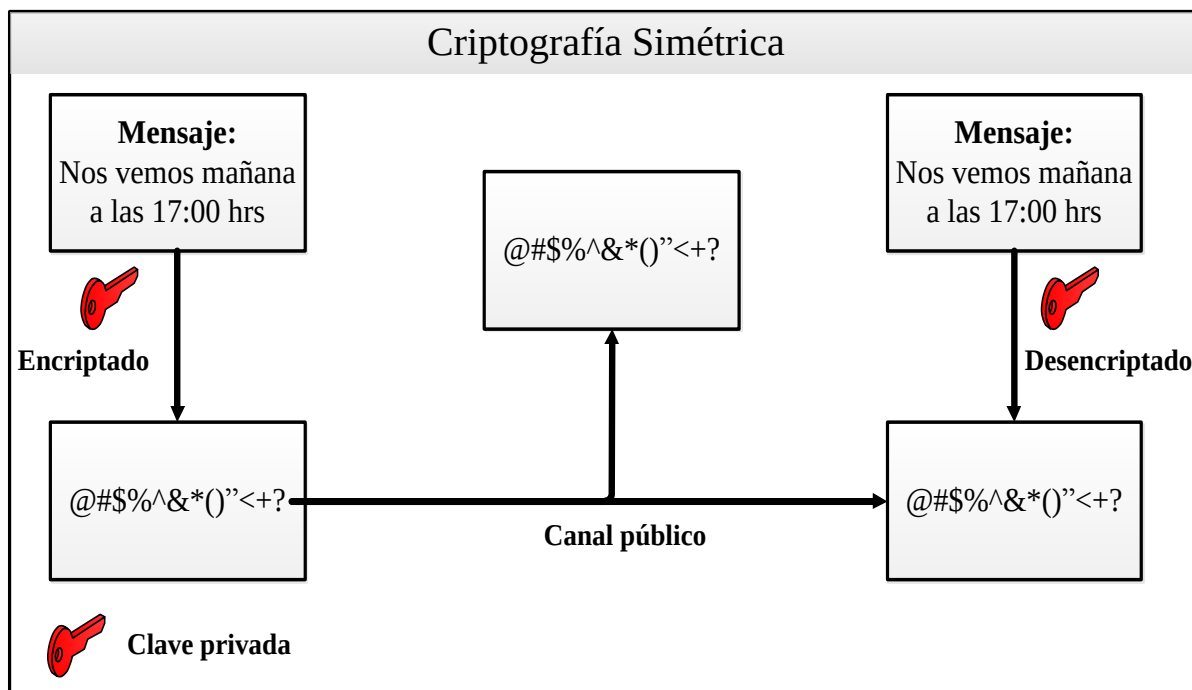


Figura 3.1: Diagrama a bloques del cifrado de información utilizando criptografía de clave simétrica.

Criptografía asimétrica. En este algoritmo de encriptado se emplean *dos tipos de claves, una la clave pública y la otra clave privada*, (ver figura 3.2) ambas pertenecientes al destinatario (Bob), para comunicarse secretamente sólo se requiere proporcionar la clave pública al remitente (Alicia) para cifrar la información, pero una vez siendo encriptado, el remitente ya no podrá descifrar la información, solamente el destinatario con su clave privada. Una de las desventajas que presenta este tipo de cifrado es que la información cifrada ocupa mayor espacio que la información original, también la clave que se emplea es de mayor tamaño en comparación con la clave en el encriptado simétrico.

Algunos algoritmos de clave asimétrica son: Diffie-Hellman, RSA, DSA, ElGamal, criptografía de curva elíptica, Merkle-Hellman, etc.

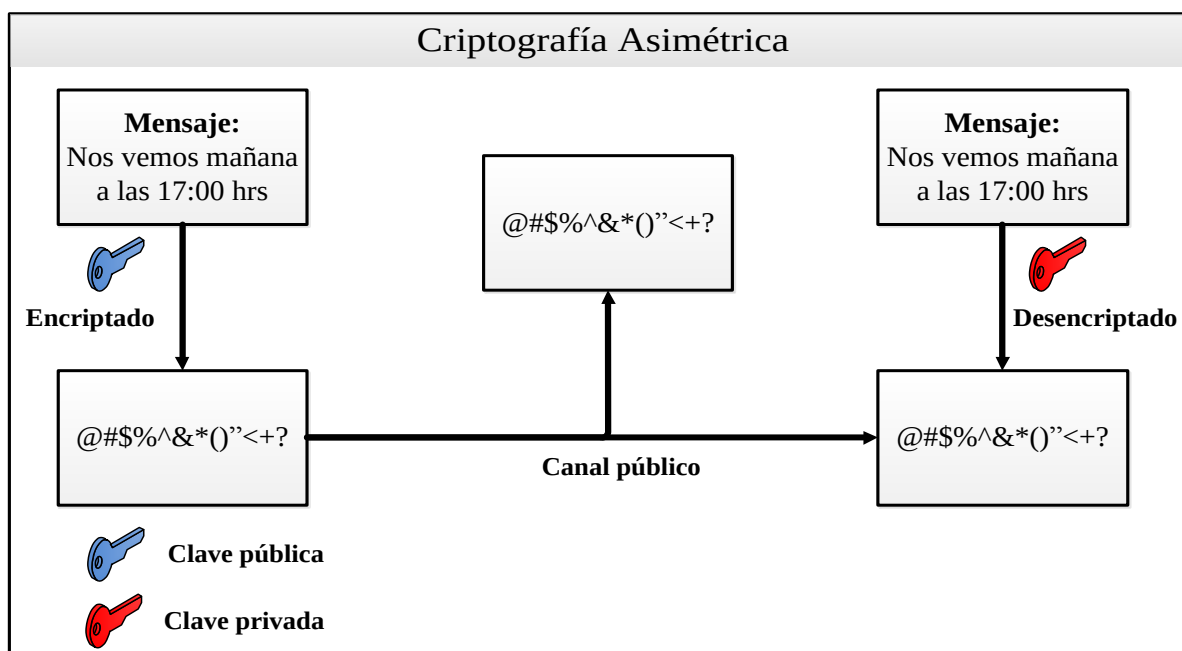


Figura 3.2: Diagrama a bloques del cifrado de información utilizando criptografía de clave asimétrica.

Nota: En este trabajo de tesis, el tipo de encriptado que se pretende implementar será de tipo no convencional, llamado *caótico*, el cual, se encuentra en la clasificación de criptografía simétrica, esto debido a que se requiere que sea el mismo sistema caótico en el receptor como en el transmisor y empleando la misma clave Alicia y Bob.

3.5. Algoritmos criptográficos

Existen diversos algoritmos criptográficos en la actualidad por lo que se podrían mencionar varios de ellos, sin embargo a continuación se proporcionará información sólo de algunos de los algoritmos más conocidos.

3.5.1. Algoritmo Blowfish

El algoritmo ***Blowfish*** fue desarrollado en 1993 por Bruce Schneier, quien se inspiró por la necesidad de un nuevo algoritmo estándar de encriptado en las comunicaciones, después de observar lo vulnerable que se encuentra el algoritmo DES ante los ataques de fuerza bruta, criptoanálisis diferencial, criptoanálisis lineal, entre otros. Siendo esto un indicador de que, el algoritmo DES está llegando al final de su vida útil. Pensando en ello Bruce Schneier, decidió desarrollar una alternativa rápida de algoritmo de encriptado que fuera de dominio público en comparación con los existentes, los cuales, están protegidos por una patente, pertenecen a la propiedad privada o su desarrollo está protegido por algún gobierno.

Blowfish es un cifrado en bloques de 64 bits de clave de longitud variable, su algoritmo consta de dos partes principales: la parte de expansión de clave y la parte de encriptado de datos. En la expansión de clave, la clave se convierte en varios arreglos de subclaves aumentando su tamaño; por ejemplo, una clave de 448 bits se convierte en una clave de 4168 bytes.

El cifrado de datos se produce a través de una red de Feistel 16-redonda. Cada ronda consiste de una permutación de llave dependiente, así como también de una sustitución de llave y clave dependientes. Todas las operaciones son realizadas en compuertas XORs y adiciones de palabras de 32 bits. Las únicas operaciones adicionales son cuatro arreglos de búsquedas de datos indexados por cada ronda. Para mayor información sobre su funcionamiento, implementación y uso, consultar por ejemplo las referencias [SCH93], [SW10], [GRGM08] y [SCH10].

3.5.2. Algoritmo AES

El algoritmo **AES** fue diseñado por Joan Daemen y Vincent Rijmen. Este algoritmo fue presentado para su estandarización por NIST (por sus siglas en inglés, National Institute of Standards and Technology), el 26 de noviembre de 2001 como estándar de encriptado avanzado (AES, Advanced Encryption Standard), pero no fue aceptado como un estándar efectivo hasta el 26 de mayo de 2002. El AES sigue la estrategia de diseño wide-trail y consiste en una lista de llaves, así como también de un estado de actualización de transformación. El algoritmo AES es capaz de utilizar claves de cifrado de 128, 192 y 256 bits para cifrar o descifrar datos en bloques de 128 bits. Dependiendo de la longitud de la clave del algoritmo se puede referenciar como AES-128, AES-192 o AES-256.

Cada una de las entradas y salidas del algoritmo AES se compone de secuencias de 128 bits. Internamente, las operaciones del algoritmo AES se realizan en un arreglo bidimensional (Matriz) de bytes llamado *Estado* (ver figura 3.3). El estado está compuesto de cuatro filas de bytes, cada una con Nb bytes, donde Nb es la longitud del bloque dividido entre 32. En la matriz de estado denotada por el símbolo s , cada byte individual tiene dos subíndices, con la letra r se indica el número de filas (rango: $0 < r < 4$) y con la letra c se indica el número de columnas (rango: $0 < c < Nb$). Por tanto, cada uno de los bytes de la matriz de estado es representado como $s_{r,c}$ o $s[r, c]$. Para mayor información consultar [MPRS09], [DR01b] y [DR01a].

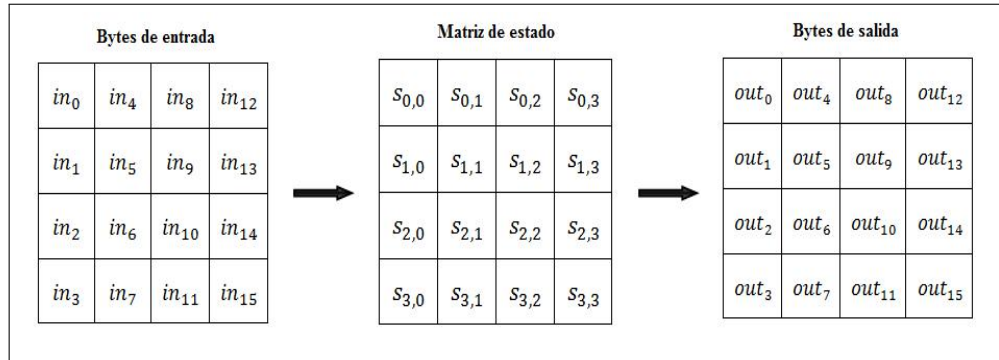


Figura 3.3: Matriz de estado, bytes de entrada y bytes de salida.

3.5.3. Algoritmo IDEA

El algoritmo **IDEA** fue desarrollado originalmente como una modificación del PES (por sus siglas en inglés, Proposed Encryption Standard), el algoritmo PES fue publicado en 1990 por Lai y Massy, quienes desarrollaron el algoritmo como una alternativa de encriptado que remplazara al encriptado estándar de datos (DES). El algoritmo PES fue modificado y publicado en 1991 por Lai, Massey y Murphy, originalmente el nuevo algoritmo obtenido fue nombrado como IPES (por sus siglas en inglés, Improved Proposed Encryption Standard), posteriormente en 1992 su nombre cambio a **IDEA** (por sus siglas en ingles International Data Encryption Algorithm).

El algoritmo IDEA es un cifrado iterativo que consiste de 8 rondas similares, con una simple transformación de salida. Las partes que componen las funciones de ronda son una multiplicación modulo $2^{16} + 1$, una suma modulo 2^{16} y una operación XOR a nivel de bit. El algoritmo IDEA consta de una clave de 128 bits y cifra/descifra los datos en bloques de 64 bits. El proceso computacional de encriptado del algoritmo IDEA se muestra en la figura 3.4. Para mayor información consultar por ejemplo las referencias [DGV93] y [HOF07].

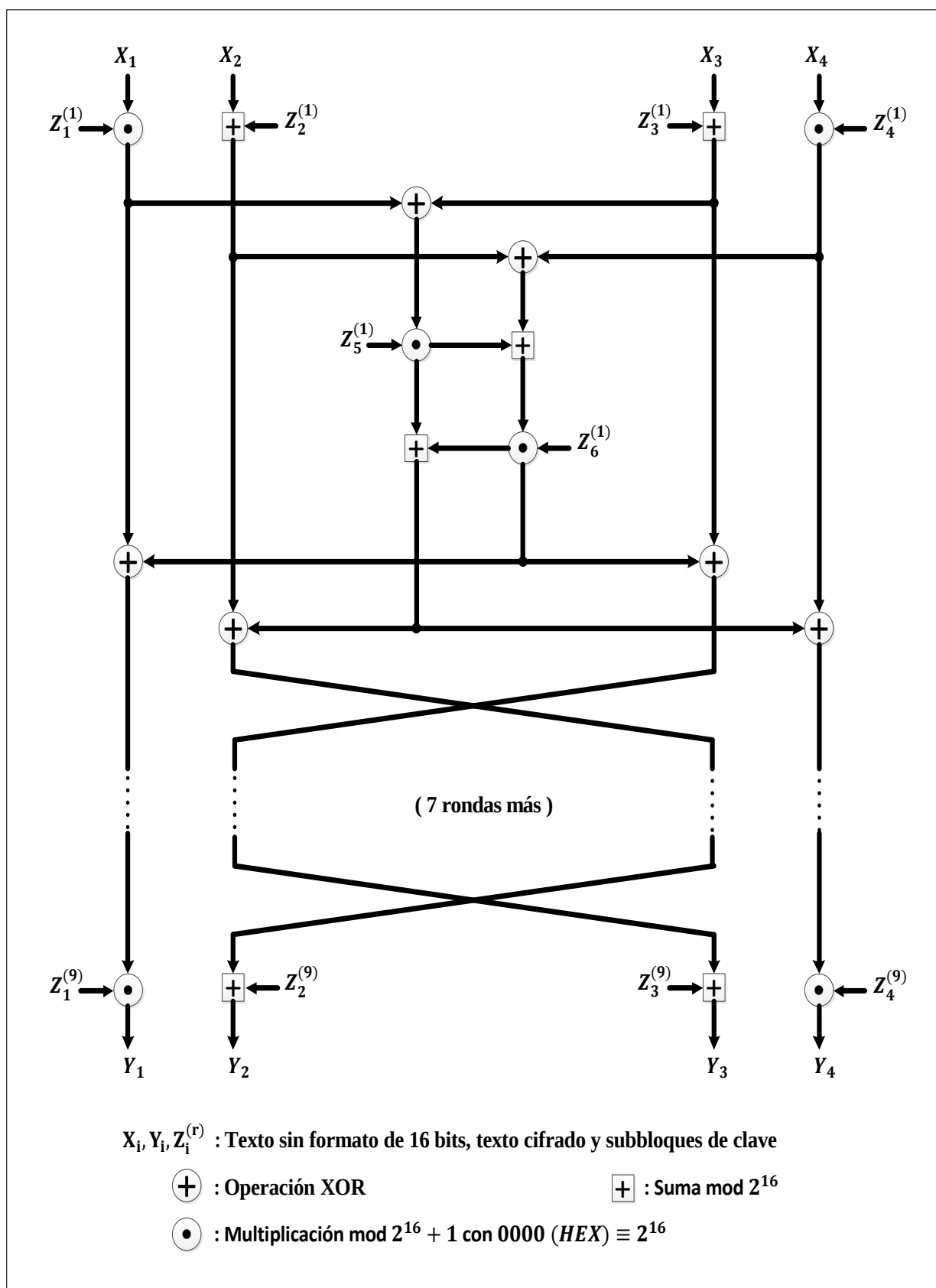


Figura 3.4: Proceso de encriptado del algoritmo IDEA.

3.5.4. Algoritmo DES

El Algoritmo **DES** (por sus siglas en inglés, Data Encryption Standard), fue desarrollado originalmente por IBM, siendo presentado el algoritmo como una propuesta de encriptado, en el concurso realizado por NBS (por sus siglas en inglés, National Bureau of Standards), donde solicitaba un algoritmo de encriptado para la protección de datos de computadora durante su transmisión y almacenaje.

En 1974 la corporación IBM, presentó la propuesta del algoritmo DES, siendo éste una modificación de su sistema LUCIFER, después en 1977 el gobierno de Estados Unidos modificó y adoptó el algoritmo como estándar de cifrado de todas las informaciones sensibles no clasificadas. Posteriormente, en 1980, el NIST (por sus siglas en inglés, National Institute of Standards and Technology) estandarizó los diferentes modos de operación del algoritmo.

El DES es un algoritmo de cifrado en bloque, su longitud de bloques es de 64 bits y su longitud de clave es de 56 bits, lo que equivale a que existan $2^{56} = 7,2 \times 10^{16}$ claves diferentes. El algoritmo DES trabaja alternativamente sobre las dos mitades del bloque a cifrar.

Lo primero que realiza el algoritmo es una permutación inicial fija, la cual, no tiene ningún valor criptográfico, posteriormente se divide el bloque en dos partes, la derecha y la izquierda. Después se realiza una operación modular que se repite 16 veces, esta operación consiste en una suma módulo 2 de la parte izquierda con una función $F(K_i)$ de la parte derecha, gobernada por una clave K_i . posteriormente se intercambian las partes derecha e izquierda, en la vuelta 16 se omite el intercambio, pero al final se realiza una permutación, siendo esta la inversa de la inicial.

En la figura 3.5 se puede observar el inicio y primera vuelta del proceso del algoritmo DES. Si el lector requiere mayor información de su funcionamiento e implementación puede consultar las referencias [SM+01], [MOV00], [FIPSP99] y [SA99].

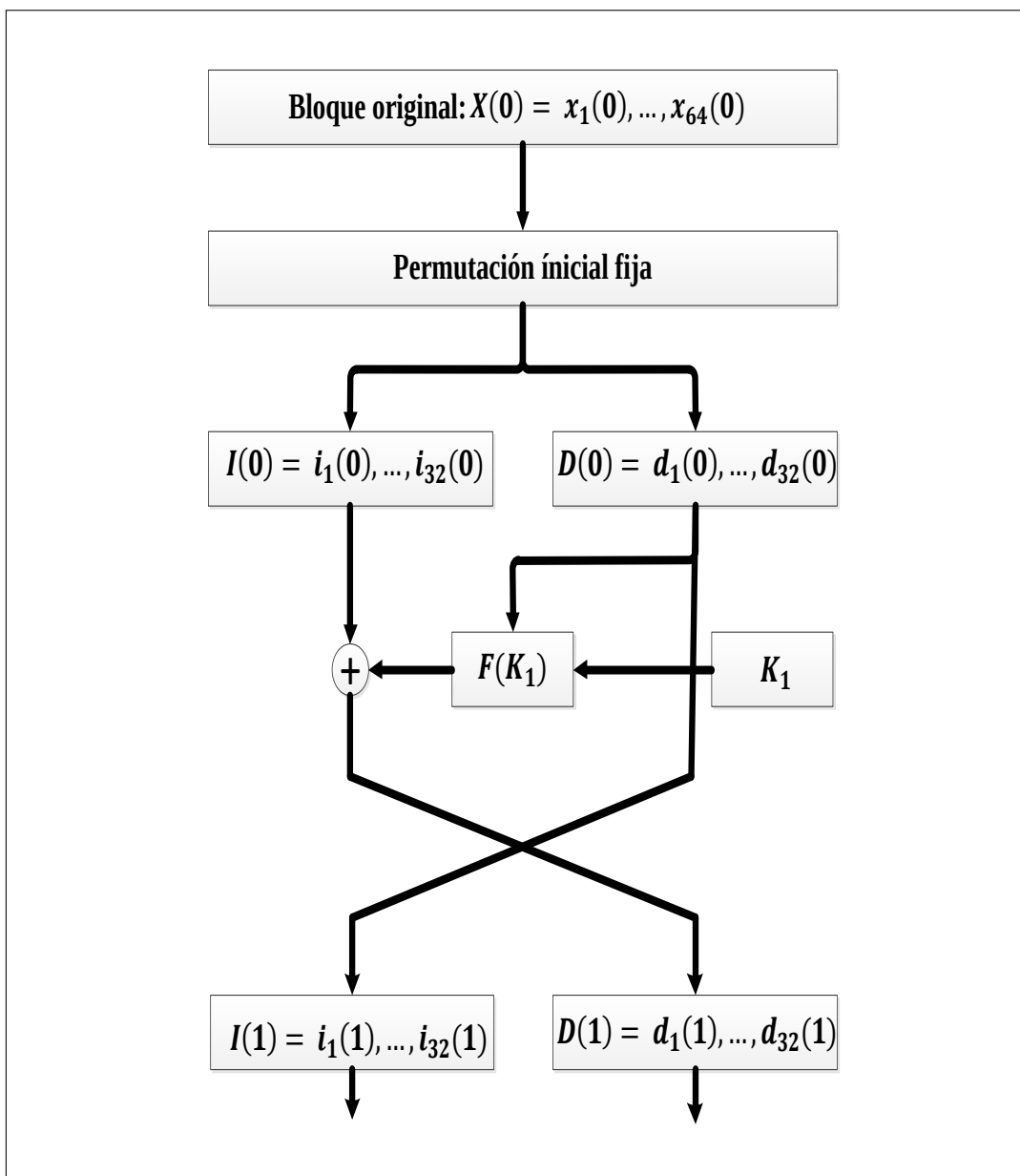


Figura 3.5: Inicio y primera vuelta del proceso de encriptado del algoritmo DES.

3.5.5. Algoritmo RSA

El sistema criptográfico **RSA** fue desarrollado en 1977 por Ronald Rivest, Adi Shamir y Leonard Adleman. Este algoritmo utiliza dos claves una pública y otra privada, el sistema es robusto cuando se utilizan claves de 2, 048 bits. La seguridad del sistema criptográfico RSA está basada en la complicación que existe en encontrar la clave pública a partir de la clave privada, es decir, la factorización del módulo n en sus dos factores primos. El protocolo desarrollado por Rivest, Shamir y Adleman, el cual, se conoce como RSA y a continuación se describe brevemente:

1. Primero se generan aleatoriamente dos números primos p y q , cada uno aproximadamente del mismo tamaño, actualmente se recomienda que estos números, contengan más de 200 dígitos. Con los números generados se calcula $n = p \cdot q$. El grupo a utilizar por el usuario es entonces, $\mathbb{Z}_n^*$. El orden de este grupo es $\phi(n) = \phi(p \cdot q) = (p-1)(q-1)$.
2. Después se selecciona aleatoriamente un entero positivo e , $1 \leq e < \phi(n)$, de modo que sea primo con el orden del grupo, es decir $\text{mod } (e, \phi(n)) = 1$.
3. Por medio del algoritmo extendido de Euclides, se calcula el inverso de e en $\mathbb{Z}_{\phi(n)}, d$; se tiene entonces $e \cdot d \equiv 1 \pmod{\phi(n)}$, con $1 \leq d < \phi(n)$.
4. La clave pública del usuario es la pareja (n, e) , mientras que la clave privada será d . También se deben mantener en secreto los números p , q y $\phi(n)$.

Por ejemplo, supóngase que un usuario **A** (Alicia) quiere enviar un mensaje cifrado al usuario **B** (Bob), en este caso, el usuario **A** utilizará la clave pública del usuario **B** (n_b, e_b) para cifrar la información, es decir calcula $c = m^{e_b} \pmod{n_b}$, siendo c el mensaje cifrado que se enviará a **B**. Para que **B** descifre el mensaje tendrá que calcular $c^{d_b} = (m^{e_b})^{d_b} = m^{e_b d_b} \equiv m \pmod{n_b}$. En el sistema criptográfico los números e , d y n , son llamados exponente de encriptado, exponente de descifrado y módulo, respectivamente. Para mayor información consultar por ejemplo las referencias [SM+01], [MOV00], [AB08] y [HW03].

3.5.6. Algoritmo DSA

El DSA (por sus siglas en inglés, Digital Signature Algorithm) es una firma digital propuesta por el Instituto Nacional de Estándares y de Tecnología de los Estados Unidos (NIST, por sus siglas en inglés, National Institute of Standard and Technology). En agosto de 1991, el NIST propuso un estándar para la firma digital (ver referencia [FIPSP09]), llamado Digital Signature Standard (DSS) y su algoritmo correspondiente el DSA. El objetivo de la propuesta era proporcionar una firma digital estándar para las comunicaciones entre oficinas gubernamentales de los Estados Unidos. El algoritmo DSA es una variante del esquema ElGamal, el protocolo del algoritmo DSA es el siguiente:

Generación de claves. Cada entidad **A** debe crear una clave pública y su correspondiente clave privada, para ello debe seguir los siguientes pasos:

1. Seleccionar un número primo q tal que $2^{159} < q < 2^{160}$.
2. Seleccionar un número primo p tal que $2^{511+64t} < p < 2^{512+64t}$, donde t se debe elegir de modo que $0 \leq t \leq 8$ y q debe tener la propiedad de ser un divisor primo de $(p-1)$.
3. Seleccionar un generador α del único subgrupo cíclico de $\mathbb{Z}_p^*$ de orden q .

- a) Seleccionar un elemento $g \in \mathbb{Z}_p^*$ y calcular $\alpha = g^{(p-1)/q} \bmod p$.
- b) Si $\alpha = 1$, entonces regresar al paso a).
- 4. Seleccionar aleatoriamente un entero a tal que $1 \leq a \leq q - 1$.
- 5. Calcular $y = \alpha^a \bmod p$.
- 6. Las llaves creadas por la entidad **A** son: la llave pública es (p, q, α, y) y la llave privada es a .

Generación y verificación de firma Cualquier entidad **B** que pretenda verificar la firma del mensaje m enviado por la entidad **A** deberá utilizar la clave pública proporcionada por la entidad **A**.

1. *Generación de la firma.* Para que la entidad **A** firme un mensaje m deberá seguir los siguientes pasos:
 - a) Seleccionar aleatoriamente un número entero, secreto k , tal que, $0 < k < q$.
 - b) Calcular $r = (\alpha_k \bmod p) \bmod q$.
 - c) Calcular $k^{-1} \bmod q$.
 - d) Calcular $s = k^{-1}\{h(m) + ar\} \bmod q$.
 - e) Una firma para el mensaje m es el par (r, s) .
2. *Verificación de firma.* Para que la entidad **B** verifique la firma (r, s) del mensaje enviado por la entidad **A**, deberá seguir los siguientes pasos:
 - a) Obtener una clave pública auténtica (p, q, α, y) .
 - b) Verificar que $0 < r < q$ y $0 < s < q$. Si no, entonces rechazar la firma.
 - c) Calcular $w = s^{-1} \bmod q$ y $h(m)$.
 - d) Calcular $u_1 = w \cdot h(m) \bmod q$ y $u_2 = rw \bmod q$.
 - e) Calcular $v = (\alpha^{u_1} y^{u_2} \bmod p) \bmod q$.
 - f) La firma es aceptada si y sólo si $v = r$.

Si el lector desea mayor información puede consultar las referencias [MOV00], [SM+01] y [FIPSP09].

3.6. Alternativas de encriptado

Las técnicas de encriptado mencionadas son algoritmos convencionales, en la actualidad se están estudiando algoritmos no convencionales para cifrar la información. Como son la *criptografía cuántica* y la *criptografía caótica*, que a continuación brevemente se describen.

Criptografía cuántica. Se basa en el principio de incertidumbre de Heisenberg, el cual dice que no se pueden conocer dos propiedades distintas de una partícula subatómica en un mismo instante de tiempo.

Criptografía caótica. Se basa en ecuaciones diferenciales o en diferencias no lineales que presentan comportamientos caóticos haciendo de estos un beneficio para poder ocultar la información y confundiéndolo con un aparente ruido.

3.7. Aplicaciones de criptografía

Actualmente la aplicación de criptografía se podría clasificar en dos casos; el primero caso, es para proteger la información en su sitio de almacenamiento, por ejemplo, en una computadora se puede cifra un documento Word para que ninguna persona tenga acceso a la información del documento durante su almacenamiento, con excepción de la persona que conoce la clave. El segundo, es para proteger la información durante su envío en la línea de comunicación de una red de transmisión, por ejemplo, las comunicaciones del ejército durante una guerra son cifradas con el propósito de que el contrario no se entere de la información transmitida en la comunicación.

Sin embargo, la protección de la información no es lo único que proporciona el cifrar la información. Al aplicar un cifrado nos proporciona algunos servicios de seguridad como son:

Confidencialidad. Este servicio se refiere, que sólo la persona autorizada pueden acceder a la información transmitida o almacenada, lo cual, indica que la información se encuentra reservada sólo a las personas autorizadas. Por ejemplo, si la comunicación de dos individuos por medio de un teléfono se encripta, ambos tendrán la seguridad, que su conversación es confidencial.

Integridad. Este servicio se refiere, que la persona no autorizada, no tendrá la facultad de alterar o modificar la información previamente encriptada, durante su almacenamiento o transmisión. Por ejemplo, si un mensaje es encriptado y enviado, el destinatario tendrá la plena seguridad que el mensaje llegó sin ninguna modificación.

Autenticidad. Este servicio se refiere, que el origen de la información transmitida puede identificarse, para asegurarse que el mensaje fue originalmente enviado por el transmisor

conocido y no por un extraño. Por ejemplo, actualmente la autenticidad de un correo electrónico puede verificarse por medio de firmas digitales que es un tipo de cifrado que se utiliza para verificar la autenticidad del correo.

No rechazo. Este servicio se refiere a que los mensajes enviados o recibidos por un usuario quedaran registrados, para que en un futuro no se pueda negar que el mensaje fue enviado o recibido. Por ejemplo, si un cliente realiza una compra a una tienda por medio del correo electrónico, pero la tienda desea demostrar en el futuro que el cliente realmente realizó el pedido y comprobar que no creó el e-mail, con el motivo de asegurarse que el cliente no pueda negar la compra. El no rechazo quita la capacidad del usuario para negar la responsabilidad de sus mensajes.

Control de accesos. Este servicio se refiere, que sólo los usuarios autorizados, debidamente identificados pueden obtener permiso de acceso. Por ejemplo, para realizar operaciones bancarias vía internet primero se debe accederse a la página oficial del banco, la cual se encuentra cifrada y después se debe ingresar la cuenta de usuario, para acceder al servicio.

Si el lector requiere mayor información puede consultar las referencias [SM+01], [AB08], [MOV00] y [SC05].

3.8. Conclusión

En la actualidad, la utilización de algoritmos criptográficos para proporcionar seguridad en las líneas de comunicación tiene una gran importancia, debido a esto existen diversos métodos de encriptado. Hoy en día se realizan diversos estudios e investigaciones, en los cuales, se presentan alternativas nuevas de encriptado o mejoras de algoritmos ya existentes, todo esto con el objetivo de contribuir a resolver el problema de la seguridad de la información en la transmisión o en el almacenamiento en la base de datos. Tomando en consideración lo mencionado, se puede decir que el encriptado propuesto en este trabajo de tesis puede contribuir en la seguridad de algunas de las líneas de comunicación de un sistema biométrico y es una contribución más en las diversas aplicaciones que puede tener el encriptado caótico.

Capítulo 4

Caos

4.1. Introducción

Cuando se menciona la palabra **caos**, lo primero que se nos viene a la mente es desorden, algo que es muy confuso, debido a esto, cuando surge en una conversación, crea mucha curiosidad y controversia entre las personas que están dialogando. De hecho, su significado puede ser distinto para los matemáticos, físicos, filósofos, médicos, economistas, ingenieros, etc. La palabra **caos** puede tener un significado distinto para cada una de las personas, esto puede deberse a su campo de estudio. Sin embargo; ¿qué significado tiene la palabra **caos** en este trabajo de tesis realmente?, debido a esta pregunta se decidió escribir este capítulo, para mencionar algunos aspectos de importancia respecto a este concepto, el capítulo se divide en las siguientes partes: ¿qué es el caos?, antecedentes históricos, ejemplos de sistemas caóticos, aplicaciones del caos y una breve conclusión.

4.2. ¿Qué es el caos?

¿Qué es el caos?, para contestar esta pregunta, primero se dará la definición que proporciona el *Diccionario de la Lengua Española* (Real Academia Española, ver referencia [ESP92]) esta definición es la siguiente:

“Del latín **Chaos** y este del griego $\chi\alpha\omicron\varsigma$, abertura. Estado amorfo e indefinido que se supone anterior a la constitución del cosmos. También, admitido como confusión, desorden”.

Sin embargo, como se mencionó esta palabra puede tener diferente significado según sea el campo de estudio. Por tanto, se decidió buscar una definición más apegada a los campos de la física y matemáticas. En la tesis “El caos y la caracterización de series de tiempo a través de técnicas de la dinámica no lineal” [EE04], se encuentran las siguientes definiciones:

“**Caos**. En mecánica y matemáticas, es un comportamiento de algunos sistemas dinámicos gobernados por leyes determinísticas, el cual, manifiesta un comportamiento aparentemente azaroso y resulta impredecible a largo tiempo. Por tanto, un nombre más preciso sería

caos determinístico ...”

“**Caos.** Comportamiento estocástico que ocurre en un sistema determinista.”

Como puede verse en las dos últimas definiciones, se menciona que el **caos** tiene un comportamiento estocástico, sin embargo se presenta en un sistema determinista, estas dos cuestiones ha provocado mucha polémica entre los físicos y matemáticos, debido a que en un sistema determinista su comportamiento puede predecirse y repetirse, bajo las mismas condiciones, mientras que en un sistema estocástico esto no es posible (véase la referencia [FS07]).

En esta tesis, cuando se mencione sistema caótico, se refiere a que el sistema tiene un comportamiento complejo, que aparenta un total desorden, sin embargo este comportamiento se puede predecir bajo ciertas condiciones y en periodos breves de tiempo.

4.3. Algunos antecedentes históricos

Un hecho de importancia con respecto al concepto de caos, comenzó cuando los científicos se hicieron la siguiente pregunta: ¿si el sistema solar era estable?. Debido a esto, en 1887 el **Rey Oscar** II de Suecia ofreció un premio a quien pudiera determinar si el sistema solar era estable, para poder contestar a esta interrogante primero había que resolver el problema de los tres cuerpos, desafortunadamente nadie pudo hacerlo. Sin embargo, el ganador fue **Jules Henri Poincaré** por sus aportaciones a los estudios de la dinámica de los sistemas complejos, siendo este investigador uno de los primeros en percatarse del fenómeno que ahora llamamos **caos**. Poincaré publicó un trabajo en el que concluyó que el problema de los tres cuerpos no tiene una solución general mediante las ecuaciones de la mecánica clásica y que un sistema de tres cuerpos es extremadamente dependiente de sus condiciones iniciales. Sin embargo, el trabajo de Poincaré contenía errores, debido a esto, su edición fue sacada de circulación. Las contribuciones que escribió Poincaré referente al tema de caos, fueron olvidados con el pasar del tiempo.

Con la llegada de las súper computadoras también llegó un redescubrimiento del **caos**, el cual, fue hecho por el meteorólogo y matemático estadounidense **Edward N. Lorenz**, considerado por muchos como el descubridor del caos. En 1963 Edward N. Lorenz comenzó a trabajar en unas ecuaciones diferenciales que permitieran predecir el clima. Sin embargo, las ecuaciones que describían el comportamiento del clima requerían el cálculo de miles de operaciones debido a esto, Lorenz se dio a la tarea de simplificar al máximo su sistema de ecuaciones. Esto lo realizó descartando variables que consideró de menor relevancia en el sistema de ecuaciones para predecir el clima, al final, su sistema de ecuaciones consideraba solamente tres variables, **la dirección de los vientos, la presión del aire y la temperatura**, actualmente a este sistema de ecuaciones se les conoce como **ecuaciones de Lorenz**.

Al terminar de plantear sus ecuaciones, Lorenz se auxilió de las computadoras para obser-

var el comportamiento dinámico de su sistema de ecuaciones. Después, el pretendió verificar unos resultados obtenidos durante estas pruebas en computadora pero para ahorrarse tiempo decidió comenzar a la mitad de una de sus pruebas, así que tomó una de las hojas de resultados que había obtenido anteriormente e introdujo estos resultados a la computadora. Debido a que las computadoras en su tiempo eran muy ruidosas y lentas Lorenz decidió salir de su laboratorio, al cabo de una hora decidió volver a checar los resultados de su prueba, se encontró con resultados totalmente distintos a los que había obtenido anteriormente, desconcertado comenzó a verificar si tenía un error en su programa o equipo de computo, pero no encontró nada. Después, habiendo reflexionado se dio cuenta de que al introducir los datos para ahorrar tiempo, él discriminó algunos dígitos y al hacer ésto, los resultados cambiaron. Debido a esto, pudo observar que con cada iteración de sus ecuaciones, el error se iba magnificando, por consiguiente su sistema de ecuaciones tenía un comportamiento que era *sensible a las condiciones iniciales* y gráficamente la trayectoria de este sistema de ecuaciones presentaba un atractor extraño (ver figura 4.1). Con esto Lorenz concluyó que la predicción del clima a largo plazo estaba destinada al fracaso, debido a que cualquier pequeña alteración en el ambiente puede alterar los resultados.

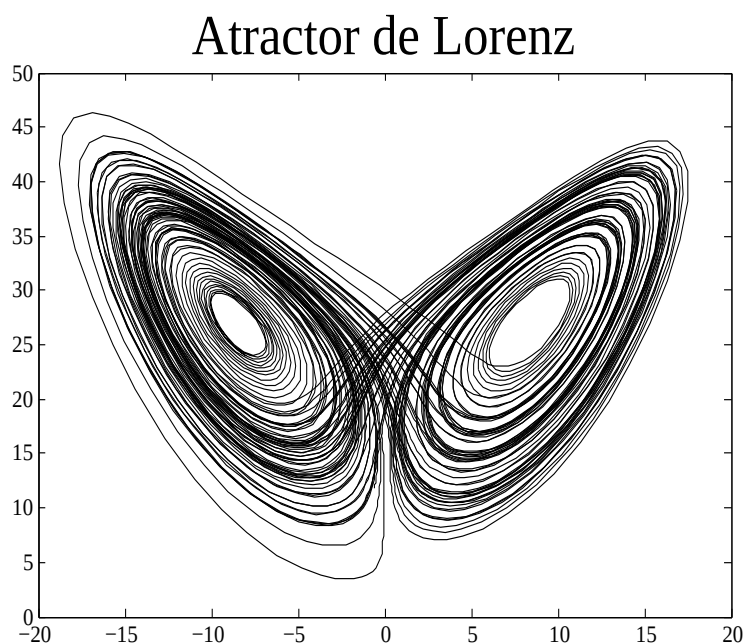


Figura 4.1: Imagen que resulta de graficar una variable con respecto a otra de las ecuaciones de Lorenz conocida como atractor de Lorenz.

En la década de 1970 muchos investigadores interesados en el tema del caos se encontraron con ecuaciones que presentaba un comportamiento caótico, entre las ecuaciones más conocidas están las ecuaciones de Rössler y Hénon. Hoy en día, se sabe que el comportamiento caótico está presente en un gran número de sistemas en ingeniería y en la naturaleza.

4.4. Sistemas caóticos

Dentro de los estudios de los sistemas dinámicos en física y matemáticas, se encuentran varias clasificaciones, una de las más conocidas es la clasificación de que si un sistema es lineal o no, a continuación se menciona esta clasificación:

Sistemas lineales. Un sistema lineal es aquel que está formado por un conjunto de ecuaciones lineales (las cuales, pueden ser diferenciales o discretas) que podrían describir el comportamiento de alguna variable física, estas ecuaciones son un planteamiento de igualdad que involucran operaciones de sumas, restas, etc. De una o más variables a la primera potencia. La mayor características de los sistemas lineales es que cumplen con el principio de proporcionalidad y superposición véase [FS07]. A continuación se muestra un ejemplo de este tipo de sistemas.

$$\begin{aligned} 3\frac{dx_1}{dt} + 2\frac{dx_2}{dt} + \frac{dx_3}{dt} &= 1 \\ 2\frac{dx_1}{dt} + 2\frac{dx_2}{dt} + 4\frac{dx_3}{dt} &= -2 \\ -\frac{dx_1}{dt} + \frac{dx_2}{dt} - \frac{dx_3}{dt} &= 0 \end{aligned}$$

Sistemas no lineales. Un sistema no lineal es aquel que está formado por un conjunto de ecuaciones no lineales (los cuales, pueden ser diferenciales o discretas) que podrían describir el comportamiento dinámico de un variables física, estas ecuaciones son un planteamiento de igualdad que involucran operaciones de suma, resta, multiplicación, etc. De una o más variables a la segunda potencia o a una potencia mayor. La diferencia más notable con respecto a los sistemas lineales es que estos sistemas no cumplen el principio de proporcionalidad y superposición. A continuación se muestra un ejemplo de este tipo de sistemas.

$$\begin{aligned} 3\frac{d^2x_1^4}{dt} + \frac{d^3x_2^6}{dt} + \frac{dx_3^3}{dt} &= 1 \\ 2\frac{dx_1^3}{dt} + 2\frac{dx_2}{dt}x_1^3 + 4\frac{d^3x_3^5}{dt} &= -4 \\ -\frac{d^3x_1}{dt} + x_3^3\frac{d^2x_2^5}{dt} - \frac{d^2x_3}{dt} &= 5 \end{aligned}$$

La región del espacio de estados hacia la cual convergen todas las trayectorias posibles de un sistema dinámico, se llama **atractor**. El comportamiento dinámico de un sistema no lineal puede ser estable, inestable, periódico o caótico. Si el comportamiento del sistema es estable, sus trayectorias convergen a un punto de equilibrio, si es periódico sus trayectorias convergen a un ciclo límite, ahora si su comportamiento es inestable las trayectorias

del sistema escapa a infinito (es decir, no hay una región de atracción) en otras palabras, no existe atractor. Si el comportamiento del sistema es caótico, este presenta un peculiar comportamiento en sus trayectorias, por un lado el sistema exhibe un atractor por el cual, el sistema se ve atraído, pero a su vez escapa de esta zona de atracción, debido a esto su trayectoria forma una imagen de una geometría complicada y con dimensión fractal, siendo esta la razón por la que el atractor de un sistema caótico se le llama ***atractor extraño***.

Un parámetro que nos indica la dependencia de las condiciones iniciales de un sistema caótico son ***los exponentes de Lyapunov***, los cuales, proporcionan información de la divergencia de las trayectorias de un sistema para condiciones iniciales cercanas. Un exponente de Lyapunov positivo, es necesario para indicar la presencia de caos en un sistema dinámico, debido a esto, los exponentes de Lyapunov positivos son utilizados para determinar la complejidad de un sistema caótico (ver por ejemplo [LG00]). Los exponentes de Lyapunov son proporcionales al orden del sistema dinámico, es decir, para un sistema de orden N , tendrá N exponentes de Lyapunov.

Se podría decir que los sistemas dinámicos que presentan caos reúnen las siguientes propiedades:

- Son sistemas modelados con ecuaciones diferenciales o en diferencias **no lineales**. A continuación se presentan dos ejemplos, uno en tiempo continuo (Sistema de Lorenz) y otro en tiempo discreto (Mapeo de Rössler):

Mapeo de Rössler

$$\begin{cases} x_1(k+1) = \alpha x_1(k)(1 - x_1(k)) - \beta(x_3(k) + \gamma)(1 - 2x_2(k)), \\ x_2(k+1) = \delta x_2(k)(1 - x_2(k)) + \varsigma x_3(k), \\ x_3(k+1) = \eta((x_3(k) + \gamma)(1 - 2x_2(k)) - 1)(1 - \theta x_1(k)), \end{cases} \quad (4.1)$$

Sistema de Lorenz

$$\begin{cases} \frac{dx}{dt} = \sigma(y(t) - x(t)), \\ \frac{dy}{dt} = rx(t) - y(t) - x(t)z(t), \\ \frac{dz}{dt} = x(t)y(t) - bz(t), \end{cases} \quad (4.2)$$

- Presentan **oscilaciones no periódicas**, ver figuras 4.2 y 4.3 donde se ilustran para el mapeo de Rössler y el sistema de Lorenz, respectivamente.

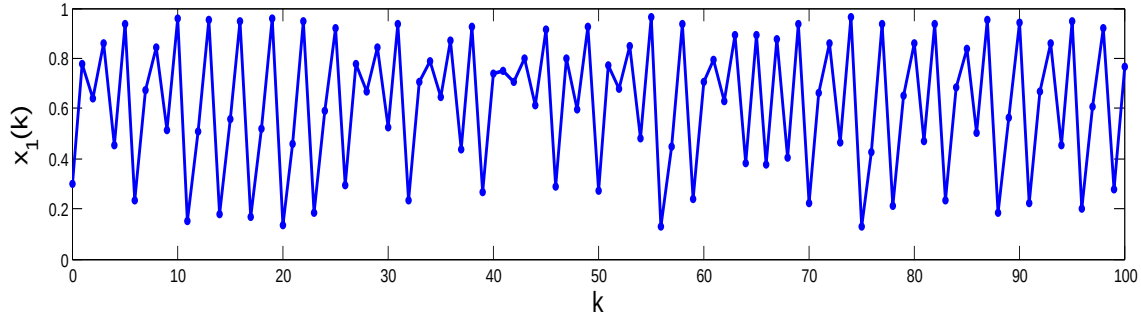


Figura 4.2: Oscilaciones no periódicas de la variable $x_1(k)$ versus el número de iteraciones k de las ecuaciones de Rössler con condiciones iniciales $x(0) = (0.3, 0, 0.05)$ y valores de parámetros $\alpha = 3.8$, $\beta = 0.05$, $\gamma = 0.35$, $\delta = 3.78$, $\varsigma = 0.2$, $\eta = 0.1$, $\theta = 1.9$ y con $k = 100$ iteraciones.

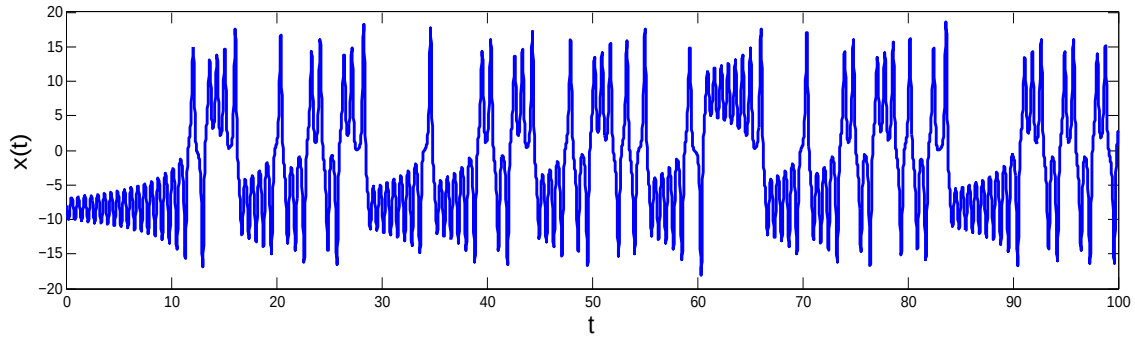


Figura 4.3: Oscilaciones no periódicas de la variable $x(t)$ versus el tiempo t de las ecuaciones de Lorenz con condiciones iniciales $(-8, -8, 24)$ y valores de parámetros $\sigma = 10$, $r = 28$ y $b = \frac{8}{3}$.

- Presentan **sensibilidad a condiciones iniciales**, ver figuras 4.4 y 4.5 para el mapeo de Rössler y el sistema de Lorenz, respectivamente. Nótese la cercanía de las condiciones iniciales que dan origen a dos trayectorias completamente distintas y en un breve lapso de tiempo.

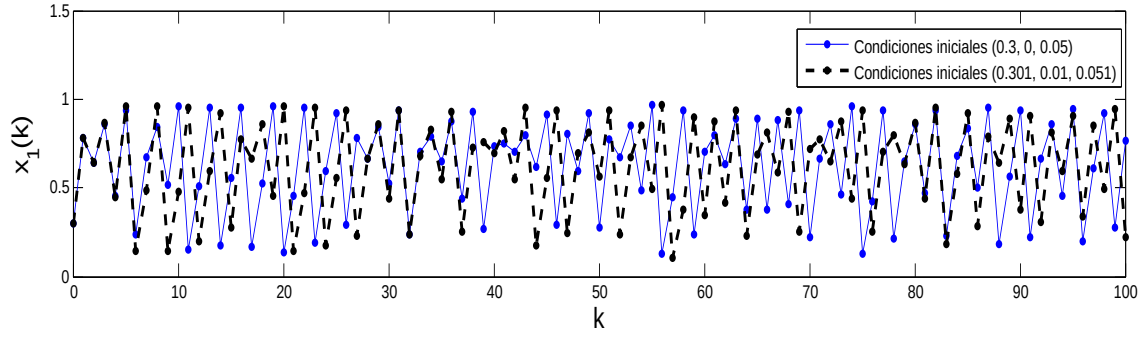


Figura 4.4: Ilustración de la sensibilidad a condiciones iniciales de la variable $x_1(k)$ de las ecuaciones de Rössler con valores de parámetros $\alpha = 3.8$, $\beta = 0.05$, $\gamma = 0.35$, $\delta = 3.78$, $\varsigma = 0.2$, $\eta = 0.1$, $\theta = 1.9$.

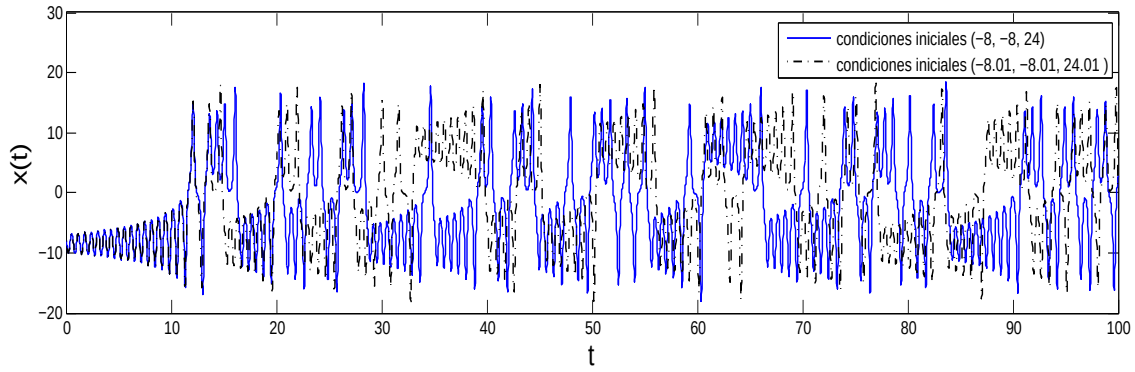


Figura 4.5: Ilustración de la sensibilidad a condiciones iniciales de la variable $x(t)$ de las ecuaciones de Lorenz con valores de parámetros $\sigma = 10$, $r = 28$ y $b = \frac{8}{3}$.

- Generan **atractores extraños** (no son puntos de equilibrio ni ciclos límites), ver figura 4.6.

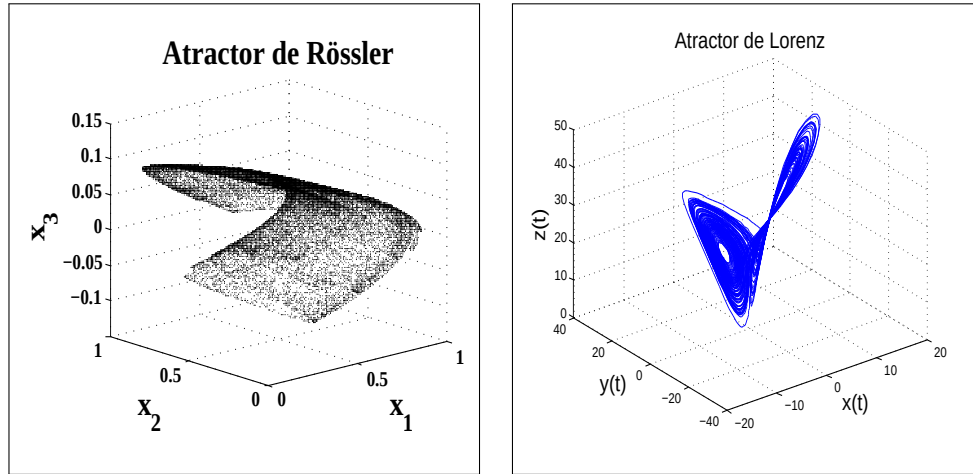


Figura 4.6: Atractores extraños proyectados en el espacio tridimensional.

- Tienen un **exponente de Lyapunov positivo**, los exponentes de Lyapunov para el mapeo de Rössler con los siguientes parametros $\alpha = 3.8$, $\beta = 0.05$, $\gamma = 0.35$, $\delta = 3.78$, $\varsigma = 0.2$, $\eta = 0.1$, $\theta = 1.9$ y condiciones iniciales $(1, 0, 0)$ son: $(\lambda_1=0.313879, \lambda_2=0.019228, \lambda_3=-0.010713)$. Los exponentes de Lyapunov para el sistema de Lorenz con valores de parametros $\sigma = 16$, $r = 40$, $b = 4$ y condiciones iniciales $(0, 1, 0)$ son: $(\lambda_1=1.315952, \lambda_2=0.0, \lambda_3=-22.338674)$. Las expresiones matemáticas para determinar los exponentes de Lyapunov en tiempo discreto y continuo, son por ejemplo como siguen.

$$\left\{ \lambda_i = \frac{1}{KT} \sum_{K=1}^{K-1} \ln \left\| \delta x_i^{(k)} \right\| \quad i = 1, \dots, n \right. \quad (4.3)$$

Ecuación 4.3: Expresión matemática para determinar los exponentes de Lyapunov en tiempo discreto, en donde: λ_i es el i -ésimo exponente de Lyapunov, T es el tiempo entre muestras, x_i es la i -ésima componente de x y δx es la desviación en la trayectoria de dos condiciones iniciales arbitrariamente cercanas. Información obtenida de la referencia [LOCV01].

$$\left\{ \lambda = \frac{1}{t} \ln \frac{d(t)}{d_0} \right. \quad (4.4)$$

Ecuación 4.4: Expresión matemática para determinar el exponente de Lyapunov en tiempo continuo, en donde: λ es el exponente de Lyapunov, t es el tiempo entre muestras, d_0 es la separación en el tiempo $t = 0$ de dos trayectorias caóticas cercanas y $d(t)$ es la separación de las trayectorias debido a la divergencia de las trayectorias en el tiempo. Información obtenida de la referencia [HIL00].

- El atractor tiene una **dimensión fractal**. Por ejemplo, ver figura 4.7 donde se ilustra la generación de una figura fractal, es decir, su dimensión no es un número entero, así que los sistemas caóticos su dimensión es un número fraccional.

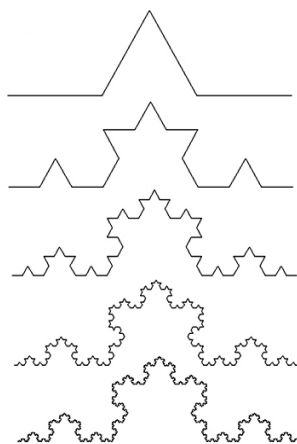


Figura 4.7: Curva de Koch obtenida de la página Web http://labellateoria.blogspot.com/2010_06_01_archive.html

4.5. Aplicaciones del caos

Actualmente la teoría del caos está siendo estudiada en diferentes campos de estudio como son; la física, la biología, la medicina, la filosofía, la música, la economía, la ciencias sociales, la literatura, la meteorología, la ingeniería, entre otras disciplinas. Ejemplos de algunas aplicaciones del caos que se podrían dar en las disciplinas mencionadas serían, en física se podría estudiar la caída de una hoja, en la biología se podría estudiar el comportamiento de crecimiento de una población y su evolución en el tiempo, en la medicina se podría estudiar la frecuencia cardiaca ejemplo de esto se puede ver en la referencia [LOCV01], en la música los tonos que pueden producirse por medio de un sistema caótico, en la economía se podría estudiar el desarrollo económico de un país, para poder obtener una predicción de su comportamiento futuro. En meteorología se puede nombrar el caso más conocido de la teoría del caos, el cual, es el de Lorenz quien trabajó en unas ecuaciones con las que pretendía predecir el clima. Por último, en el área de la ingeniería podremos mencionar el estudio de encriptado realizado en este trabajo de tesis enfocándose en las comunicaciones seguras de un sistema biométrico.

4.6. Conclusión

Las aplicaciones del caos son diversas esto se debe a sus propiedades. En este trabajo de tesis se relaciona con una de sus aplicaciones más importantes las comunicaciones seguras, aun cuando hay diversos trabajos en los cuales se presenta el encriptado caótico para comunicaciones seguras, en la actualidad se encuentran escasos trabajos sobre el encriptado caótico dirigidos a las comunicaciones seguras de un sistema biométrico, por lo que la contribución de este trabajo puede ser sustanciosamente aprovechado a este propósito.

Capítulo 5

Sistema biométrico de control de acceso con base en huellas dactilares

5.1. Introducción

En el segundo capítulo se mencionó lo que son los sistemas biométricos, tomando en consideración este hecho, en esta ocasión el presente capítulo se centrará en los sistemas biométricos que utilizan como identificador biométrico las **huellas dactilares**. Las alternativas que se utilizaron para comparar las huellas, el tipo de encriptado que se presenta como alternativa entre otros aspectos que se tomaron en consideración al momento de realizar el trabajo de tesis, Serán tratados en este capítulo.

5.2. Sistema biométrico de huellas dactilares

A continuación se describirá brevemente algunos aspectos de importancia con respecto al identificador biométrico que se utilizó en este trabajo de tesis, también se describen los diferentes escenarios que se podrían presentar con respecto a la distribución de dispositivos de un sistema biométrico que utiliza como identificador la huella dactilar, después se presenta el tipo de lector que se utilizó para adquirir las imágenes de las huellas dactilares y algunos aspectos que se consideran al adquirir una imagen de la huella dactilar.

5.2.1. Identificador biométrico de huellas dactilares

Entre los diversos identificadores biométricos, en este trabajo de tesis se decidió seleccionar la huella dactilar como característica biométrica, que utilizará el sistema biométrico para realizar el trabajo de identificación. Esto se debe a que la huella dactilar es uno de los identificadores biométricos que cumple satisfactoriamente con los atributos de un identificador biométrico, esto se puede apreciar en la tabla 5.1, estos datos fueron obtenidos de la referencia [JRP04], otra razón sería, que es el identificador biométrico más utilizado en la actualidad.

Identificador Biométrico	Universalidad	Única	Permanente	Cuantificación	Desempeño	Aceptabilidad	Fiabilidad
Huella Dactilar	M	A	A	M	A	M	M
Retina	A	A	M	B	A	B	B
Iris	A	A	A	M	A	B	B
Voz	M	B	B	M	B	A	A
Termografía Facial	A	A	B	A	M	A	B
Geometría de la Mano	M	M	M	A	M	M	M
Firma	B	B	B	A	B	A	A

Tabla 5.1: Comparación entre identificadores biométricos. La información está simbolizada como: Alto (A), Medio (M) o Bajo (B).

5.2.2. Lectura de la huella dactilar

Dependiendo del proceso de adquisición, una imagen de huella dactilar puede clasificarse como “*off-line*” o “*live-scan*”. Una huella clasificada como *off-line* es la que se obtiene al manchar las puntas de los dedos, los cuales, después se plasman sobre un papel, el cual, se escanea para obtener una imagen digital de la huella. Una huella clasificada como *live-scan* es la que se obtiene directamente de un lector de huellas dactilares, el cual, genera una imagen digital de la huella.

Las imágenes utilizadas para los sistemas biométricos basados en huellas dactilares, deben cumplir con algunos *parámetros* que caracterizan a la imagen de la huella, algunos de éstos son: *resolución*, *área*, *número de píxeles*, *contraste* y *distorsión geométrica*. Estos parámetros dependen del dispositivo de captura, la imagen puede tener un rango del valor de sus parámetros. Comúnmente, los píxeles son de valores de 8 bits, los cuales, dan un rango de intensidad de 0 a 255. La resolución de la imagen es el número de píxeles por unidad de longitud, en las imágenes de huellas dactilares su rango es de 250 puntos por pulgada (100 puntos por centímetro) a 625 puntos por pulgada (250 puntos por centímetro). Siendo un estándar común la resolución de 500 puntos por pulgada (200 puntos por centímetro). El área de la imagen es de 0.5 pulgadas cuadradas (1.27 centímetros cuadrados) a 1.25 pulgadas cuadradas (3.175 centímetros cuadrados), siendo comúnmente el estándar de 1 pulgada cuadrada (2.54 centímetros cuadrados). Para mayor información consultar las referencias [JBP99] y [MMJP03].

Nota: El lector que se utilizó para adquirir las imágenes de las huellas dactilares en este trabajo de tesis, fue el lector de huellas de la compañía Microsoft modelo DG2-00004, lector de 8 bits que genera una imagen de dimensión de 355×390 píxeles, ver figura 5.1.



Figura 5.1: Lector de huellas dactilares Microsoft DG2-00004.

5.2.3. Escenario de distribución del sistema biométrico

Los sistemas biométricos físicamente están compuestos principalmente por tres componentes básicos. El primero se encarga de adquirir analógica o digitalmente algún identificador biométrico de una persona, en nuestro caso es el lector de huellas dactilares. El segundo maneja la compresión, el procesamiento, almacenamiento (para la etapa de registro del usuario) y comparación de los datos almacenados en la base de datos (para la etapa de verificación ó identificación), en nuestro caso este componente será una PC, que contiene un programa implementado en MatLab para realizar estas funciones. El tercer componente establece una interfaz con aplicaciones ubicadas en el mismo u otro sistema, en nuestro caso, no contamos con este componente. A continuación en la figura 5.2 se muestra un diagrama a bloques de los tres componentes básicos mencionados.

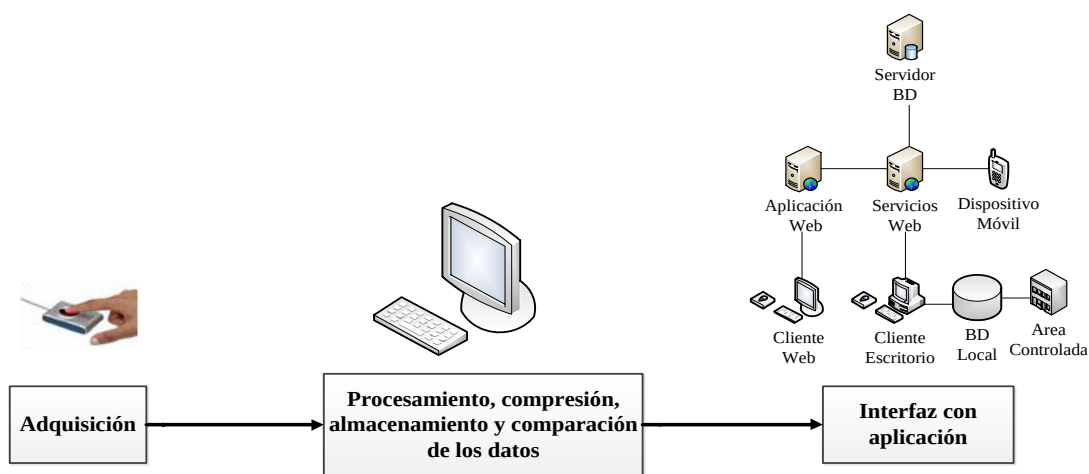


Figura 5.2: Diagrama a bloques de componentes físicos básicos de un sistema biométrico con huella dactilar.

A continuación se describen brevemente algunos equipos comerciales utilizados en *los sistemas biométricos para el control de acceso ó chequeo de personal*, los cuales, realizan el proceso de verificación o identificación. La información de estos dispositivos se obtuvo de la página Web donde se encuentran a la venta. Se presentará un esquema a bloques de la posible distribución, esto se debe a que no se recibió respuesta de la página a la cual, se solicitaron los diagramas originales, como consecuencia, se tuvo que interpretar el escrito para representar los diagramas de distribución.

Dispositivo Viridi 300

El primer equipo es el **Viridi 300**, el cual, es un sistema sumamente confiable y permite una gran versatilidad de funciones, por ejemplo: *control de horario, asistencia y permitir el control de acceso a áreas restringidas al personal autorizado*. Algunas de sus aplicaciones son en edificios inteligentes como sistemas de control de acceso, ya que cuenta con un módulo de “relay” que activa una señal que permite la apertura de una puerta, encender un equipo electrónico, activar una barrera de control, etc. Otra de sus aplicaciones es en fábricas como sistema de control de horario y asistencia del personal, para más detalles consultar las referencias [SCB09] y [SEC09]. A continuación en la figura 5.3 se muestra un diagrama a bloques del sistema biométrico para este dispositivo.

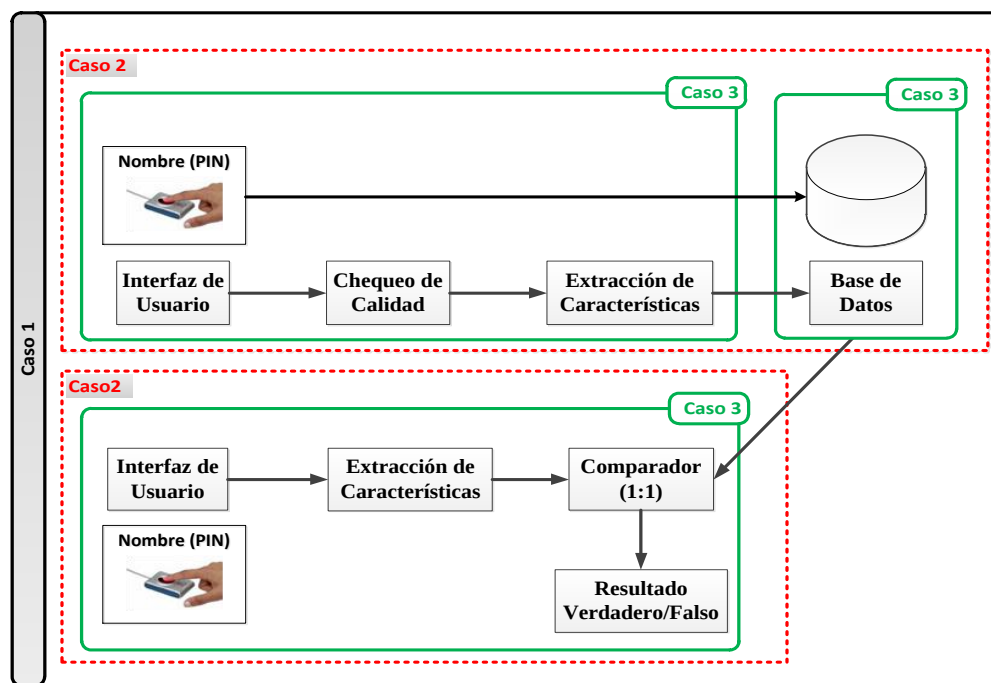


Figura 5.3: Diagrama a bloques de un sistema biométrico Viridi 300.

Como se puede observar en la figura 5.3 se muestra el diagrama a bloques del sistema biométrico y la posible distribución para el equipo Viridi 300, el cual, puede operar de tres formas distintas, que se describen a continuación.

Caso 1: El equipo **Virdi 300** realiza las dos etapas del sistema biométrico (indicado con un cuadro negro), el enrolamiento y la verificación ó identificación según sea el caso. De acuerdo al resultado, realiza una función según como se esté usando el equipo, como control de asistencia ó control de acceso.

Caso 2: El equipo **Virdi 300** realiza una etapa del sistema biométrico (indicado con cuadros rojos), la verificación ó identificación según sea el caso. La etapa de enrolamiento se realiza en otra parte remota al dispositivo y las plantillas de las huellas en ese mismo lugar donde se realiza el enrolamiento. De acuerdo al resultado, realiza una función según como se esté usando el equipo, como control de asistencia ó control de acceso.

Caso 3: El equipo **Virdi 300** realiza una etapa del sistema biométrico (indicado con cuadros verdes), la verificación ó identificación según sea el caso. La etapa de enrolamiento se realiza en otra parte remota al dispositivo y las plantillas (o patrones) de las huellas son almacenadas en una base de datos remota tanto al dispositivo como al área de enrolamiento. De acuerdo al resultado, realiza una función según como se esté usando el equipo, como control de asistencia ó control de acceso.

Dispositivo STCA02

El segundo equipo es el **STCA02**, el cual, es un sistema profesional para control de acceso a instalaciones o zonas restringidas permitiendo el paso sólo al personal autorizado. El sistema **STCA02** le ofrece el control de accesos por zonas, personas y horarios. De esta forma, se puede definir: los horarios de acceso al personal en general, los horarios de acceso restringido por puertas específicas, así como también los horarios especiales para personal clave. Por ejemplo, directivos con acceso libre. Contiene un lector óptico que almacena hasta 500 plantillas de huellas digitales que se almacenan internamente para autenticar el acceso de los usuarios a sus instalaciones.

El sistema **STCA02** ofrece conectividad a través de una red local o Internet, lo cual, permite tener el control de sucursales, almacenes y oficinas geográficamente distribuidas. Tiene la capacidad de operar de manera independiente y almacenar hasta 30,000 eventos para mayor información consultar la referencia [ST09b]. El diagrama a bloques de este sistema biométrico es similar al equipo Virdi 300, ver figura 5.3. Presenta igualmente los tres casos anteriores.

Dispositivo STCA04

El tercer equipo es el **STCA04** es un sistema profesional de accesos y tiene el control del paso a sus instalaciones y zonas restringidas permitiendo el paso sólo a las personas autorizadas en su organización. Contiene un lector óptico que almacena hasta 1.000 plantillas de huellas digitales que se almacenan internamente para autenticar el acceso de los usuarios a sus instalaciones.

Dispositivo STLK1000

Este dispositivo es la cerradura biométrica de la línea **STLK1000**, es ideal para instalarse en puertas de acceso restringido. Posee una capacidad de almacenamiento hasta 150 usuarios. Es ideal para controlar el acceso a las instalaciones, que solo requiera el franqueo de paso al mismo. El diagrama a bloques de este sistema biométrico se muestra en la figura 5.3. Este dispositivo sólo opera de acuerdo al caso 1, donde todo el sistema biométrico lo realiza el dispositivo y solamente opera del modo identificación, para mayor información consultar la referencia [ST09a].

Dispositivo STUU5

El último dispositivo que se mencionará es el lector de huellas dactilares **STUU5**, el cual, es un sistema óptico fino, sólo mide 6 mm, La imagen puede alcanzar 700 dpi, es excelente para comparar huellas y para inscribirlas, es adecuado para utilizarse ampliamente en muchos ámbitos como en seguridad social, seguridad pública, administración de asistencia, etc. Para saber más se puede consultar la referencia [ST09c]. El diagrama a bloques del sistema biométrico se muestra en la figura 5.4.

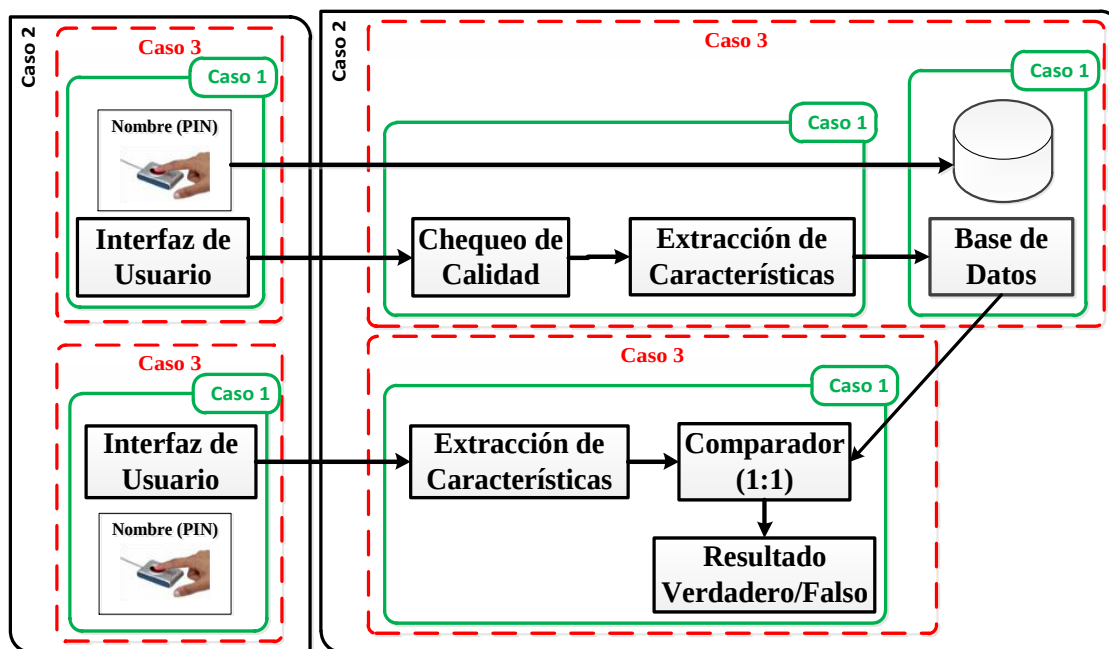


Figura 5.4: Diagrama a bloques del sistema biométrico STUU5.

Comentario: Como se puede ver en la figura 5.4 el lector se conecta en un dispositivo que realiza los demás procesos del sistema biométrico, el papel del lector es solamente de adquirir la imagen, la cual, envía al dispositivo que la procesa. En nuestro caso, como el lector y el dispositivo que realiza los procesos se localizan en el mismo lugar se consideran los casos que se muestran en la figura 5.3.

En la figura 5.4 se muestra el diagrama a bloques del sistema biométrico usando el lector de huellas STUU5, como este lector solo adquiere la imagen de forma digital, el sistema biométrico puede estar distribuido de tres formas distintas, descritas a continuación.

Caso 1: El lector de huellas se conecta a un dispositivo que realiza la etapa de registro, el cual, realiza el chequeo de la imagen y extracción de características para después almacenarlas en una base de datos remota. O bien, se puede conectar a un dispositivo que realiza la etapa de verificación ó identificación según sea el caso.

Caso 2: El lector se conecta a un dispositivo que realiza las dos etapas del sistema biométrico, la identificación ó verificación según sea el caso y el enrolamiento.

Caso 3: El lector de huellas se conecta a un dispositivo que realiza la etapa de registro, el cual, realiza el chequeo de la imagen y extracción de características para después almacenarlas en una base de datos. O bien, se puede conectar a un dispositivo que realiza la etapa de verificación ó identificación según sea el caso.

Después de describir los diferentes escenarios que se pueden presentar utilizando los dispositivos mencionados en un sistema biométrico de huella dactilar. Se puede llegar a la decisión de seleccionar el caso que se muestra en la figura 5.5, en el que, la etapa de registro, el sistema de base de datos y la etapa de verificación o identificación según se al caso de aplicación, se encuentran separados remotamente. La decisión de seleccionar este escenario se debió, a que es el escenario donde más expuestas están a sufrir ataques las líneas de comunicación de los sistemas biométricos.

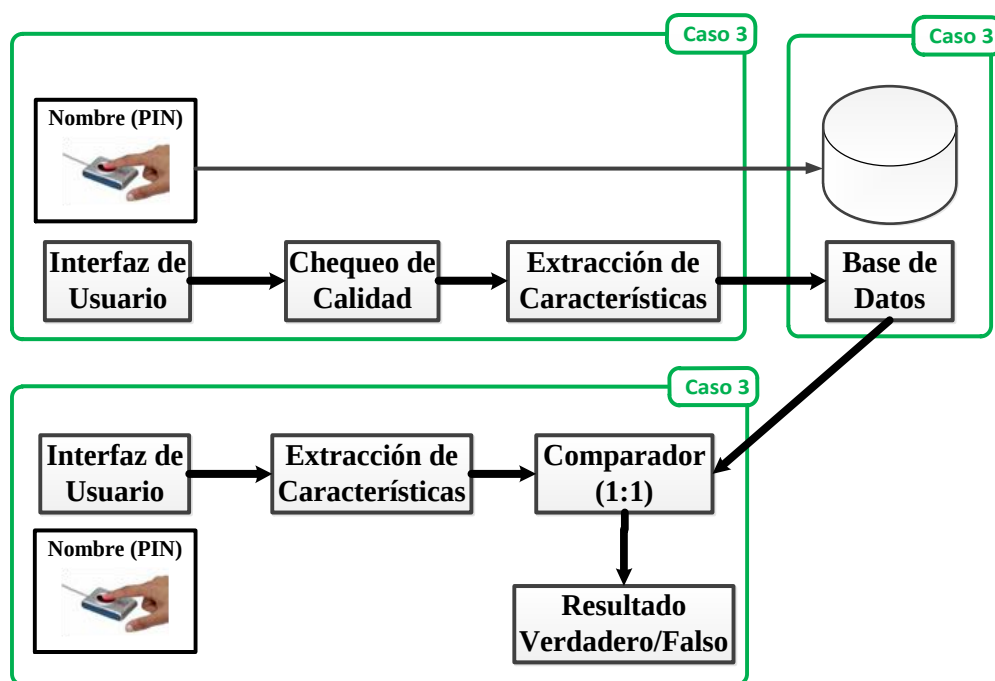


Figura 5.5: Diagrama de distribución del escenario seleccionado del sistema biométrico.

Otro aspecto que se debe tomar en cuenta, es que se decidió omitir al proceso de extracción de características que realizan los sistemas biométricos, debido a esto se buscó una forma de comparación alternativa para la comparación de la imagen de las huellas dactilares.

5.3. Alternativa de comparación

En los sistemas biométricos la comparación de huellas dactilares se realiza por medio de las minucias que contiene la huella dactilar. Sin embargo, en este trabajo de tesis se utilizará un método alternativo de comparación de imágenes de huellas dactilares, el cual, consiste en comparar sus histogramas siendo esta una función discreta que representa el número de pixeles que componen a una imagen en función de los niveles de intensidad como se muestra en la figura 5.6, cuando se trata de una imagen en escala de gris.

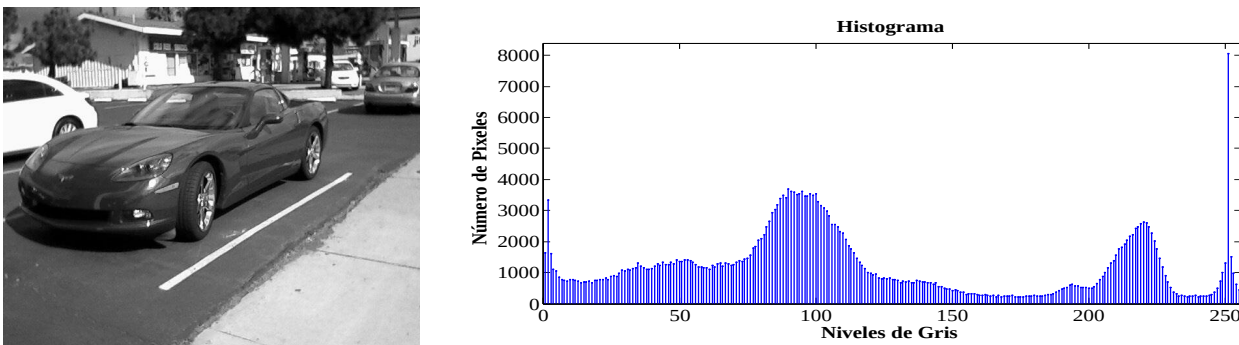


Figura 5.6: Imagen de un carro y del lado derecho el histograma obtenido de la imagen.

Hay una característica que se tomó en cuenta para considerar la comparación de huellas por medio de sus histogramas, es que una imagen puede generar sólo un histograma. Sin embargo, un histograma puede pertenecer a varias imágenes, esto se puede entender mejor observando la figura 5.7.

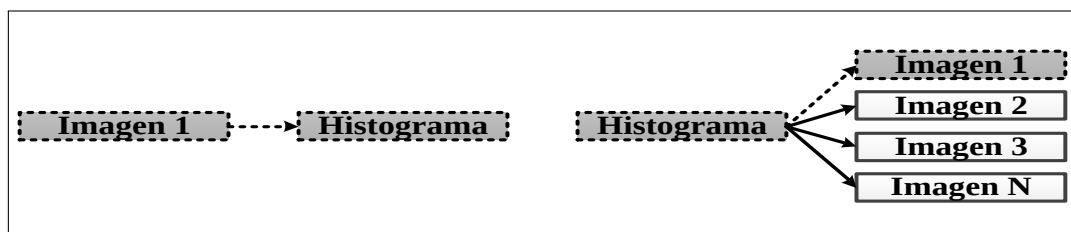


Figura 5.7: Diagrama a bloques que muestra como un histograma puede pertenecer a numerosas imágenes.

Comentario: Tomando en cuenta el hecho de que el histograma que se genera es básicamente una gráfica y que una imagen puede generar sólo un histograma, se realizaron algunas pruebas para comprobar ¿qué tan efectiva podría ser la comparación por medio de este método?.

Para determinar la diferencia de la comparación de histogramas de las imágenes de las huellas, se recurrió a las siguientes expresiones matemáticas:

$$Diferencia(i) = |Histograma_{BD}(i) - Histograma_{Usuario}(i)|, \quad (5.1)$$

$$Resultado_C = \sum_{i=0}^{255} Diferencia(i), \quad (5.2)$$

donde:

i	Es el valor del pixel del histograma en escala de grises.
$Diferencia(i)$	Es la diferencia de pixeles en el nivel de gris i en el histograma.
$Histograma_{BD}(i)$	Es número de pixeles de valor i en el histograma de la base de datos.
$Histograma_{Usuario}(i)$	Es número de pixeles de valor i en el histograma de los usuarios.
$Resultado_C$	Es el resultado de la comparación.

5.4. Alternativa de encriptado

En este trabajo de tesis se propone utilizar como alternativa de encriptado, un sistema caótico para ocultar la información biométrica (la huella dactilar), durante su traslado en un sistema biométrico entre lugares remotos, el proceso de encriptado que se pretende realizar será sencillamente por adición, el cual, se muestra en el diagrama a bloques de la figura 5.8. Reconociendo que en la literatura correspondiente, se reportan otros más seguros, ver por ejemplo [MC07], [DC05], [HKKP06], [PLC08], [YWC97], [CS05], [MEN98], [CRU04], [CRU08], [WON02], [CMC04], [PL01], [LC05], [AC09], [LLT06], [KRS00], [TL01], [GCLG08], [GCLG09], [AC+10], [LPLC09] y [CM10].

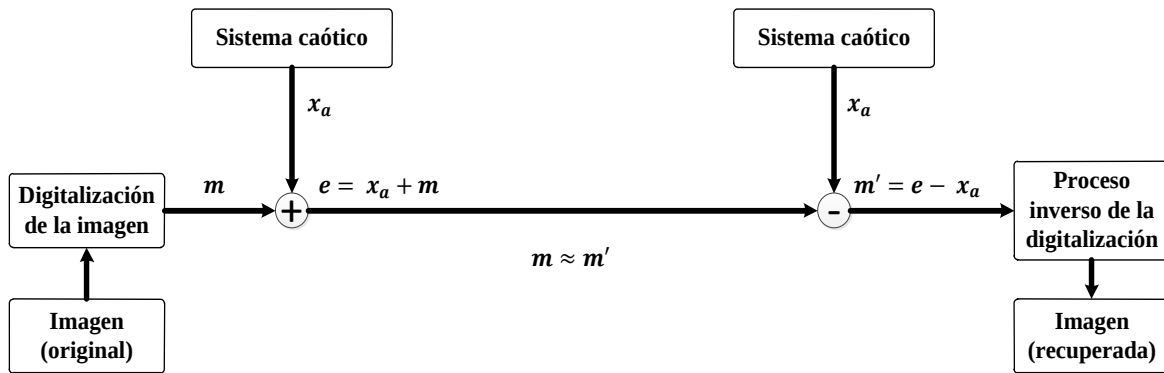


Figura 5.8: Diagramas a bloques del proceso de encriptado y desencriptado por medio de encriptado caótico aditivo.

Para una mayor comprensión del diagrama mostrado en la figura 5.8, a continuación se explicará más detalladamente cada uno de los bloques del diagrama y el significado de cada una de las variables presentadas en el diagrama.

Imagen (original). En este bloque se realizará el proceso de adquirir una imagen de la huella dactilar en forma de una matriz $I(r, c)$, la cual, se llamará **matriz imagen**, siendo ésta con la que se trabajará en el siguiente proceso.

Digitalización de la imagen En este bloque se realizará el proceso de reacomodar cada uno de los elementos de la matriz imagen en un vector columna $v(r \times c)$, para después convertir cada uno de los elementos del vector a números binarios, obteniendo una nueva matriz $M(r \times c, 8)$, que posteriormente se reacomodarán sus elementos para convertirla de nuevo en un vector columna $vd(r \times c \times 8)$, el cual, se nombrará como **vector de datos**, siendo este vector considerado como el mensaje m a ser enviado a través del canal de transmisión.

Sistema caótico. En este bloque se realizará el proceso de generar un vector columna cuyos elementos serán los valores de uno de los estados caóticos del sistema seleccionado (mapeo de Rössler), siendo estos valores obtenidos en cada una de las iteraciones realizadas al sistema caótico, este vector columna tendrá las mismas dimensiones que el vector de datos, el cual, se llamará **vector caótico** x_a .

Punto de suma. Se realiza una suma del vector de datos con el vector caótico obteniendo un nuevo vector, siendo este vector el mensaje cifrado e (criptograma), obtenido como salida del punto de suma.

Punto de resta. En este proceso se obtendrá el mensaje recuperado m' , para lo cual, se realizará una resta del mensaje cifrado menos el vector caótico.

Proceso inverso de la digitalización. En este bloque se realizará el proceso inverso realizado en la digitalización con el objetivo de recuperar la matriz de imagen a partir del vector de datos recuperado.

Imagen (recuperada). Se ilustrará la imagen recuperada.

Como se puede ver en la figura 5.8 la imagen capturada de la huella dactilar, se tendrá que digitalizar primeramente, esto se debe a que omitimos el proceso de extracción de minucias (patrones) de los sistemas biométricos, el cual, da como resultado un vector de datos, siendo estos, los que proporcionan la información requerida para reconocer al usuario. Por lo cual, en este trabajo se tomó la decisión de remplazar el proceso de extracción de minucias por el proceso de digitalización, el cual, convierte la matriz de la imagen en un vector columna de ceros y unos, los cuales, representan la imagen. Pero si se realizara el proceso de extracción de minucias, el proceso de digitalización sería innecesario, esto se debe a que el proceso de digitalización no es propio del sistema biométrico sino que es propuesto en este trabajo, con el propósito de sustituir el proceso de extracción de minucias.

5.4.1. Sistema caótico como encriptador

En este trabajo de tesis se seleccionó el mapeo de Rössler, el cual, presenta una dinámica hipercaótica. Es decir presenta un comportamiento más complejo que un sistema caótico. Brevemente diremos que un sistema es hipercaótico sí tiene más de un exponente de Lyapunov positivo, para más detalles ver por ejemplo [CM10], [VS95], [JP93], [FC92], [EO92], [HG88], [SH94], [ANDO09], [BRA92], [MSR04] y [CS03]. La elección de un sistema hipercaótico es muy conveniente para nuestro propósito de encriptado, ya que esto permite que la información encriptada (la huella dactilar) este más segura, debido a la enorme riqueza dinámica del mapeo de Rössler. A continuación se muestran las ecuaciones de Rössler (5.3). Para mayor información consultar la referencia [ACLP08]:

$$\begin{cases} x_1(k+1) = \alpha x_1(k)(1 - x_1(k)) - \beta(x_3(k) + \gamma)(1 - 2x_2(k)), \\ x_2(k+1) = \delta x_2(k)(1 - x_2(k)) + \varsigma x_3(k), \\ x_3(k+1) = \eta((x_3(k) + \gamma)(1 - 2x_2(k)) - 1)(1 - \theta x_1(k)), \end{cases} \quad (5.3)$$

Las ecuaciones del mapeo de Rössler considerando las siguientes condiciones iniciales $x(0) = (0.3, 0, 0.05)$ y valores de parámetros $\alpha = 3.8$, $\beta = 0.05$, $\gamma = 0.35$, $\delta = 3.78$, $\varsigma = 0.2$, $\eta = 0.1$, $\theta = 1.9$ y con 28000 iteraciones, presenta un comportamiento hipercaótico por lo que las ecuaciones de Rössler genera las siguientes dinámicas para los tres estados mostrados en la figura 5.9. Los siguientes atractores caóticos generados por el mapeo de Rössler, son mostrados en la figura 5.10.

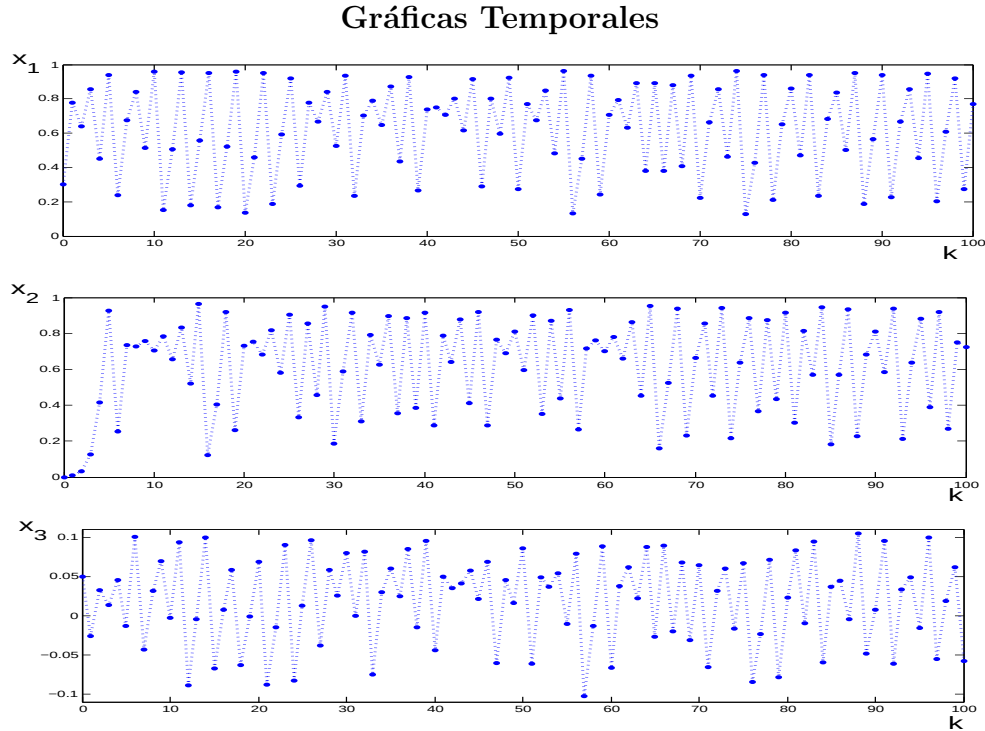


Figura 5.9: Gráficas temporales de las variables de estado del mapeo de Rössler (x_1, x_2, x_3) versus el número de iteraciones k .

Gráficas de Fase

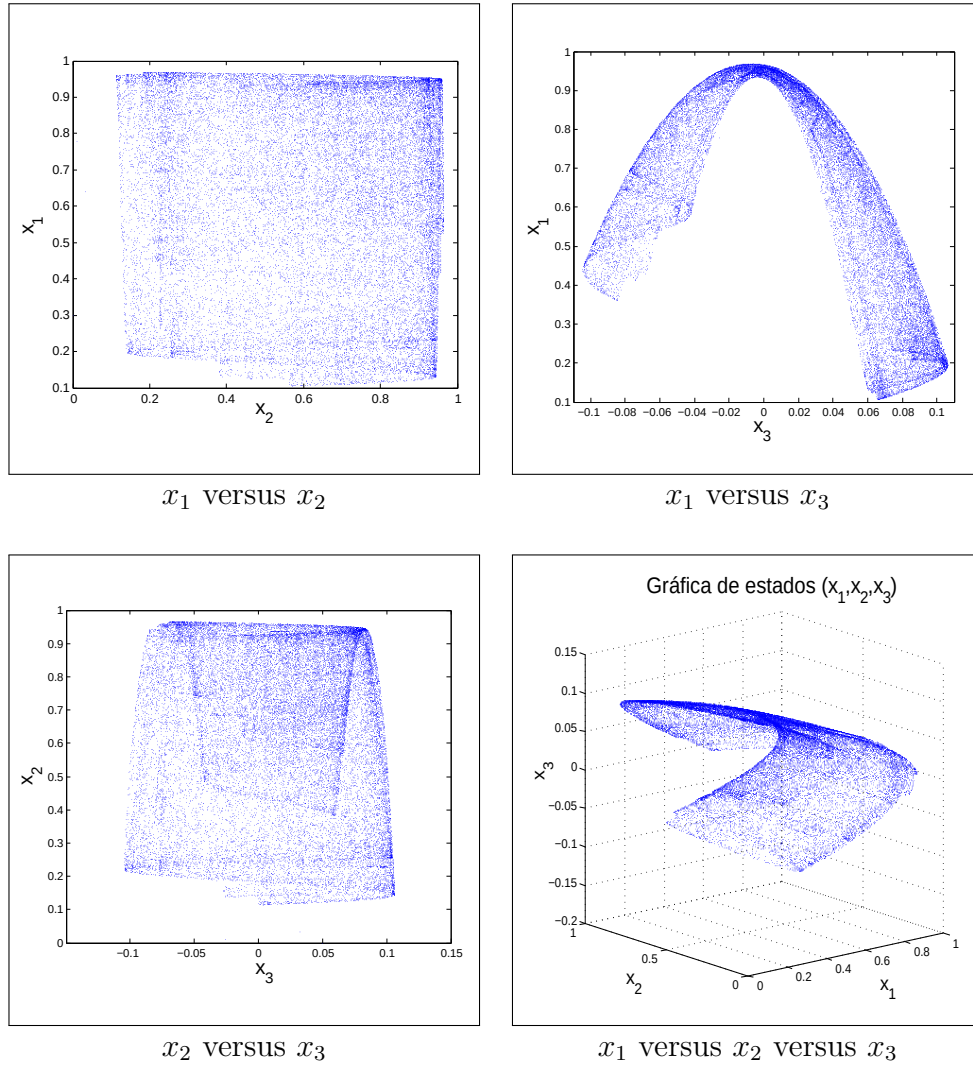


Figura 5.10: Atractores hipercaóticos del mapeo de Rössler.

5.5. Ubicación del encriptado

Tomando en cuenta los escenarios de distribución de los sistemas biométricos se consideró el caso de la figura 5.5, por lo que, el encriptado caótico se localizará después de la extracción de minucias para después enviar la información cifrada a la base de datos, esto le dará un nivel de protección a la información ante algunos posibles ataques antes de llegar a su destino, la base de datos. La parte del descryptado se localizará en la entrada de la línea de información proveniente de la base de datos a la parte de verificación o identificación según sea el caso, esto para descifrar la información antes de su comparación. Lo anterior se ilustra en el diagrama que se muestra en la figura 5.11.

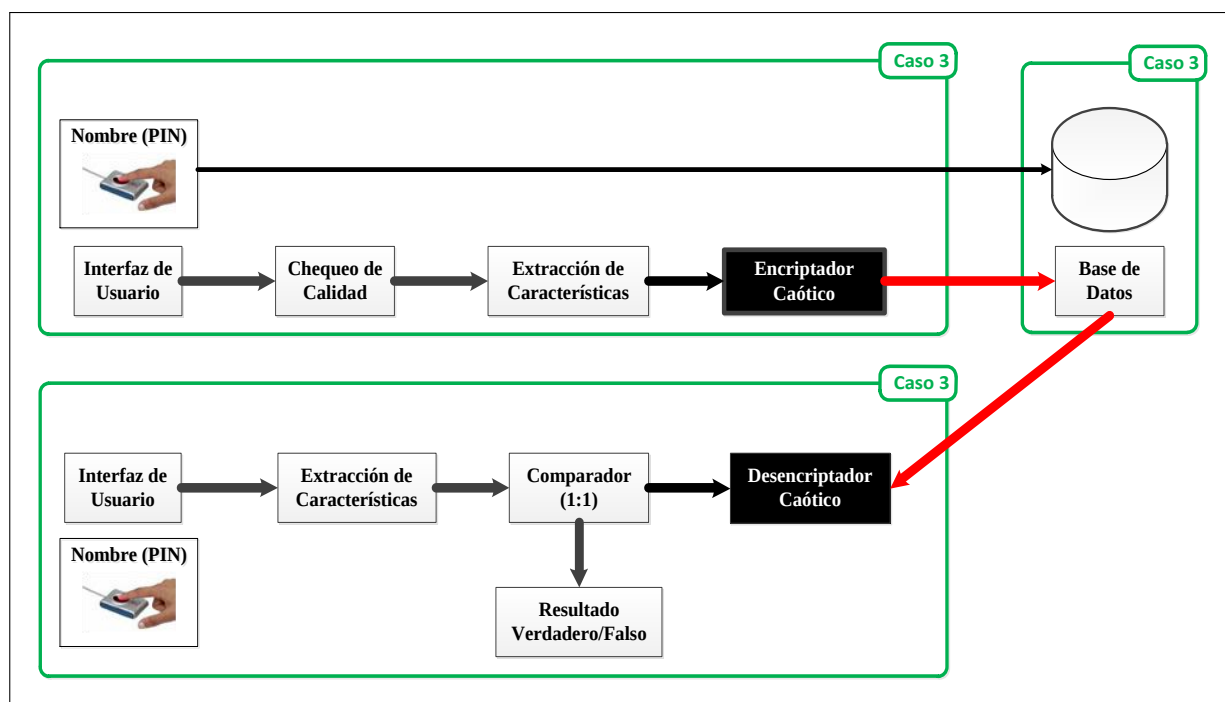


Figura 5.11: Diagrama a bloques del sistema biométrico con las etapas de encryptado y decryptado, los cuales, se representan por los bloques rojos.

Comentario: En la figura 5.11 se pueden observar unos cuadros de color verde, estos indican las etapas requeridas para realizar el encryptado caótico, también se observan unas flechas de color rojo, las cuales, simbolizan las líneas de transmisión que se protegerán ante los posibles ataques que se realicen para adquirir la información biométrica del usuario. También se puede ver que al poner el encryptado en la parte que se observa en la figura 5.11 esto permite que esta parte de la etapa del sistema biométrico, no solo protege la información durante su envío sino que además esto también sirve como protección durante el tiempo que se encuentre en la base de datos, esto se debe a que la información estará encryptada durante su almacenamiento dándole así una seguridad extra a la información.

5.6. Conclusiones

Para desarrollar el trabajo de tesis se llevaron a cabo varias modificaciones a los procesos que se realizan en los sistemas biométricos comerciales, por lo que, se obtendrá una menor precisión de reconocimiento, siendo este factor perjudicial para los sistemas biométricos. Sin embargo, el propósito del trabajo de tesis no es aumentar la precisión del reconocimiento, sino proporcionar seguridad al identificador biométrico (en nuestro caso, las huellas dactilares) en la comunicación, por lo cual, resultaban necesarias estas modificaciones para observar los resultados de la alternativa de encryptado propuesto en este trabajo de tesis.

Sin embargo, los efectos negativos que puedan provocar las modificaciones de los procesos de un sistema biométrico comercial, pueden ser tomados como ventaja, ésto se debe a que si el reconocimiento del identificador biométrico (huella dactilar) se logra después de pasar por el proceso de encriptado y desencriptado, bajo estas condiciones del sistema biométrico. Esto podría indicar que la implantación de este encriptado no afectaría en gran medida al reconocimiento del identificador biométrico de los sistemas biométricos comerciales.

Capítulo 6

Pruebas y resultados

6.1. Introducción

El presente capítulo está dedicado a mostrar las pruebas realizadas y resultados obtenidos durante la elaboración de este trabajo de tesis, esto con el objetivo de que el lector observe las contribuciones del trabajo de tesis en la dirección de la seguridad de la información biométrica. El capítulo se encuentra dividido en dos partes para su mejor comprensión. El primero es el problema de verificación de un usuario, el segundo es el problema de identificación de múltiples usuarios.

6.2. Problema de verificación de un usuario

Un problema que se abordó en este trabajo de tesis es la verificación de un usuario en el control de accesos, el cual, es uno de los problemas que comúnmente atienden los sistemas biométricos. Por tanto, se realizaron algunas pruebas utilizando el método alternativo de comparación y verificación propuesto en este trabajo de tesis.

6.2.1. Definición del problema de verificación de un usuario

El primer problema que atiende un sistema biométrico, y que incluye en su estudio esta tesis es la ***verificación de un usuario*** por medio de su huella dactilar, a que se refiere verificación de un usuario, esto quiere decir, que el sistema biométrico verificará la existencia del registro de un individuo, el cual, con anterioridad se registró en la base de datos. Algunos ejemplos de este tipo de situaciones son: ***el acceso a una cuenta de correo electrónico, el acceso al sistema operativo de una PC, incluso el acceso a una cuenta bancaria***, etc. Para una mayor comprensión el lector puede observar en la figura 6.1, la verificación de un usuario (Bob).

En las situaciones mencionadas, sólo se permite el acceso a una persona a un servicio y solamente esa persona tiene la autorización de hacer uso de él. Es decir, si otra persona intentará solicitar acceso al servicio de referencia, ésta será rechazada por el sistema biométrico.

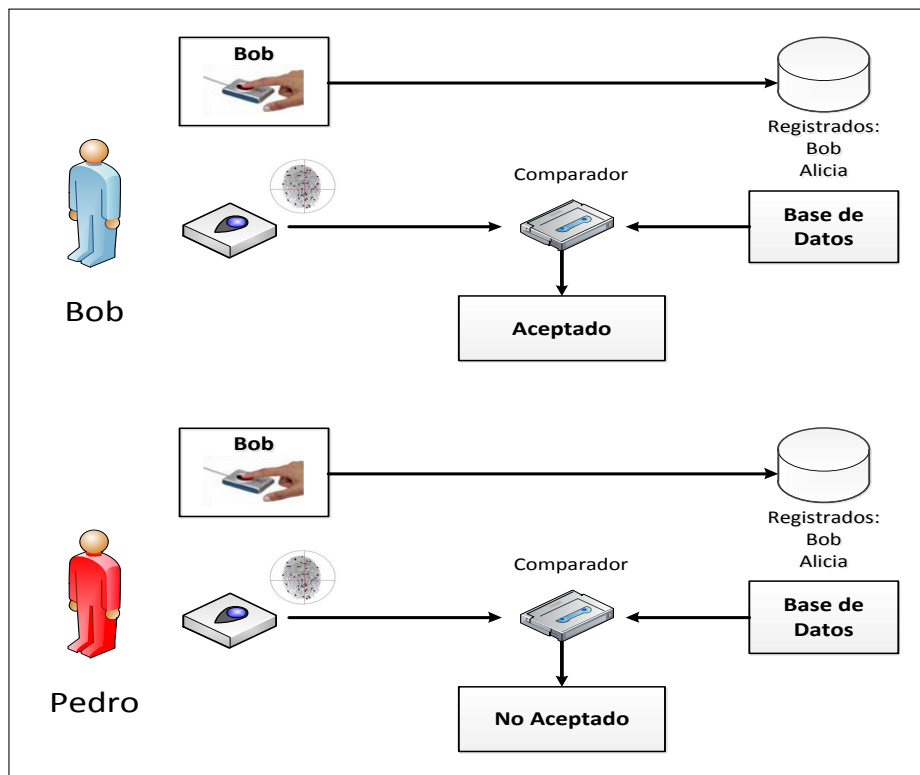


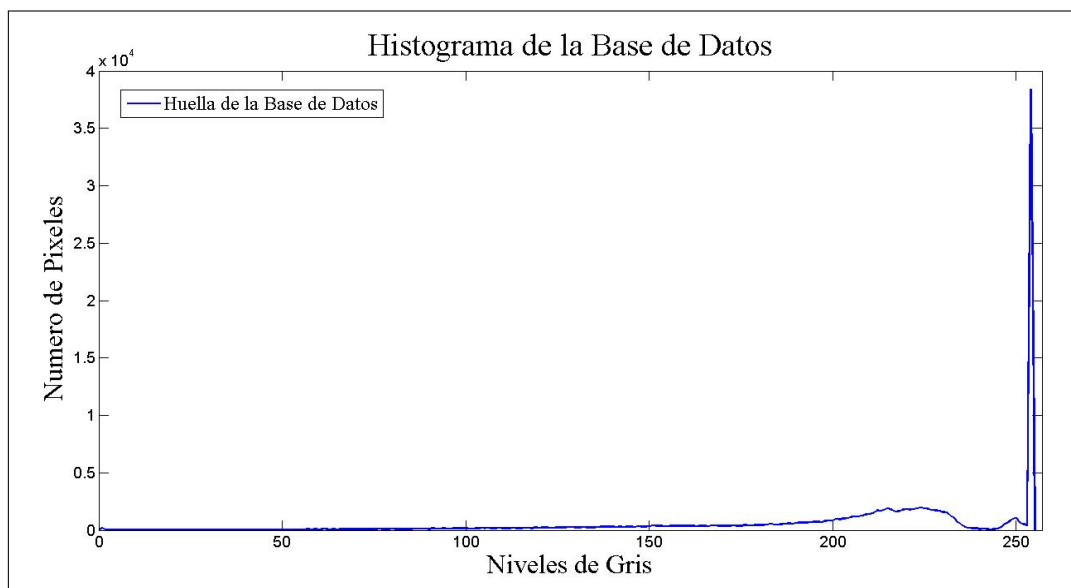
Figura 6.1: Diagrama esquemático que ilustra el problema de verificar el registro de un usuario.

6.2.2. Pruebas realizadas para la posible solución del problema

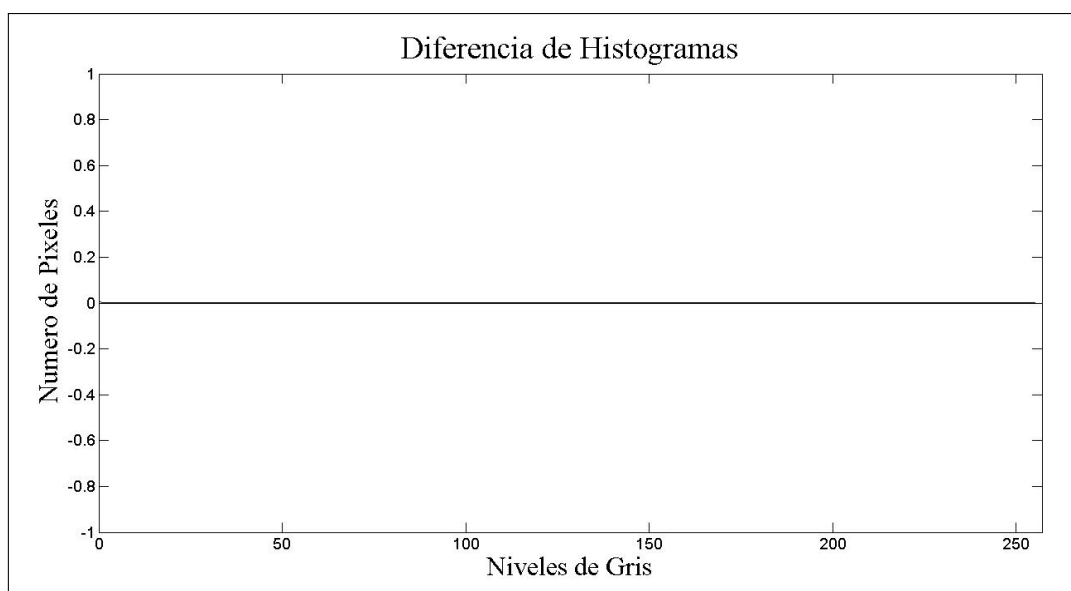
Para la realización de las siguientes pruebas se seleccionó a una persona como usuario autorizado, a la cual, se le tomó la imagen de la huella de su pulgar izquierdo, que posteriormente se almacenó en la base de datos, etiquetándolo como usuario autorizado, es decir, será al único que el sistema biométrico reconocerá y por tanto, permitirá el acceso. Después para comprobar que al obtener el histograma de la misma imagen en distintos tiempos se tendrá como resultado el mismo histograma y por consiguiente, el resultado de la comparación de histogramas de la misma imagen será igual con 0. Sabiendo esto, se procedió a obtener el histograma de la huella en dos ocasiones, para posteriormente compararlos. En la figura 6.2 se puede observar: a) la huella dactilar del pulgar izquierdo que se almacenará en la base de datos (usuario autorizado, es decir, registrado en la base de datos), b) el histograma correspondiente a la huella que se encuentra en la base de datos y c) el resultado de la comparación con los histogramas obtenidos de la huella registrada en la base de datos.



a)



b)



c)

Figura 6.2: Información de la imagen: a) la huella dactilar del pulgar izquierdo, b) histograma de la huella y c) comparación de histogramas de exactamente la misma imagen.

Lo siguiente fue tomar una imagen del mismo dedo e individuo con las condiciones lo más parecidas posible, así como también tomar las huellas del mismo dedo de otras cuatro personas para después compararlas con la que se encuentra en la base de datos, el proceso que se realizó para la comparación fue el siguiente, se toma la imagen de la primera persona que solicita acceso, con el motivo de obtener el histograma de la imagen de ingreso, después se compara con el histograma de la imagen que se encuentra en la base de datos, con el propósito de obtener la diferencia de los niveles de gris de estos histogramas, siendo esta información de ayuda para determinar el número de píxeles distintos entre las dos imágenes, esta diferencia es la que se da como resultado de la comparación. Este proceso se puede entender mejor con la ayuda del diagrama que se muestra en la figura 6.3.

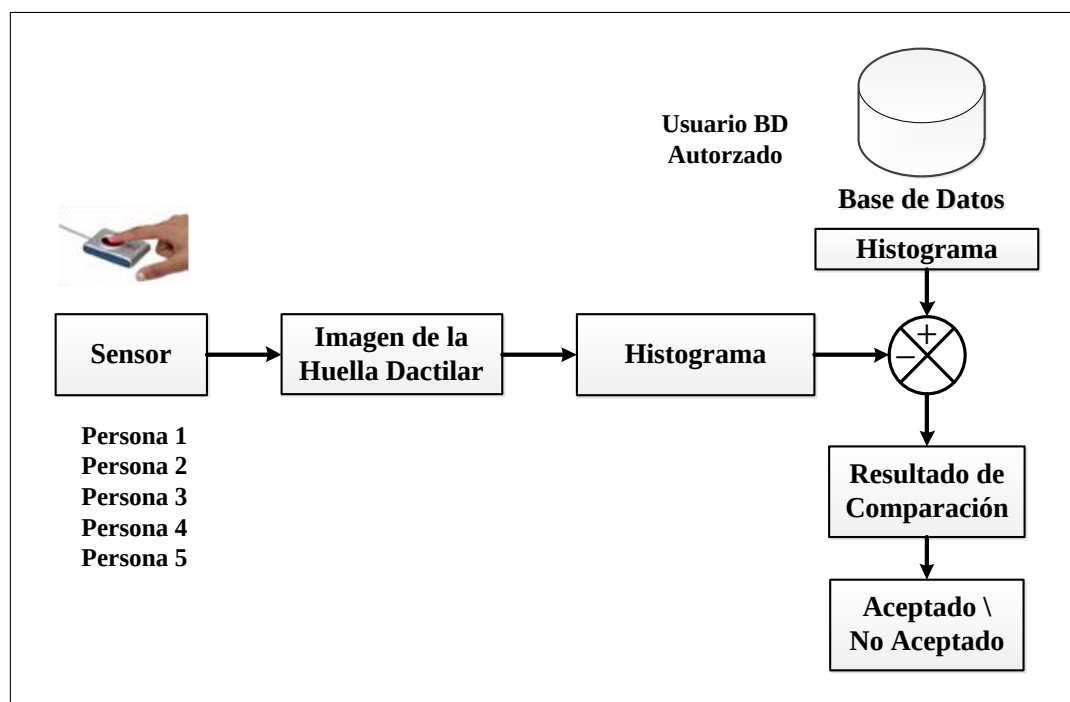


Figura 6.3: Diagrama esquemático para la comparación de cinco histogramas de personas que solicitan acceso versus el histograma del usuario autorizado.

Etapa de calibración

El propósito de la etapa de calibración consiste en realizar una prueba de comparación de histogramas para obtener un valor numérico, el cual, se utilizará como umbral para diferenciar a la persona registrada (autorizada) de las que no están registradas en la base de datos (no autorizadas). Con tal propósito, lo primero que se realizó fue adquirir cinco imágenes de huellas dactilares, dentro de las cuales una de las muestras adquiridas pertenece al pulgar izquierdo de la persona registrada en la base de datos, el resto de las muestras pertenecen al pulgar izquierdo de personas no registradas. En la figura 6.4 se pueden observar las imágenes adquiridas por el sensor, siendo este el primer conjunto de huellas dactilares. Mientras que en la figura 6.5 se pueden observar los histogramas del primer conjunto de muestras.



Figura 6.4: Imágenes del primer conjunto de cinco muestras de huellas de personas que solicitan acceso.

Comentario: Las imágenes de las huellas de las cinco personas que se muestran en la figura 6.4, se tomaron en las condiciones más similares con respecto a la imagen que se encuentra en la base de datos, esto se realizó limpiando el lector en cada lectura de muestra, limpiando el dedo de la persona a quien se le tomaba la muestra, colocando el dedo de la persona en la misma posición para los cinco individuos y teniendo la misma iluminación para todas la tomas de muestras. Se debe mencionar que la huella del pulgar de la persona 1 pertenece al mismo individuo que proporciono la huella que se encuentra almacenada en la base de datos.

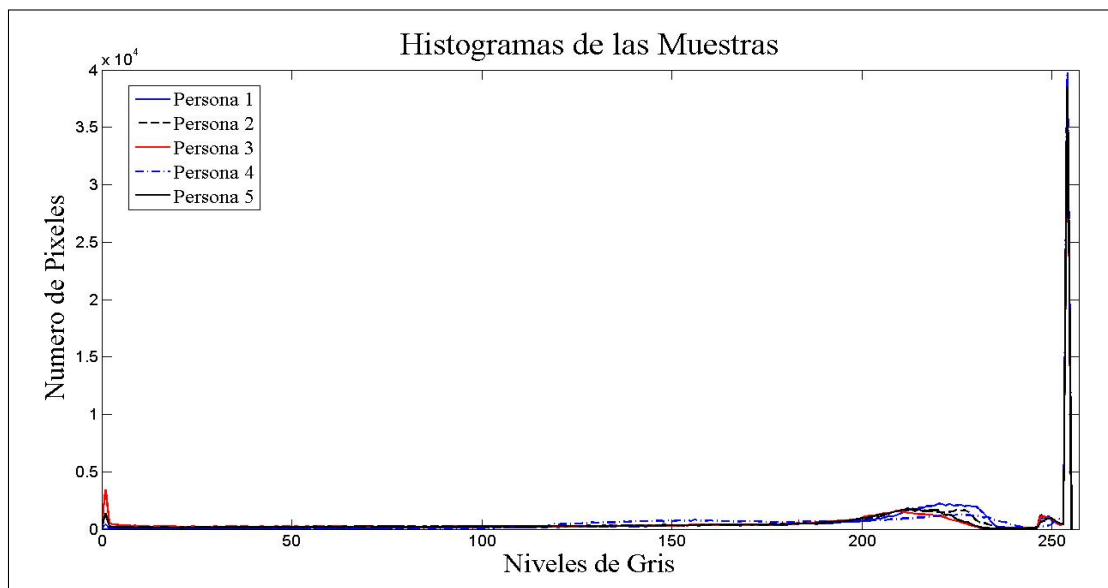


Figura 6.5: Histogramas del primer conjunto de cinco muestras de personas.

Comentario: En la figura 6.5 se muestran los histogramas del primer conjunto de huellas, de los cuales, se observa que la distribución en los niveles de gris de las huellas son muy parecidas, sin embargo tienen diferencias en algunos niveles de gris, lo cual, permitirá realizar la comparación para la toma de decisión de acceso o no acceso.

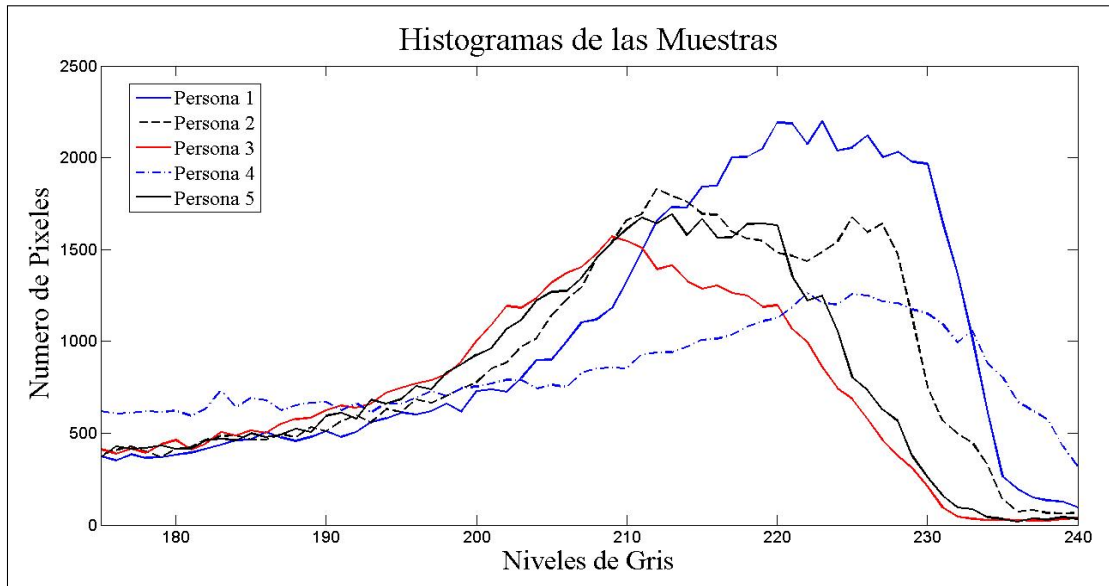


Figura 6.6: Sección ampliada de la gráfica de los histogramas del primer conjunto de cinco muestras de personas.

Comentario: Debido a que en la figura 6.5 no se alcanzan apreciar adecuadamente las diferencias de los histogramas. Se decidió mostrar una figura aumentando una sección de la gráfica, ésta se muestra en la figura 6.6, en la cual, es más sencillo apreciar estas diferencias.

Después de mostrar el primer conjunto de imágenes de huellas dactilares, lo que procede a continuación es mostrar los resultados numéricos obtenidos en la comparación de los histogramas, correspondientes a las imágenes del primer conjunto de huellas dactilares versus el histograma de la imagen registrada en la base de datos. En la tabla 6.1 se muestran los resultados numéricos obtenidos con la ayuda del programa que se incluye en el anexo A.1, el cual, realiza las operaciones que indican las expresiones matemáticas (5.1) y (5.2).

Persona	<i>Resultado_C</i>
1	14486
2	24046
3	46958
4	52990
5	35442

Tabla 6.1: Resultados de la diferencia de los cinco histogramas del primer conjunto versus el histograma de la base de datos.

Con motivo de complementar la información mostrada en la tabla 6.1, a continuación se muestran cada uno de los histogramas del primer conjunto con el histograma de la base de datos y el resultado de la comparación de los mismos. Esto se obtuvo con la ayuda del

código del programa en MatLab que se incluye en el anexo A.2, sin embargo del código sólo se utilizó lo que involucra a los histogramas.

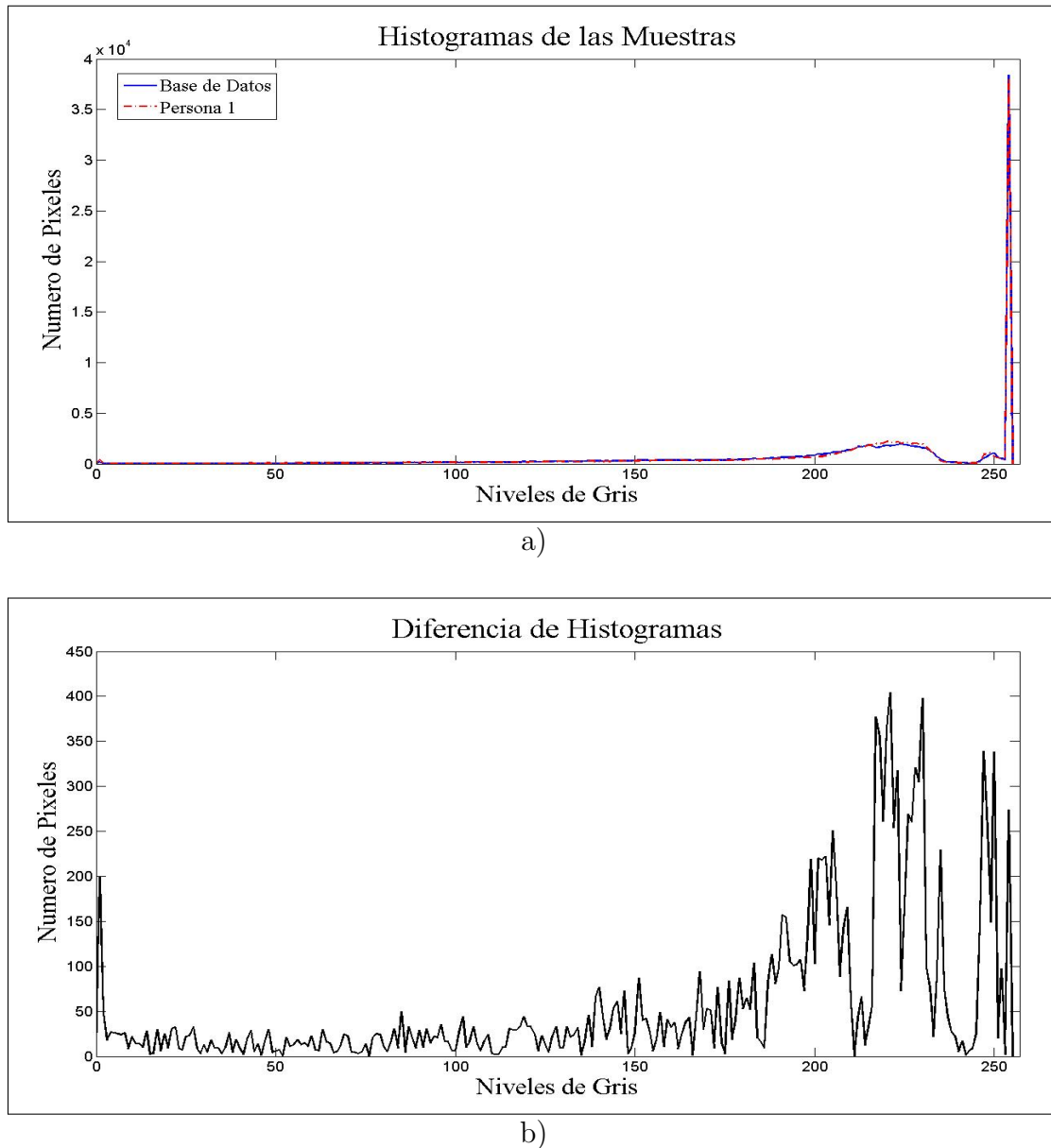
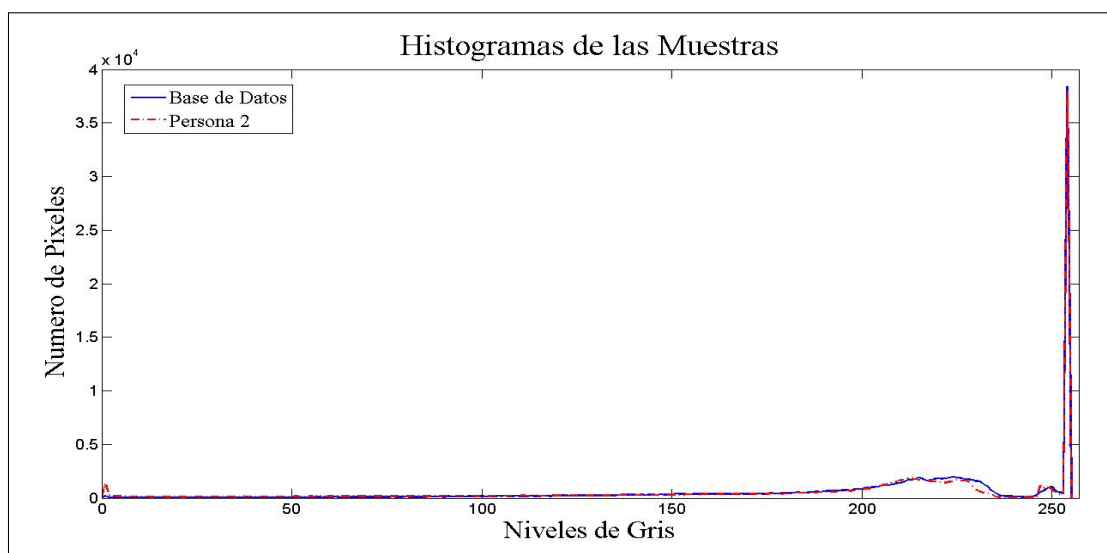
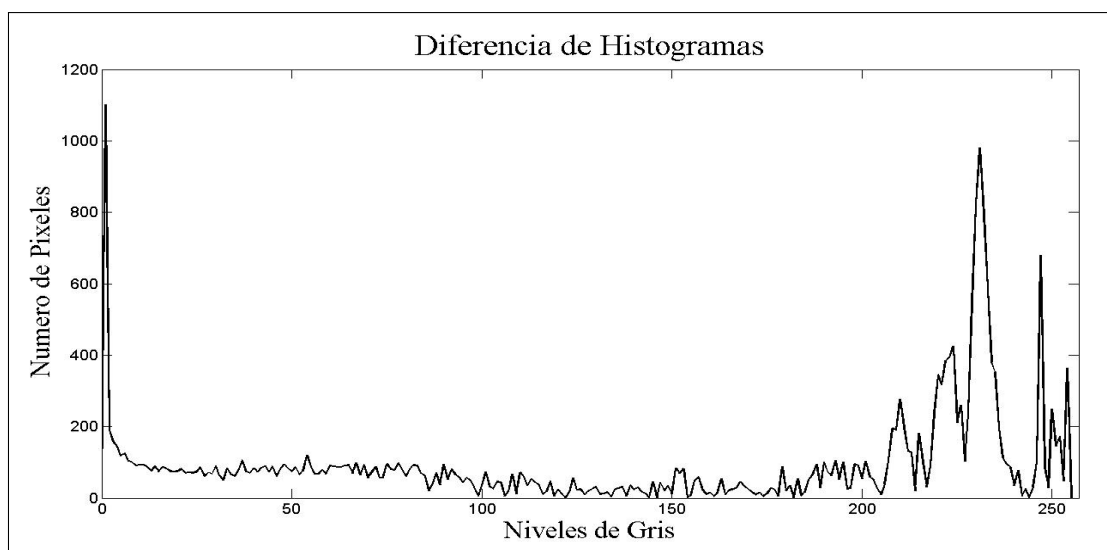


Figura 6.7: Comparación de histograma para la persona 1 del primer conjunto versus el histograma de la base de datos: a) muestras a comparar y b) diferencia.

Comentario: En la figura 6.7 a) se muestra el histograma que se encuentra en la base de datos, también el histograma de la imagen de la huella de la persona 1 del primer conjunto de muestras y en la figura 6.7 b) la diferencia de los histogramas mostrados en la figura 6.7 a). Se puede observar que en la figura 6.7 b) el número de pixeles en los niveles de gris no sobrepasan el valor de los 450.



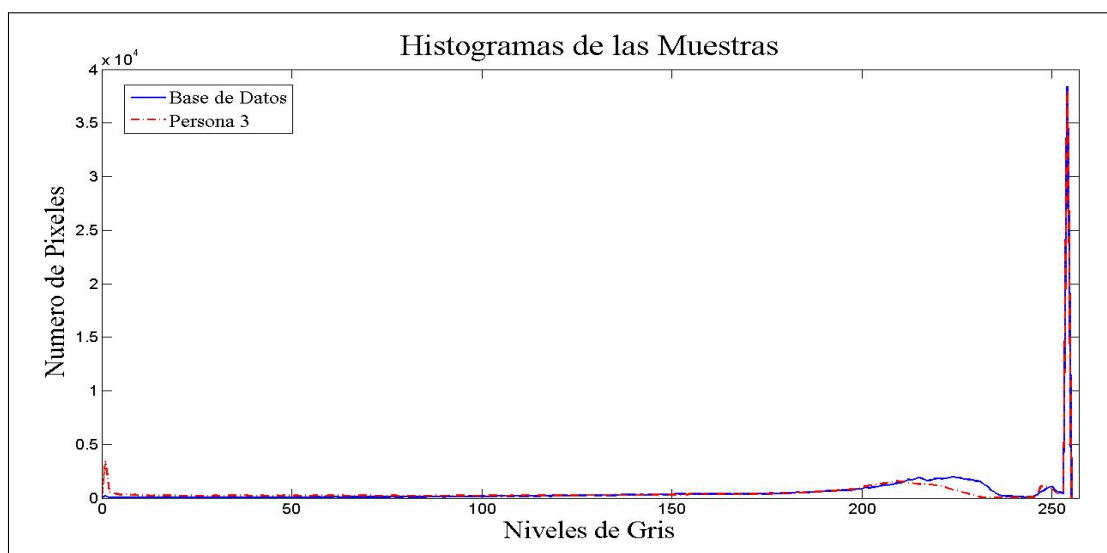
a)



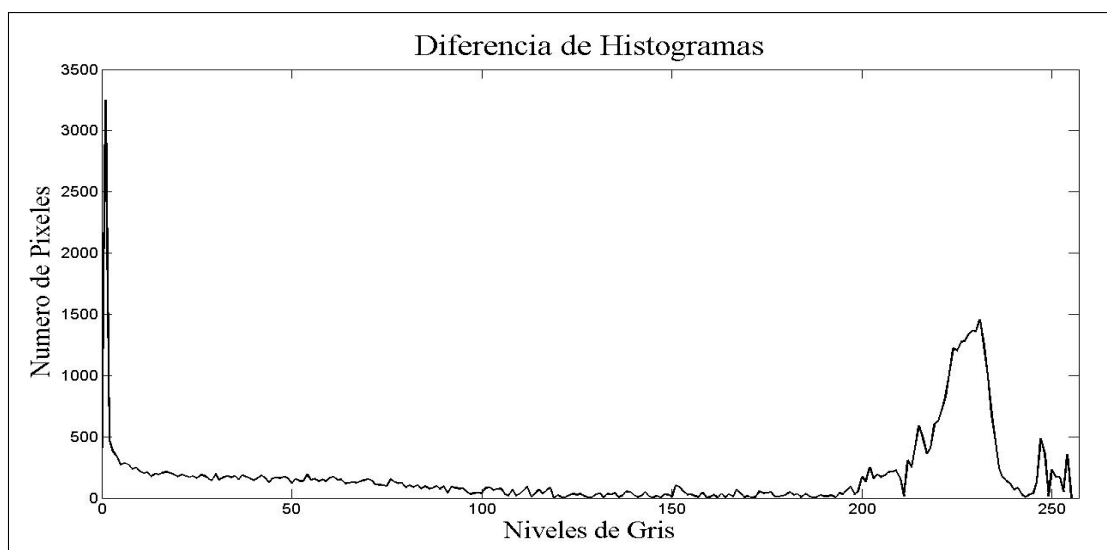
b)

Figura 6.8: Comparación de histograma para la persona 2 del primer conjunto versus el histograma de la base de datos: a) muestras a comparar y b) diferencia.

Comentario: En la figura 6.8 a) se observa el histograma que se encuentra en la base de datos, también el histograma de la imagen de la huella de la persona 2 del primer conjunto de muestras y en la figura 6.8 b) la diferencia de los histogramas mostrados en la figura 6.8 a). Se puede observar que en la figura 6.8 b) el número de pixeles en los niveles de gris sobrepasan el valor de los 450 e incluso algunos valores están cercanos a los 1 000.



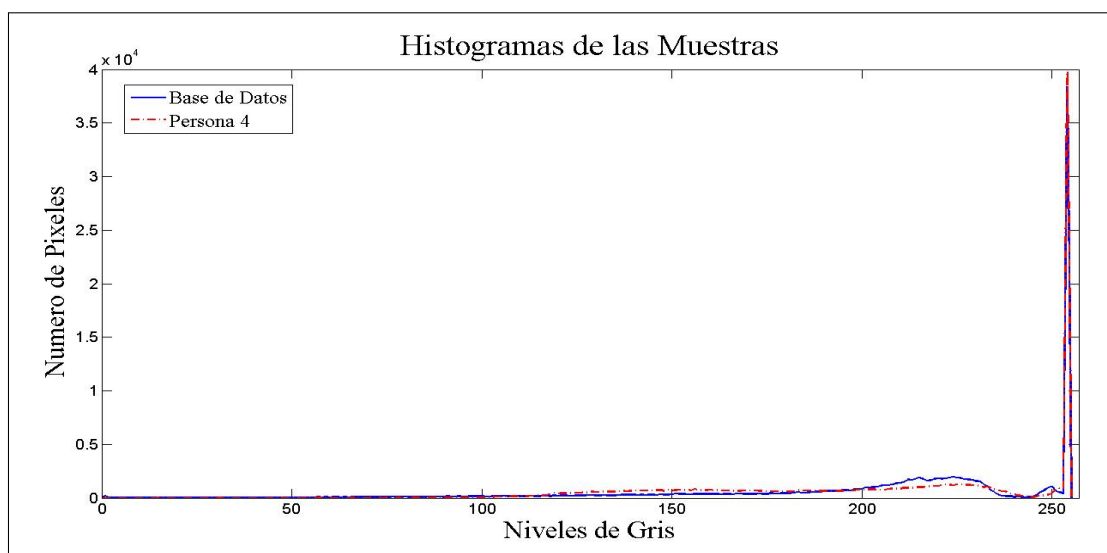
a)



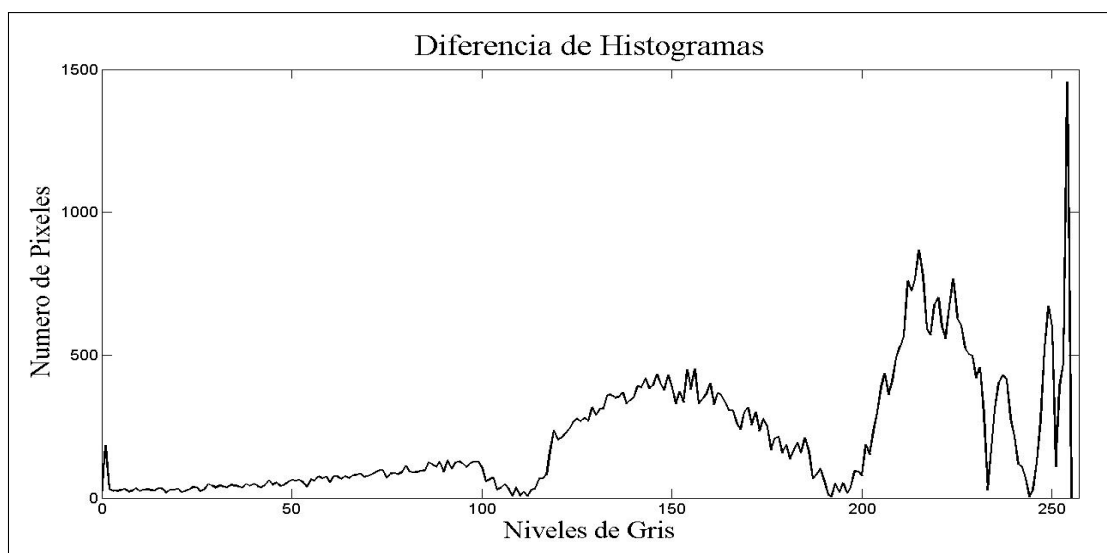
b)

Figura 6.9: Comparación de histograma para la persona 3 del primer conjunto versus el histograma de la base de datos: a) muestras a comparar y b) diferencia.

Comentario : En la figura 6.9 a) se muestra el histograma que se encuentra en la base de datos, también el histograma de la imagen de la huella de la persona 3 del primer conjunto de muestras y en la figura 6.9 b) la diferencia de los histogramas mostrados en la figura 6.9 a). Se puede observar que en la figura 6.9 b) el número de píxeles en los niveles de gris sobrepasan el valor de los 450 e incluso algunos valores también superan los 1 500.



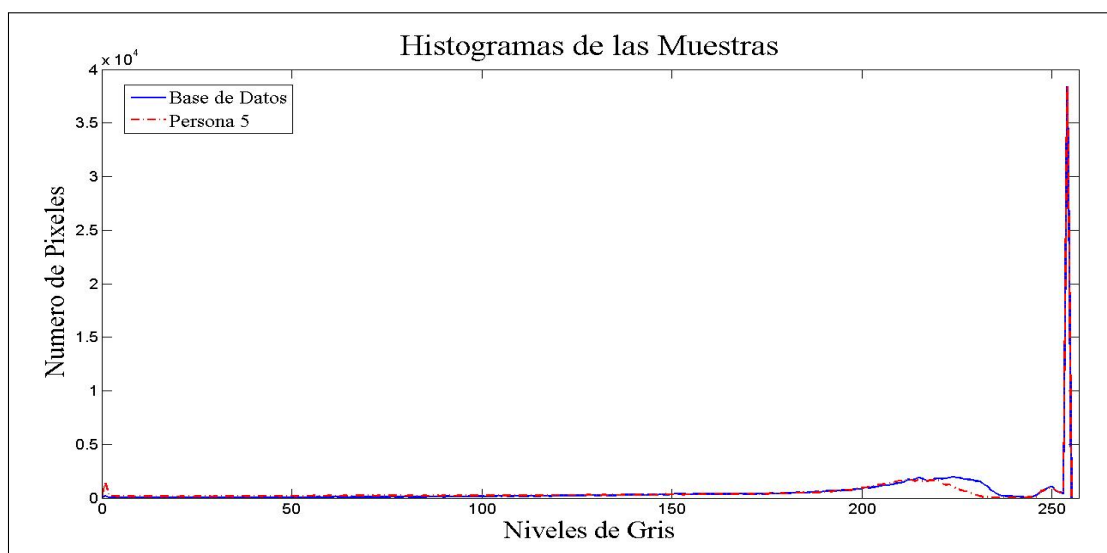
a)



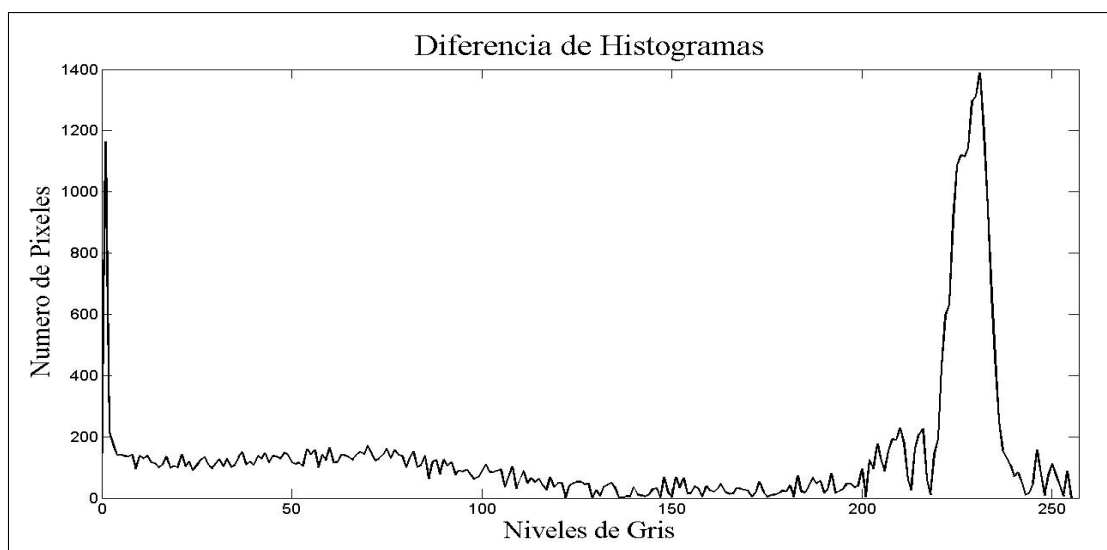
b)

Figura 6.10: Comparación de histograma para la persona 4 del primer conjunto versus el histograma de la base de datos: a) muestras a comparar y b) diferencia.

Comentario: En la figura 6.10 a) se muestra el histograma que se encuentra en la base de datos, también el histograma de la imagen de la huella de la persona 4 del primer conjunto de muestras y en la figura 6.10 b) la diferencia de los histogramas mostrados en la figura 6.10 a). Se puede observar que en la figura 6.10 b) el número de pixeles en los niveles de gris sobrepasan el valor de los 450 e incluso algunos valores están cercanos a los 1 500.



a)



b)

Figura 6.11: Comparación de histograma para la persona 5 del primer conjunto versus el histograma de la base de datos: a) muestras a comparar y b) diferencia.

Comentario: En la figura 6.11 a) se muestra el histograma que se encuentra en la base de datos, también el histograma de la imagen de la huella de la persona 5 del primer conjunto de muestras y en la figura 6.11 b) la diferencia de los histogramas mostrados en la figura 6.11 a). Se puede observar que en la figura 6.11 b) el número de pixeles en los niveles de gris sobrepasan el valor de los 450 e incluso algunos valores están cercanos a los 1 400.

Conclusión Parcial: Después de observar los resultados que se muestran en la tabla 6.1 y tomando en cuenta el hecho que la huella de la persona 1, así como la huella que

se encuentra en la base de datos pertenecen al pulgar izquierdo del mismo individuo. Se podría tomar la decisión de seleccionar un umbral para discriminar a las personas que se aceptarán o se rechazarán en la comparación. En este caso, el umbral que se selecciona es de 20 000. Debido a que se puede notar que el resultado de la comparación de la persona 1 del primer conjunto versus el usuario registrado está debajo del valor de umbral seleccionado, siendo este valor preliminar, por tal motivo tendrá que ser verificado con algunas pruebas adicionales como se muestran en la siguiente sección.

Etapa de verificación del umbral

El objetivo que persigue la etapa de verificación es comprobar que el umbral seleccionado en la etapa anterior, es adecuado para tomar la decisión correcta a la hora de discriminar a las personas que no se encuentran registradas en la base de datos.

Lo primero que se realizó en la etapa de verificación fue tomar cinco nuevas imágenes de huellas dactilares por el sensor, de las cuales, una pertenece a la persona registrada en la base de datos y el resto corresponden a personas distintas a las muestras utilizadas en la prueba anterior. En la figura 6.12 se observan las imágenes de huellas adquiridas, siendo este el segundo conjunto de muestras.

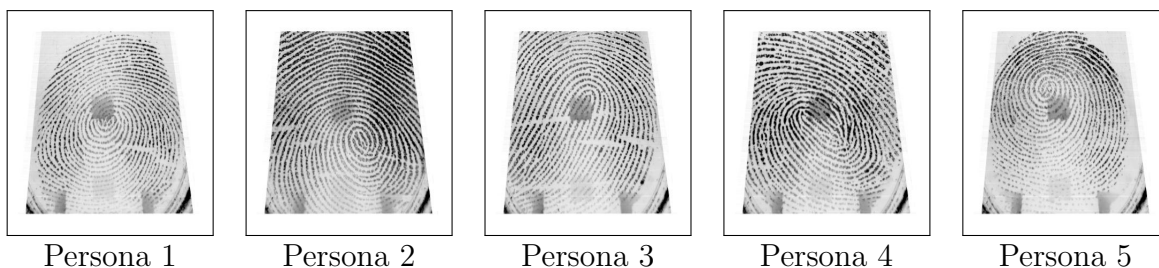


Figura 6.12: Imágenes del segundo conjunto de cinco huellas de personas que solicitan acceso.

Comentario: Las imágenes de las huellas de las cinco personas que se muestran en la figura 6.12 son de individuos distintos a los de la etapa de calibración, esto sin considerar a uno de ellos, el cual, es la persona 1 que en este caso, sigue siendo el mismo individuo que se encuentra registrado en la base de datos. También las muestras se tomaron en las condiciones más similares, esto se realizó limpiando el lector en cada lectura de muestra, limpiando el dedo de la persona a quien se le tomaba la muestra, colocando el dedo de la persona en la misma posición para todos los individuos y teniendo la misma iluminación para todas las tomas de muestras.

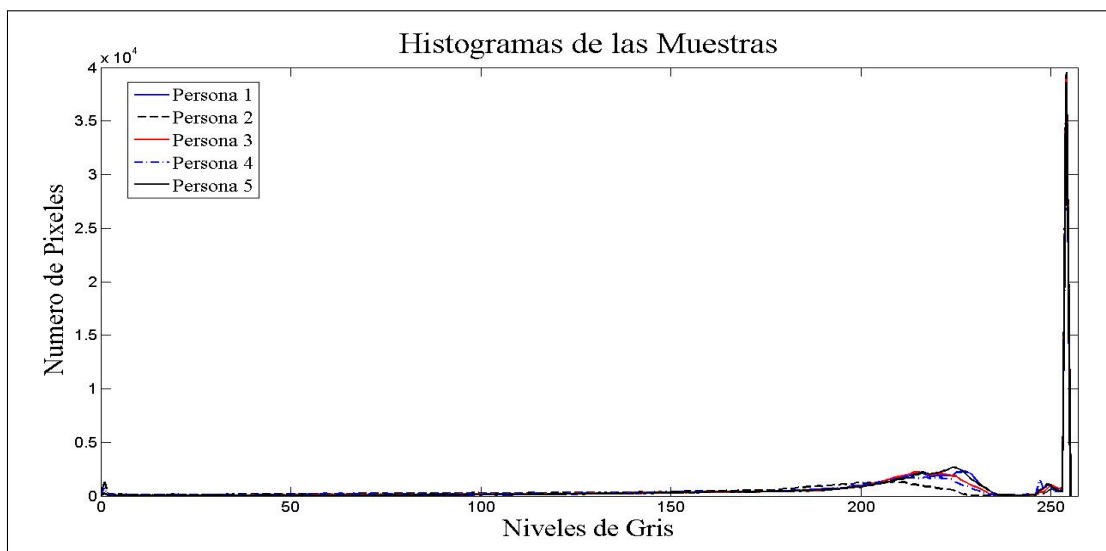


Figura 6.13: Histogramas de las muestras de cinco personas adicionales para la etapa de verificación.

Comentario: Al igual que la imagen de los histogramas que se mostraron en la figura 6.5, en la imagen de los histogramas de la figura 6.13, se observa que la distribución en los niveles de gris de las huellas son muy parecidas. Sin embargo, tienen diferencias en algunos niveles de gris, lo cual, nos permitirá realizar la comparación.

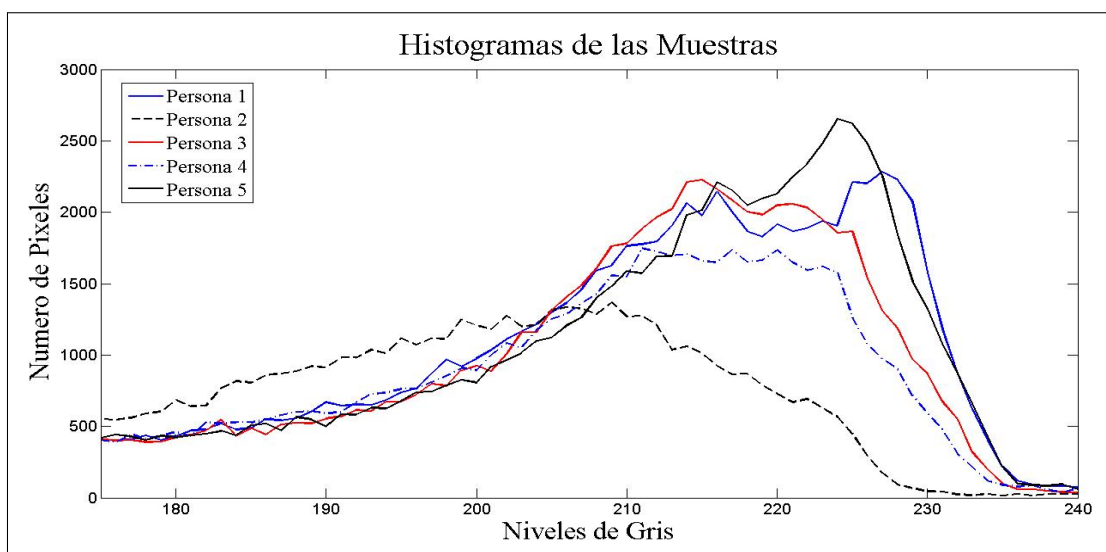


Figura 6.14: Sección ampliada de la gráfica de los histogramas del segundo conjunto de cinco muestras de personas.

Debido a que en la figura 6.13 no se aprecian las diferencias de los histogramas adecuadamente. Se decidió mostrar una imagen aumentando una sección de la gráfica ver la figura 6.14, en la cual, es más sencillo apreciar las diferencias.

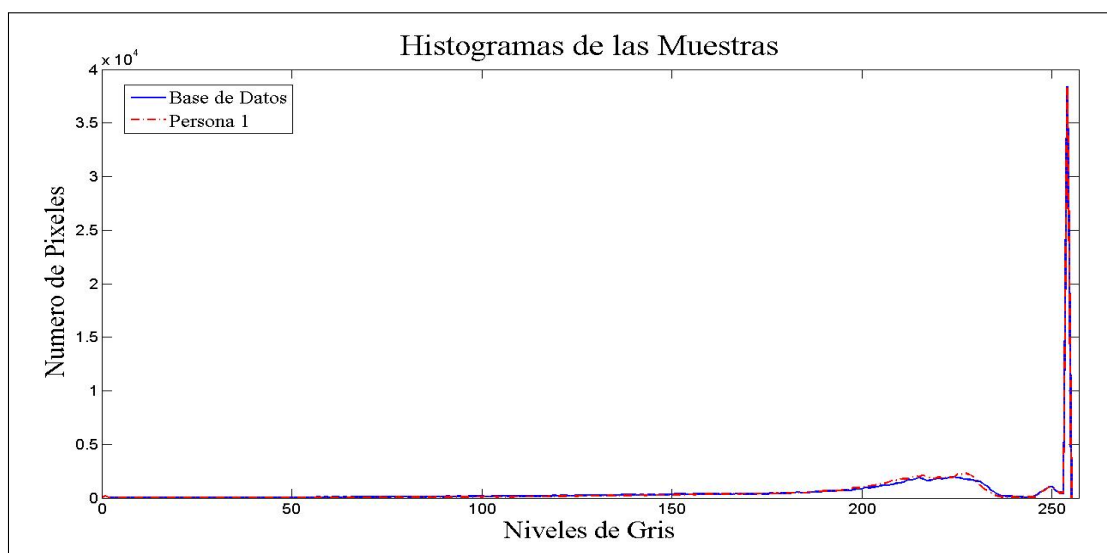
Al igual que en la etapa de calibración después de mostrar las imágenes utilizadas en la prueba, lo que sigue es presentar en la tabla 6.2 los resultados obtenidos de la comparación de los histogramas del segundo conjunto de muestras versus el histograma del usuario registrado en la base de datos.

Persona	<i>Resultado_C</i>
1	15814
2	59324
3	19430
4	26196
5	16628

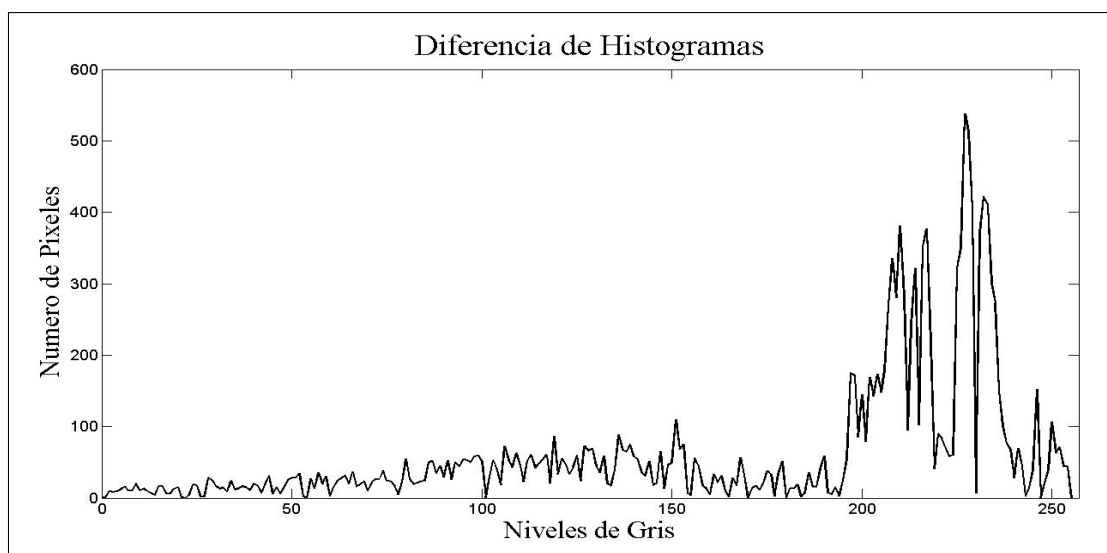
Tabla 6.2: Resultados de la diferencia de histogramas de las cinco personas adicionales, para la etapa de verificación.

Comentario: En la etapa de calibración, se seleccionó un umbral de discriminación tentativo de 20 000. Pero observando los resultados obtenidos en la tabla 6.2 se puede ver que hay 3 personas que se encuentran debajo de este valor de umbral, siendo estas la persona 1, la persona 3 y la persona 5. Por tanto, se tendrá que considerar la posibilidad de modificar el valor del umbral seleccionado, con el fin de tener una mejor discriminación de personas no registradas en la base de datos.

Igualmente que en la prueba anterior para complementar los resultados mostrados en la tabla 6.2, a continuación en las siguientes figuras se muestran los histogramas de las muestras de cada una de las personas del segundo conjunto de muestras, el histograma de la base de datos y el histograma resultante de la comparación de cada una de las cinco muestras de las personas del segundo conjunto.



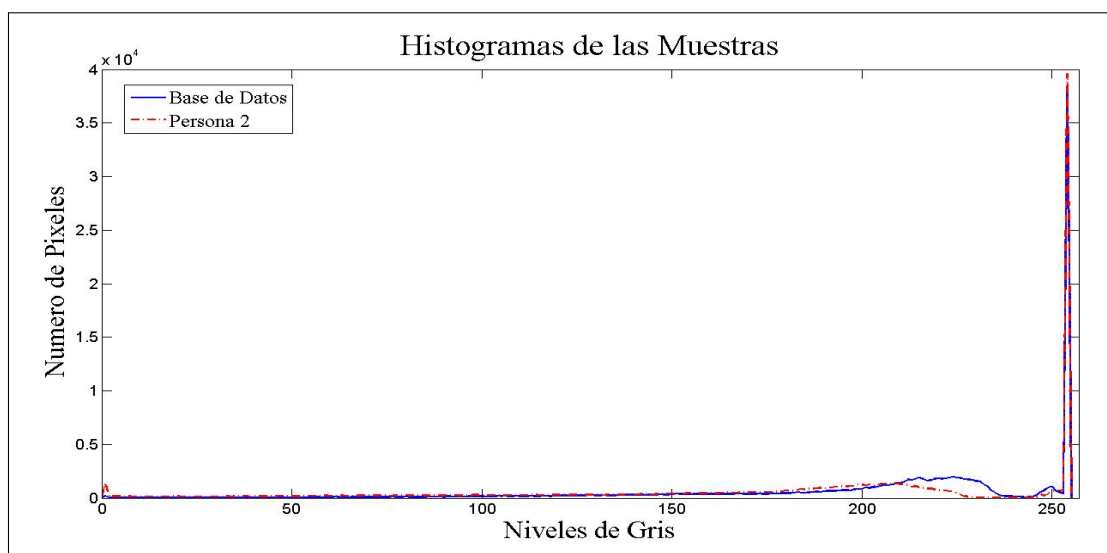
a)



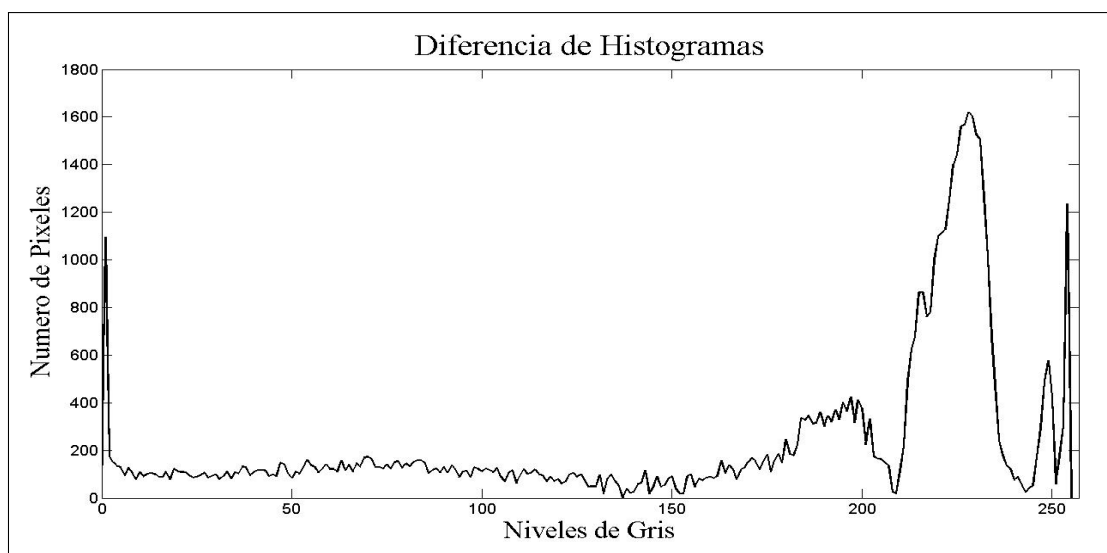
b)

Figura 6.15: Comparación de histograma para la persona 1 del segundo conjunto versus el histograma de la base de datos: a) muestras a comparar y b) diferencia.

Comentario: En la figura 6.15 a) se puede observar los histogramas que fueron comparados y en la figura 6.15 b) el resultado de esta comparación. Se puede observar que el número de píxeles en los niveles de gris no sobre pasan el valor de los 600.



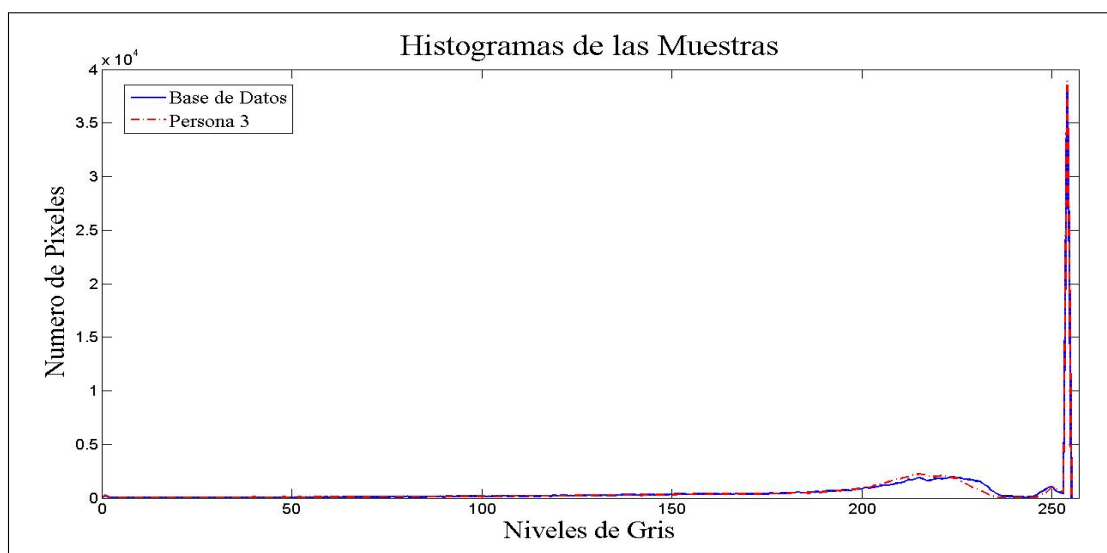
a)



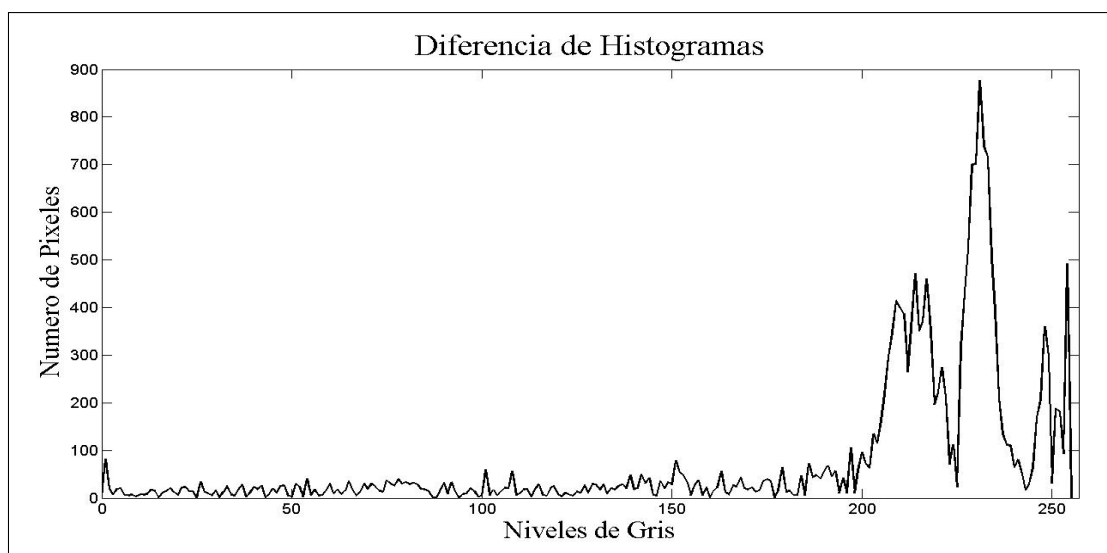
b)

Figura 6.16: Comparación de histograma para la persona 2 del segundo conjunto versus el histograma de la base de datos: a) muestras a comparar y b) diferencia.

Comentario: En la figura 6.16 a) se puede observar los histogramas que fueron comparados y en la figura 6.16 b) el resultado de esta comparación. Se puede observar que el número de pixeles en los niveles de gris están cercanos a los 1 600.



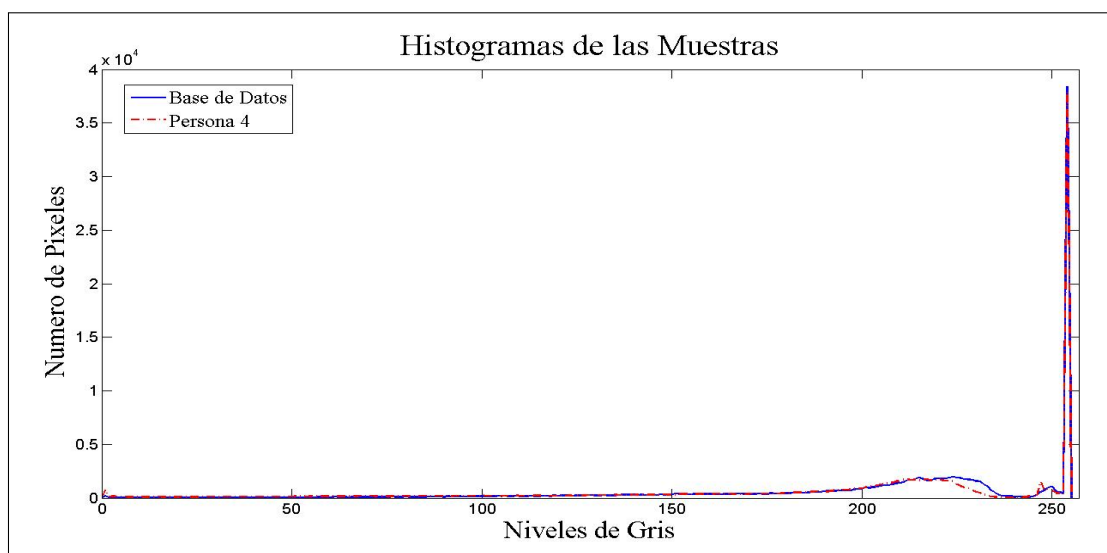
a)



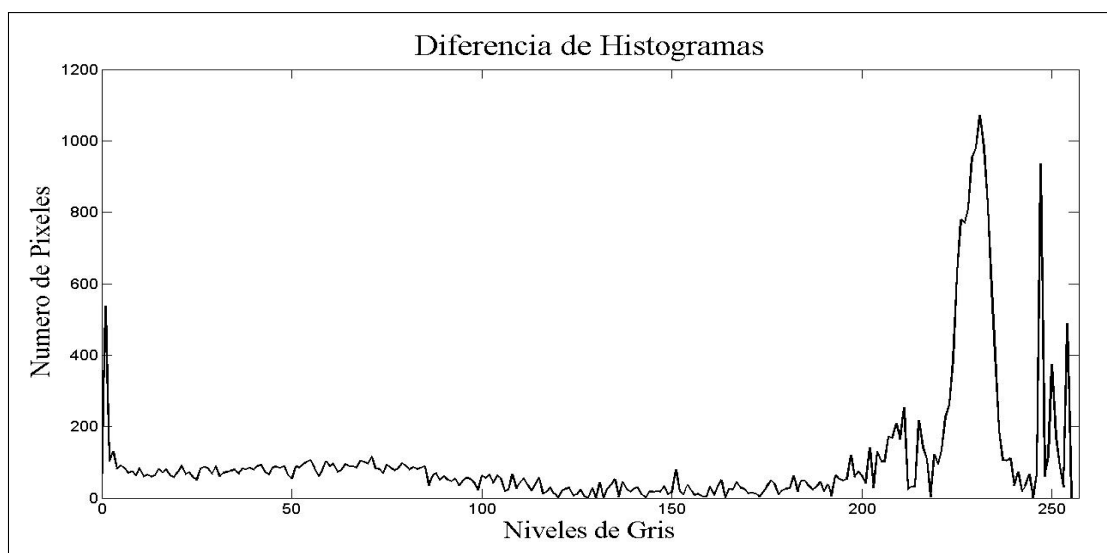
b)

Figura 6.17: Comparación de histograma para la persona 3 del segundo conjunto versus el histograma de la base de datos: a) muestras a comparar y b) diferencia.

Comentario: En la figura 6.17 a) se puede observar los histogramas que fueron comparados y en la figura 6.17 b) el resultado de esta comparación. Se puede observar que el número de pixeles en los niveles de gris están cercanos a los 900.



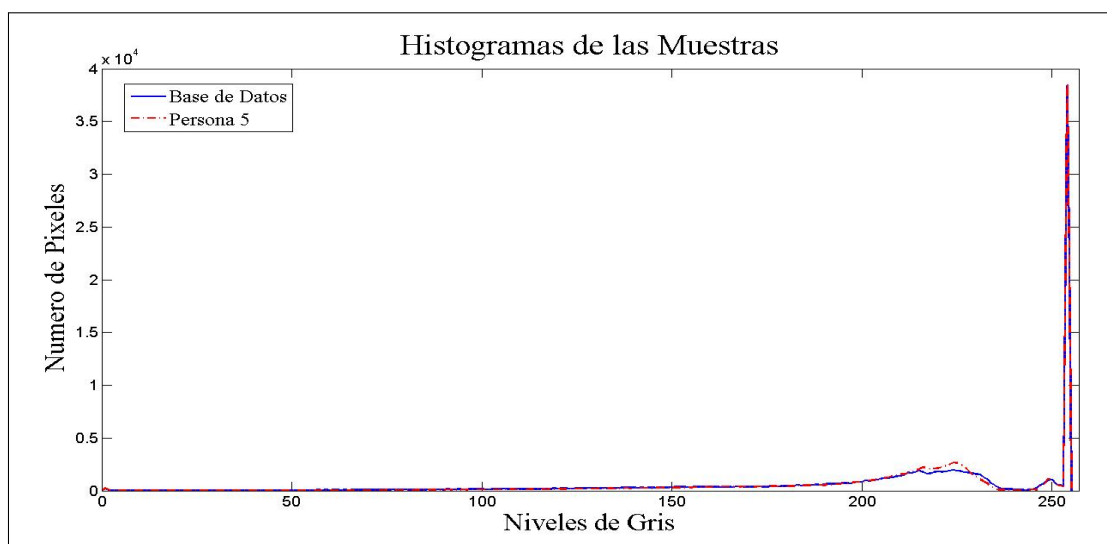
a)



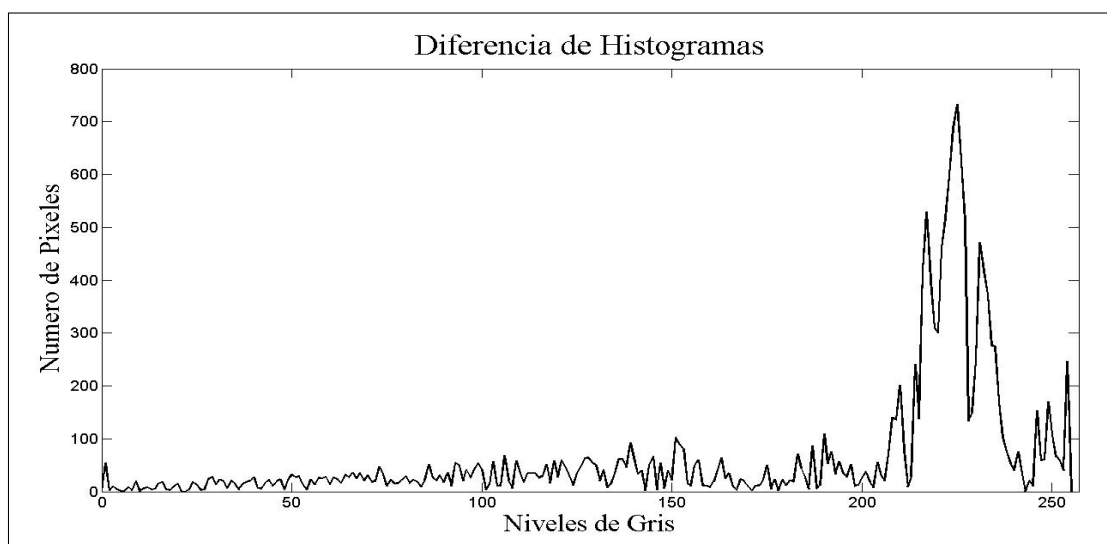
b)

Figura 6.18: Comparación de histograma para la persona 4 del segundo conjunto versus el histograma de la base de datos: a) muestras a comparar y b) diferencia.

Comentario: En la figura 6.18 a) se puede observar los histogramas que fueron comparados y en la figura 6.18 b) el resultado de esta comparación. Se puede observar que el número de pixeles en los niveles de gris están cercanos a los 1 100.



a)



b)

Figura 6.19: Comparación de histograma para la persona 5 del segundo conjunto versus el histograma de la base de datos: a) muestras a comparar y b) diferencia.

Comentario: En la figura 6.19 a) se puede observar los histogramas que fueron comparados y en la figura 6.19 b) el resultado de esta comparación. Se puede observar que el número de pixeles en los niveles de gris están cercanos a los 750.

Para ilustrar la información obtenida en esta prueba, se presenta en la figura 6.20 el despliegue gráfico (interfaz) obtenido al utilizar el programa en MatLab, donde se muestra a las personas aceptadas y rechazadas utilizando un umbral de discriminación de 20 000. Estos resultados se lograron agregando al código del programa del anexo A.1, la etapa de comparación (consultar código del anexo A.2).

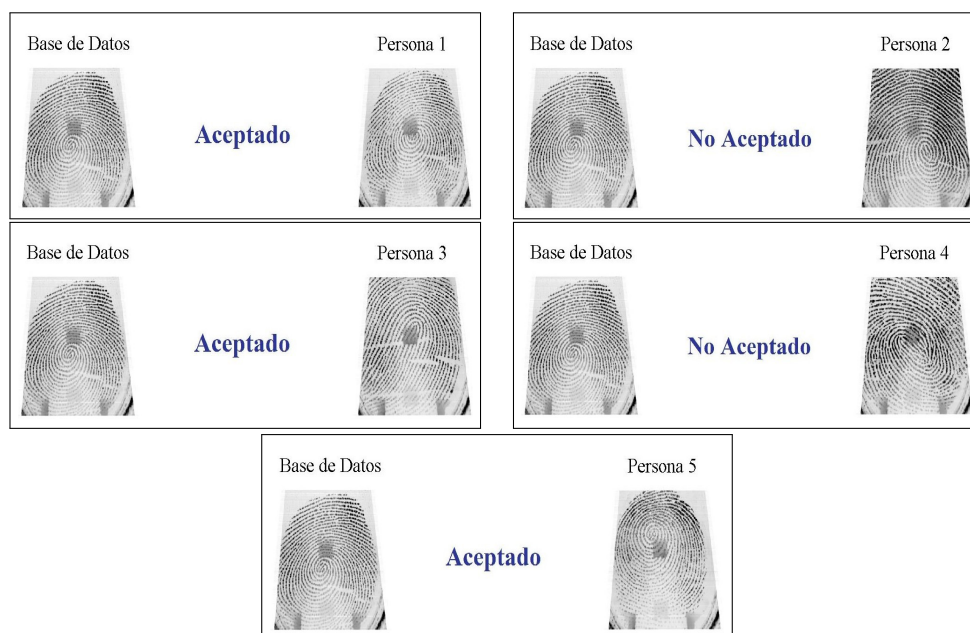


Figura 6.20: Resultados gráficos obtenidos del programa de comparación de histogramas implementado en MatLab para el conjunto 2.

Comentario: Con la ayuda de la figura 6.20 y la tabla 6.2 se puede observar que las personas aceptadas son las que se encuentran debajo del umbral de los 20 000; es decir, la persona 1, la persona 3 y la persona 5, todas los demás son rechazadas, debido a que se encuentran por encima de este valor de umbral.

Conclusión Parcial: Después de observar los resultados de la tabla 6.2, se puede ver que el umbral de 20 000 seleccionado no es adecuado, esto se puede saber observando la figura 6.20, en la cual, se puede observar que dos personas fueron aceptadas erróneamente, por tal motivo se recurre a cambiar el valor de este umbral, tomando en cuenta que la huella de la persona 1 y la huella que se encuentra registrada en la base de datos pertenecen al pulgar izquierdo del mismo individuo, se puede tomar la decisión de seleccionar un nuevo umbral para discriminar a las personas no autorizadas, siendo este nuevo umbral de 16 000. Debido a que se puede notar en la tabla 6.2 que el resultado de la comparación de la persona 1 está debajo del umbral mientras que los resultados de las demás comparaciones de personas se encuentran por encima de este umbral seleccionado.

Después de observar los resultados no alentadores que se obtuvieron en la prueba anterior, se optó por cambiar el umbral de discriminación a 16 000 y con el motivo de comprobar que este umbral pueda generar mejores resultados de comparación. Lo siguiente es realizar la prueba de comparación nuevamente, sin embargo para obtener resultados confiables de la comparación de histogramas, después de cambiar el umbral de discriminación, se optó por adquirir cinco nuevas imágenes de huellas dactilares del pulgar izquierdo. En la figura 6.21 se muestran las imágenes adquiridas, también en la figura 6.22 se pueden observar los histogramas de estas imágenes, siendo este el tercer conjunto de muestras.

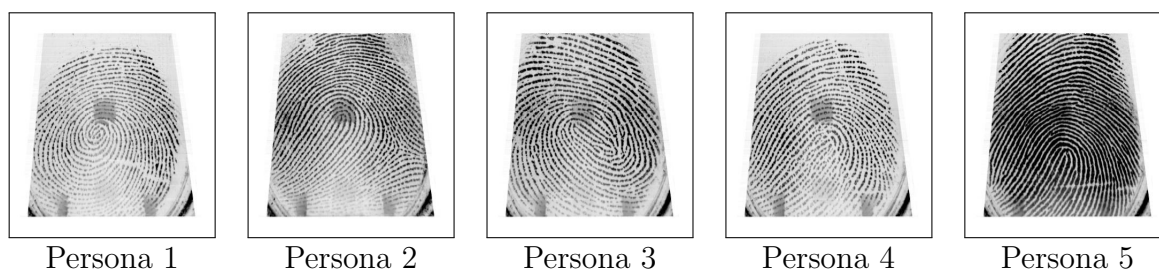


Figura 6.21: Imágenes de las muestras de cinco personas del tercer conjunto.

Comentario: Las imágenes de las huellas de las personas que se muestran en la figura 6.21 son de individuos distintos a los que se usaron en las pruebas anteriores, esto sin considerar a uno de ellos, el cual, es la persona 1 que en este caso sigue siendo la misma persona registrada en la base de datos. También las muestras se tomaron en las condiciones más similares posibles, esto se realizó limpiando el lector en cada lectura de muestra, limpiando el dedo de la persona a quien se le tomaba la muestra, colocando el dedo de la persona en la misma posición para todos los individuos y teniendo la misma iluminación para todas la tomas de muestras.

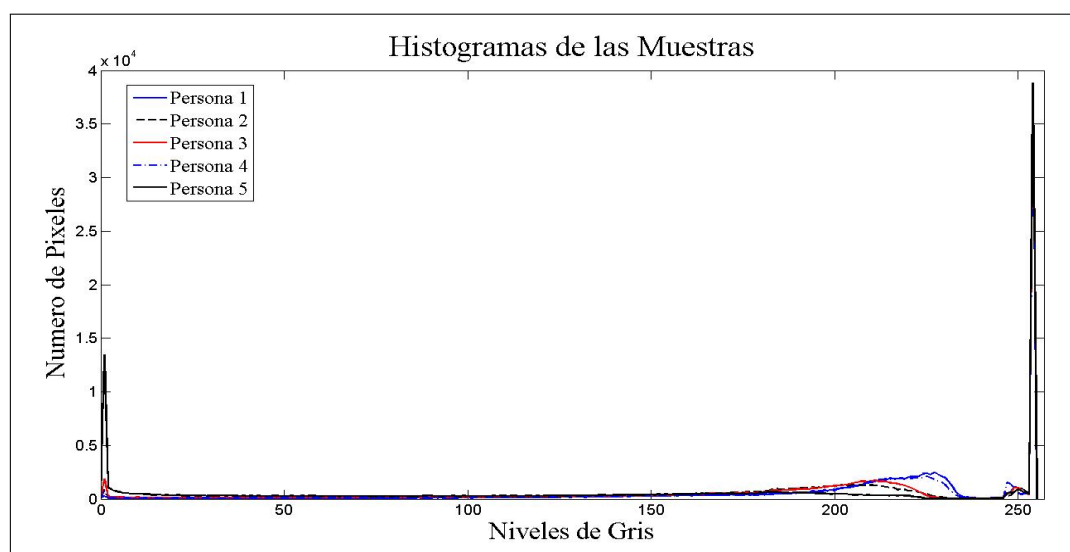


Figura 6.22: Histogramas de las muestras de cinco personas del tercer conjunto.

Comentario: En la figura 6.22 se puede observar que la distribución de los niveles de grises de los histogramas de las imágenes de las huellas son muy parecidas. Sin embargo, tienen diferencias en algunos niveles de gris, al igual que en los casos mostrados anteriormente en las figuras 6.5 y 6.13, por lo que en esta ocasión, también nos permitirá realizar las comparaciones.

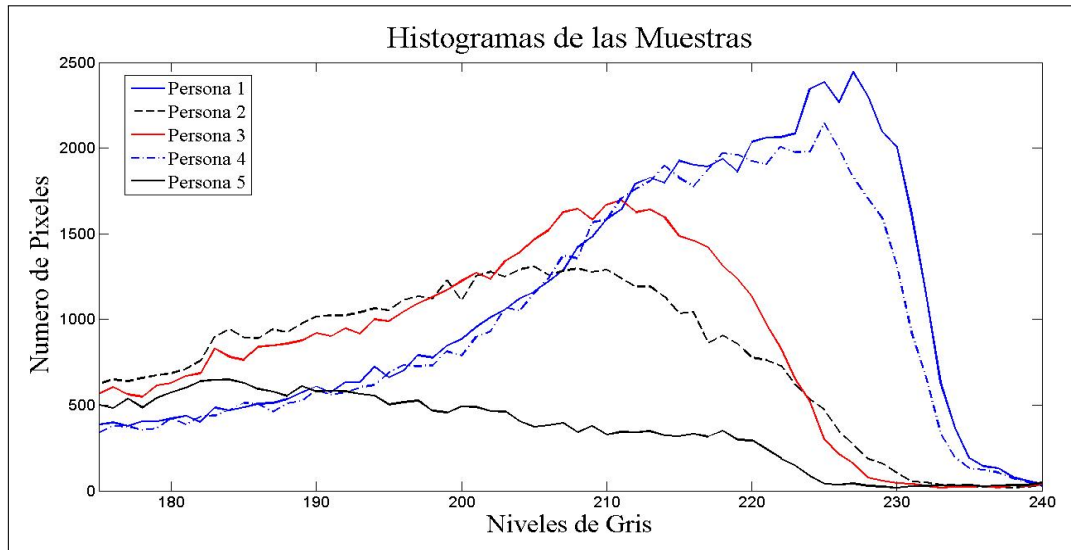


Figura 6.23: Sección ampliada de la gráfica de los histogramas del tercer conjunto de cinco de personas.

Debido a que en la figura 6.22 no se aprecian las diferencias de los histogramas adecuadamente. Se decidió mostrar una imagen aumentando una sección de la gráfica ver la figura 6.23, en la cual, es más sencillo apreciar las diferencias.

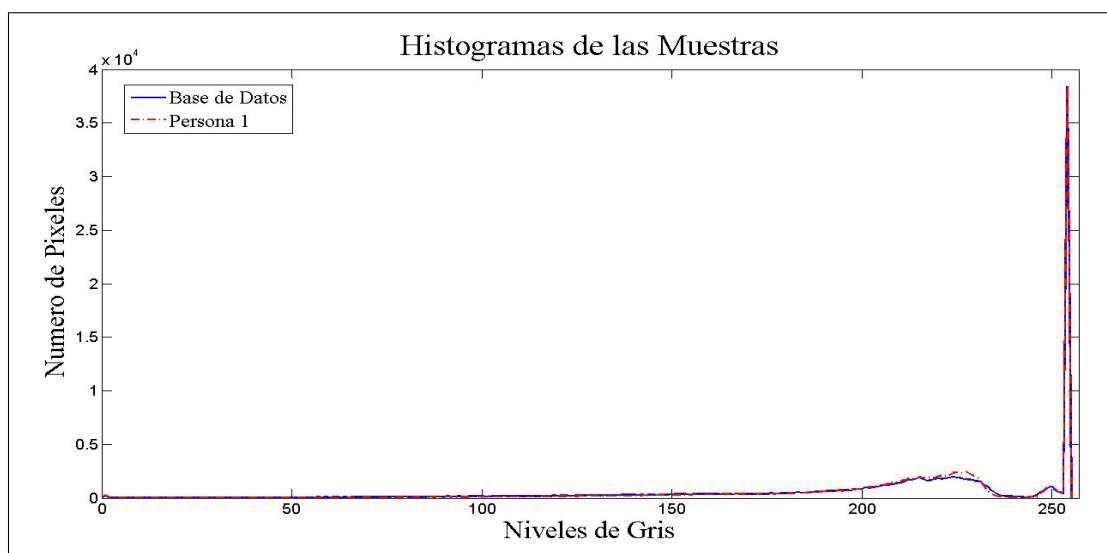
En la tabla 6.3 se muestran los resultados obtenidos del programa realizado en MatLab (ver anexo A.1), para la comparación de histogramas. En esta ocasión utilizando las imágenes del tercer conjunto de muestras.

Persona	<i>Resultado_C</i>
1	14596
2	55942
3	47056
4	18228
5	93920

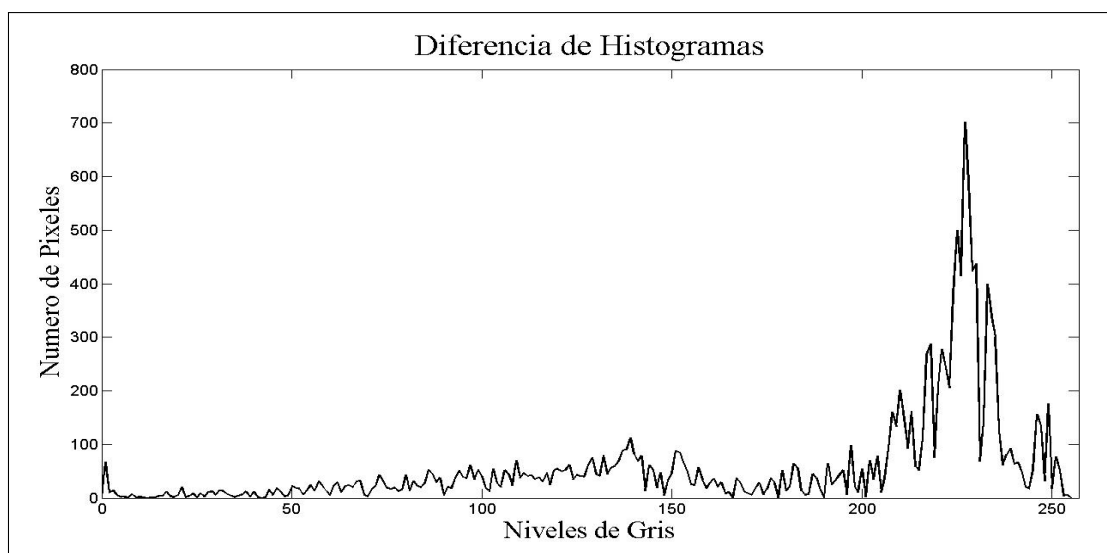
Tabla 6.3: Resultados de la diferencia de histogramas del tercer conjunto de muestras ingresadas al sistema biométrico.

Comentario: En esta tabla se puede observar que la diferencia es mayor de los 16 000 para las huellas que son distintas a la que se encuentra en la base de datos, con excepción de la huella de la persona 1 que corresponde a la huella que se encuentra en la base de datos.

Para complementar la información de la tabla 6.3, se mostrarán en las siguientes figuras los histograma de las imágenes de las huellas de ingreso, el histograma de la base de datos y el histograma resultante de la comparación.



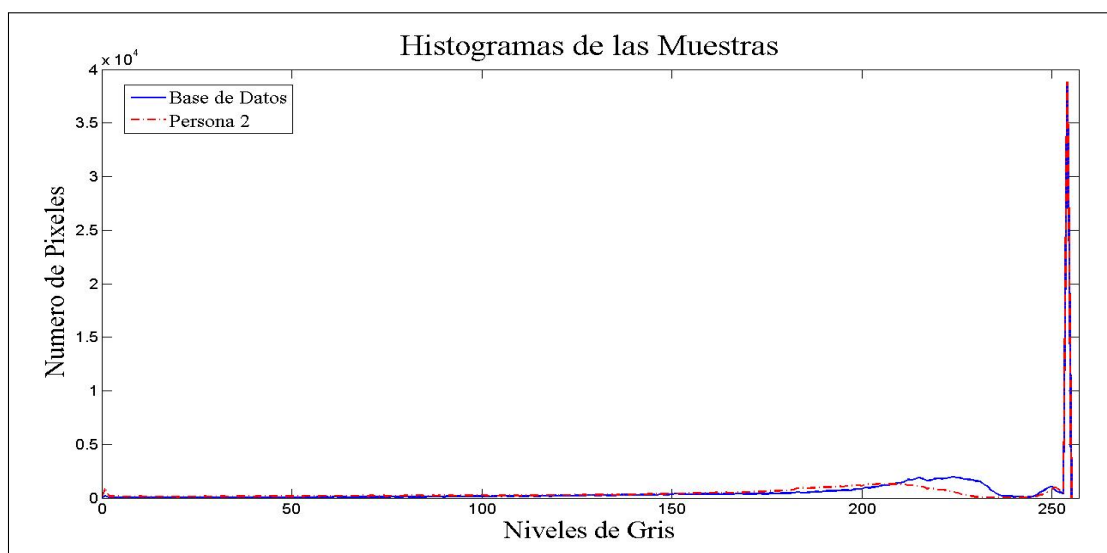
a)



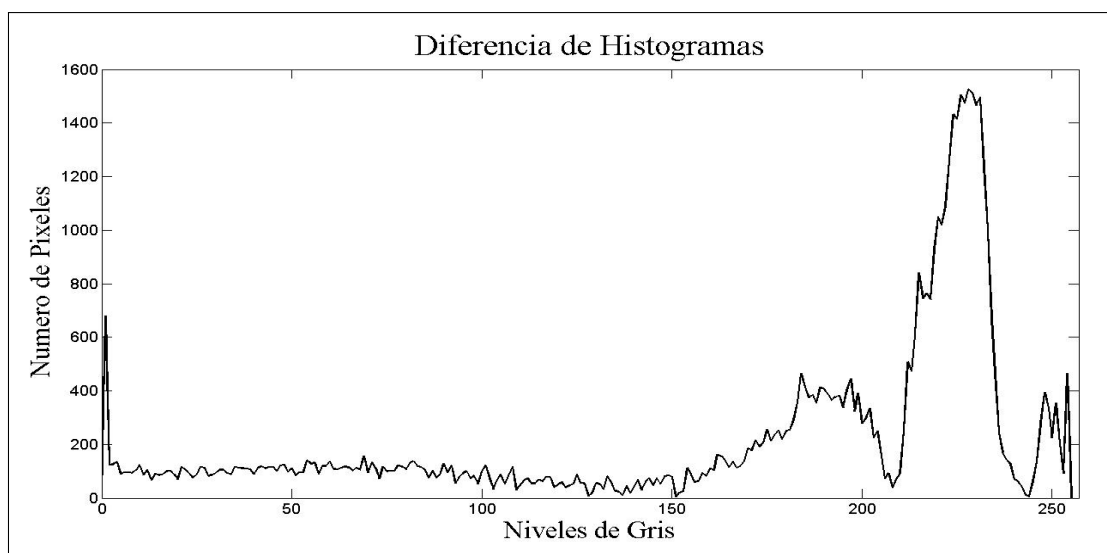
b)

Figura 6.24: Comparación de histograma para la persona 1 del tercer conjunto versus el histograma de la base de datos: a) muestras a comparar y b) diferencia.

Comentario: En la figura 6.24 a) se pueden observar los histogramas que serán comparados mientras que en la figura 6.24 b) el resultado de la comparación de los histogramas de la figura 6.24 a). Se puede ver que el número de píxeles en los niveles de gris no sobrepasan el valor de los 750.



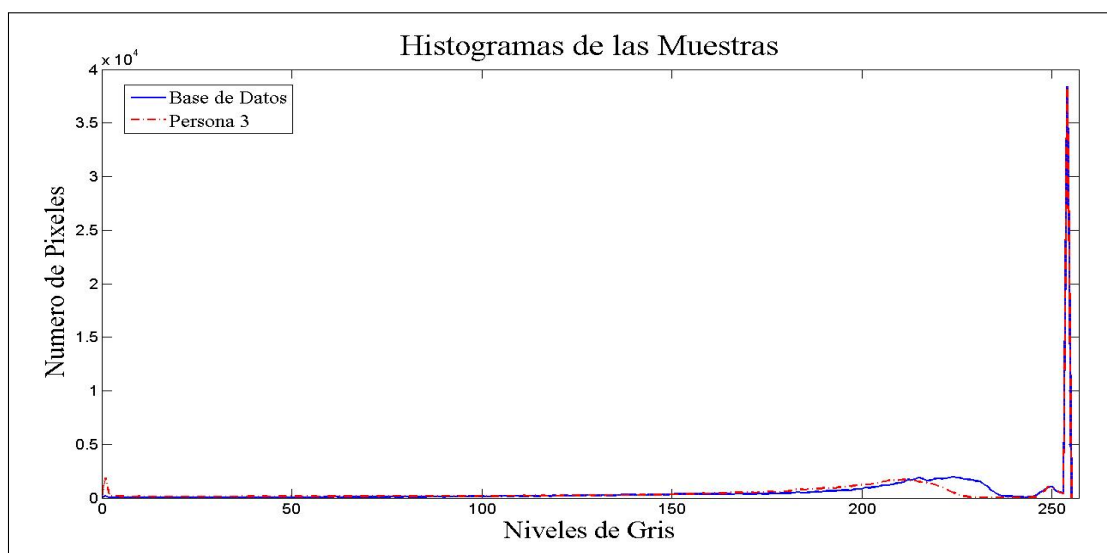
a)



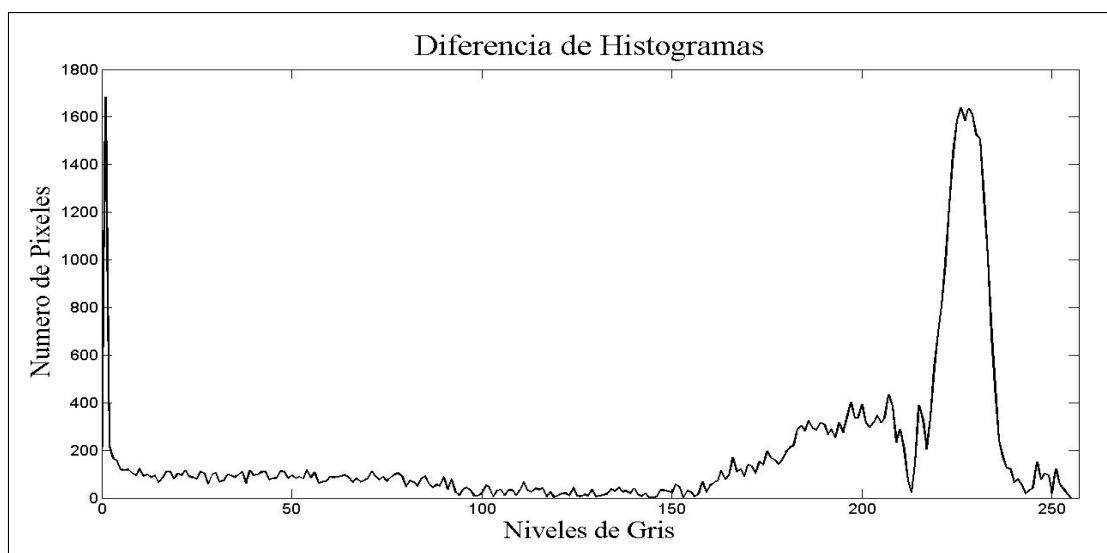
b)

Figura 6.25: Comparación de histograma para la persona 2 del tercer conjunto versus el histograma de la base de datos: a) muestras a comparar y b) diferencia.

Comentario: En la figura 6.25 a) se pueden observar los histogramas que serán comparados mientras que en la figura 6.25 b) el resultado de la comparación de los histogramas de la figura 6.25 a). Se puede ver que el número de píxeles en algunos de los niveles de gris sobrepasan el valor de los 750 e incluso algunos valores están cercanos a los 1 500.



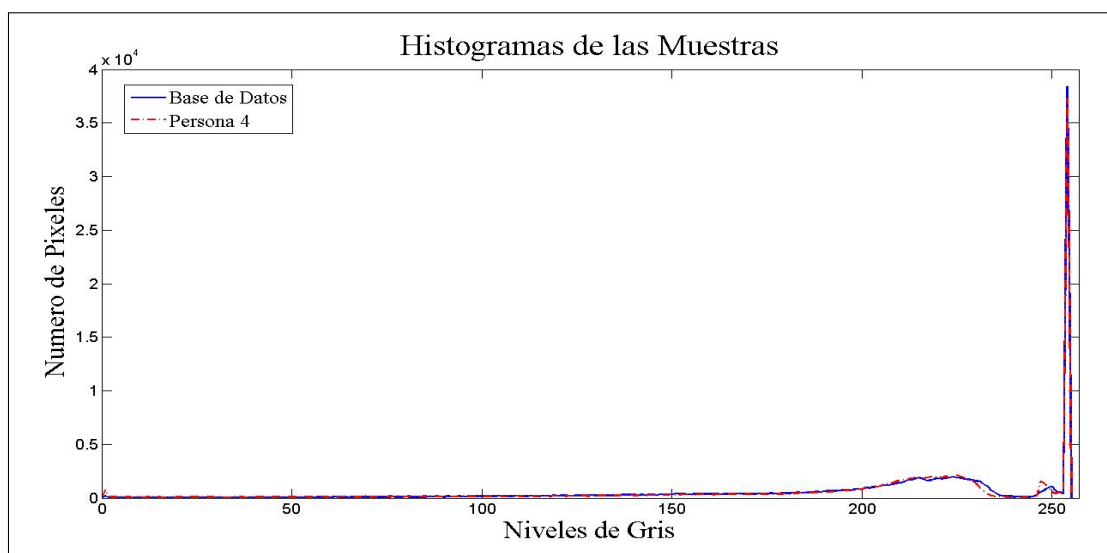
a)



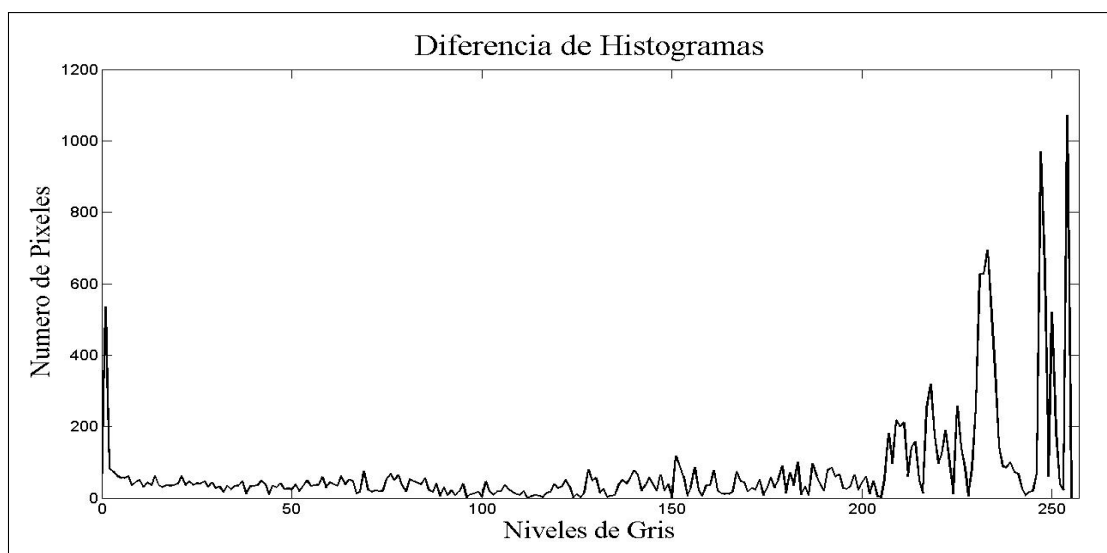
b)

Figura 6.26: Comparación de histograma para la persona 3 del tercer conjunto versus el histograma de la base de datos: a) muestras a comparar y b) diferencia.

Comentario: En la figura 6.26 a) se pueden observar los histogramas que serán comparados mientras que en la figura 6.26 b) el resultado de la comparación de los histogramas de la figura 6.26 a). Se puede ver que el número de pixeles en en algunos de los niveles de gris sobrepasan el valor de los 750 e incluso algunos valores están cercanos a los 1 600.



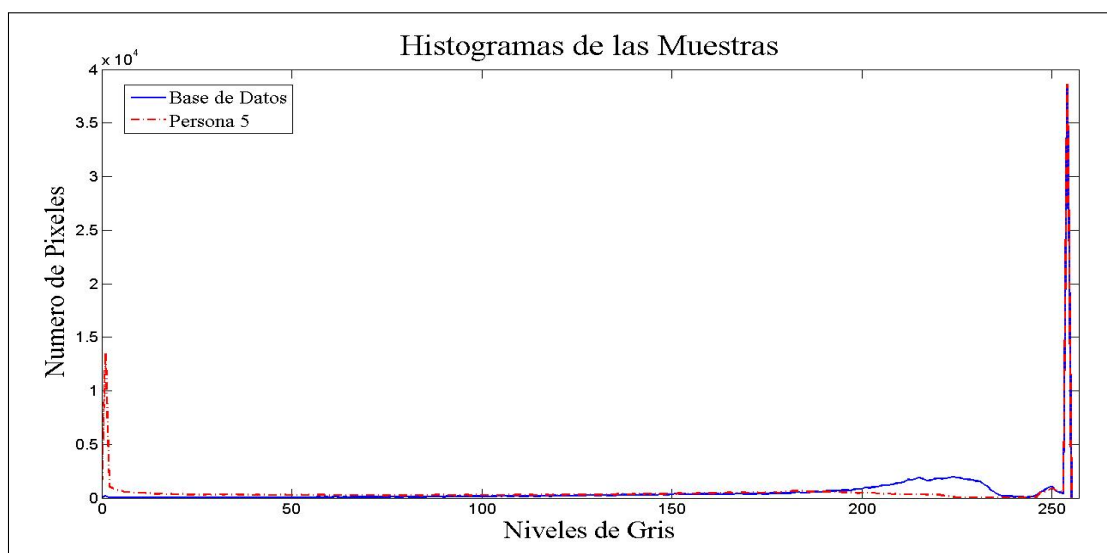
a)



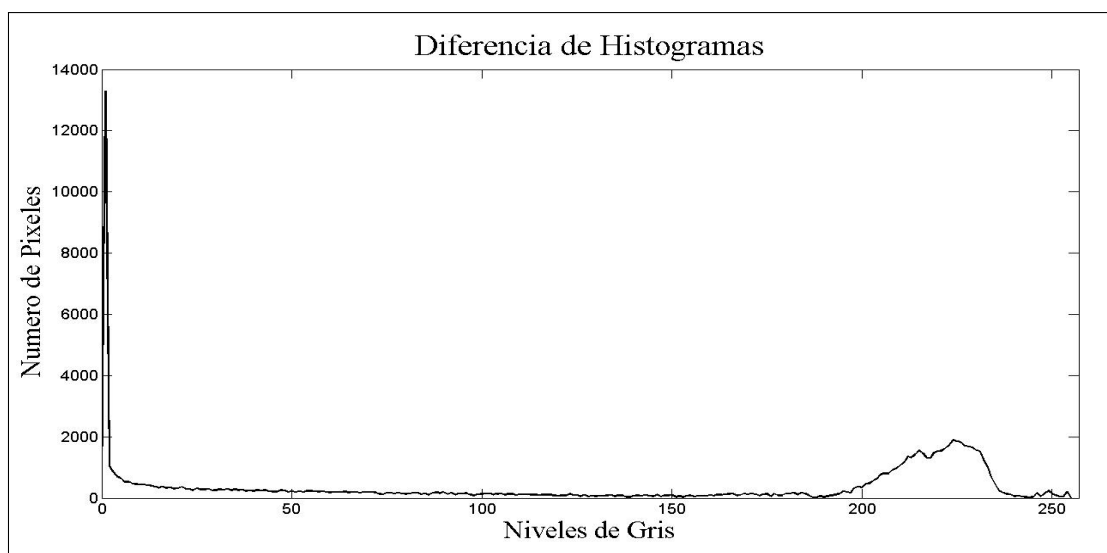
b)

Figura 6.27: Comparación de histograma para la persona 4 del tercer conjunto versus el histograma de la base de datos: a) muestras a comparar y b) diferencia.

Comentario: En la figura 6.27 a) se pueden observar los histogramas que serán comparados mientras que en la figura 6.27 b) el resultado de la comparación de los histogramas de la figura 6.27 a). Se puede ver que el número de pixeles en en algunos de los niveles de gris sobrepasan el valor de los 750 e incluso algunos valores están cercanos a los 1 100.



a)



b)

Figura 6.28: Comparación de histograma para la persona 5 del tercer conjunto versus el histograma de la base de datos: a) muestras a comparar y b) diferencia.

Comentario: En la figura 6.28 a) se pueden observar los histogramas que serán comparados mientras que en la figura 6.28 b) el resultado de la comparación de los histogramas de la figura 6.28 a). Se puede ver que el número de pixeles en algunos de los niveles de gris sobrepasan el valor de los 750 e incluso algunos valores están cercanos a los 2 000.

En la figura 6.29 se ilustran los resultados gráficos obtenidos del programa de comparación de histogramas implementado en MatLab (ver anexo A), en el cual, se ha utilizado el nuevo valor del umbral de discriminación de 16 000 para este último grupo de pruebas.

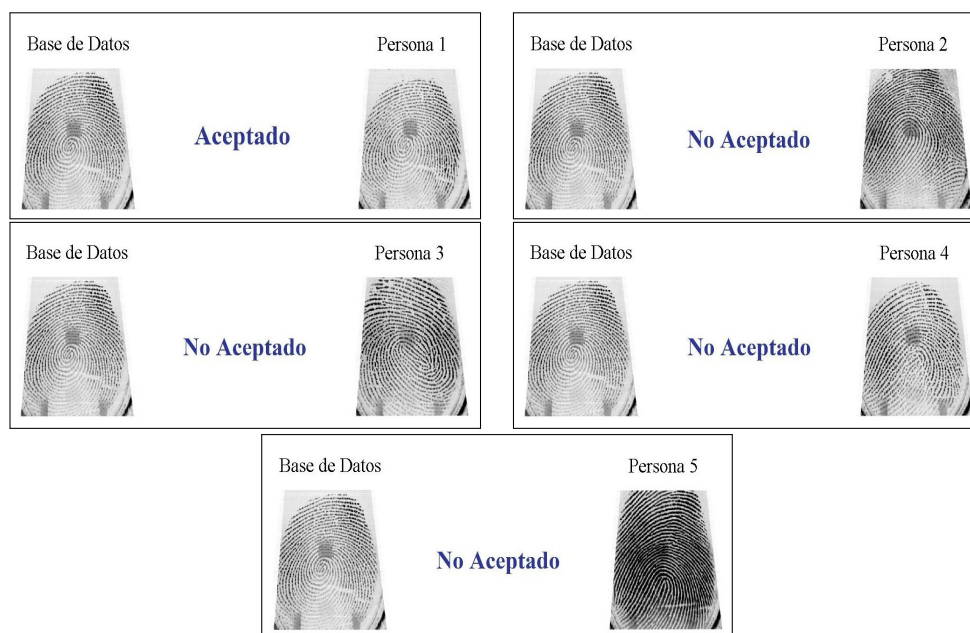


Figura 6.29: Resultados obtenidos de la comparación de histogramas en MatLab para el conjunto 3 de muestras de entrada al sistema biométrico.

Comentario: En la figura 6.29 se puede observar que el único individuo aceptado es la persona 1, siendo la diferencia de éste, menor al umbral seleccionado; por tal motivo fue aceptado y todos los demás individuos que su diferencia fue mayor a 16 000 fueron correctamente rechazados.

Conclusión Parcial: Después de observar los resultados que se muestran en la tabla 6.3 se puede ver que el umbral seleccionado (16 000) funciona para discriminar a las personas que no se encuentran registradas en la base de datos. Esto indica que las pruebas realizadas con el umbral seleccionado son aceptables para realizar una buena comparación de un sólo usuario.

Etapa de confirmación

El propósito de la etapa de confirmación es corroborar que el umbral de 16 000, seleccionado en la etapa de verificación, es el adecuado para realizar la discriminación de usuarios no registrados. En esta ocasión se realizó sólo una prueba, para realizarla se consideró la imagen de la huella registrada en la base de datos, así como las imágenes de las huellas mostradas en la etapa de calibración y verificación. A continuación en la figura 6.30 se muestran las huellas de las etapas anteriores pero renombrando alguna de ellas para una mejor visualización en esta etapa.

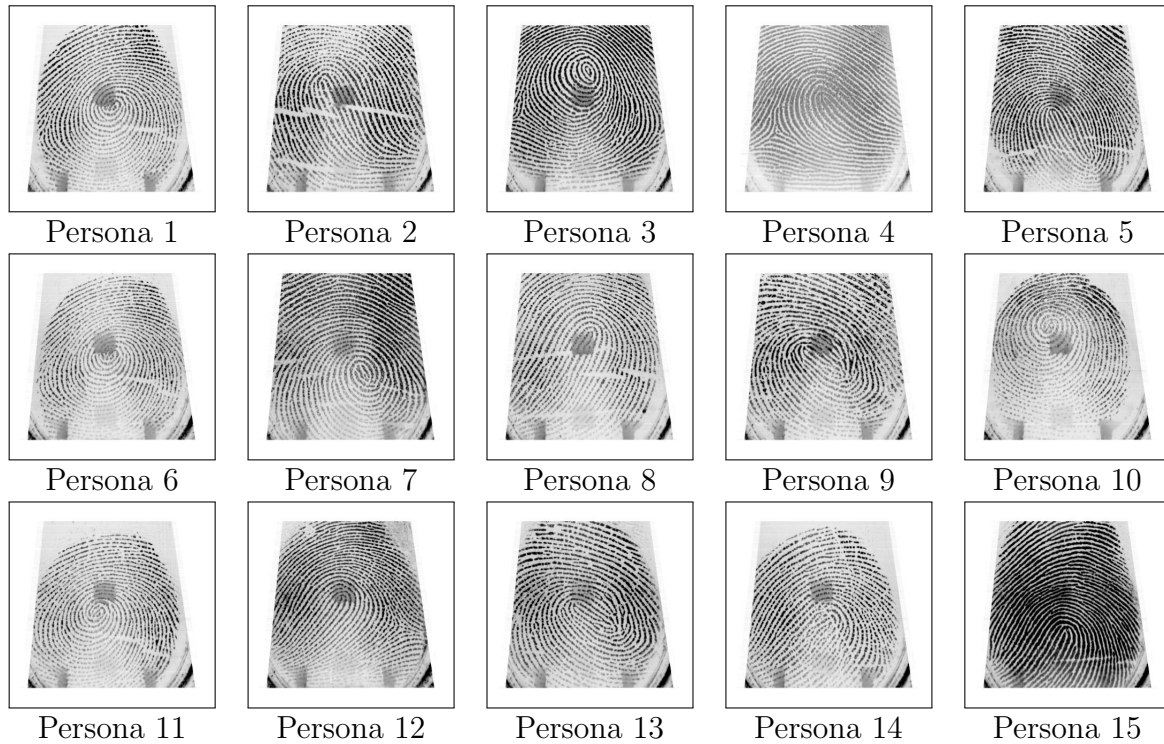


Figura 6.30: Imágenes de las muestras renombradas de las personas de las etapas de calibración y verificación.

Comentario: En la figura 6.30 se puede observar que las huellas de las personas 1, 6 y 11, son similares, esto se debe a que las tres huellas pertenecen a la imagen del pulgar izquierdo del usuario registrado en la base de datos. Los histogramas de cada una de las imágenes ya fueron mostradas por ese motivo no se exhibirán en esta etapa.

En la tabla 6.4 se muestran los resultados numéricos obtenidos de la comparación de los histogramas de los tres conjuntos de muestras versus el histograma de la base de datos, siendo un total de 15 comparaciones.

Persona	$Resultado_C$		Persona	$Resultado_C$		Persona	$Resultado_C$
1	14486		6	15814		11	14596
2	24046		7	59324		12	55942
3	46958		8	19430		13	47056
4	52990		9	26196		14	18228
5	35442		10	16628		15	93920

Tabla 6.4: Resultados de la diferencia de los 15 histogramas versus histograma de la base de datos.

En la figura 6.31 se muestran los resultados gráficos obtenidos del programa de comparación de histogramas implementado en MatLab (ver anexo A), en el cual, se ha utilizado el valor de umbral de discriminación de 16 000.

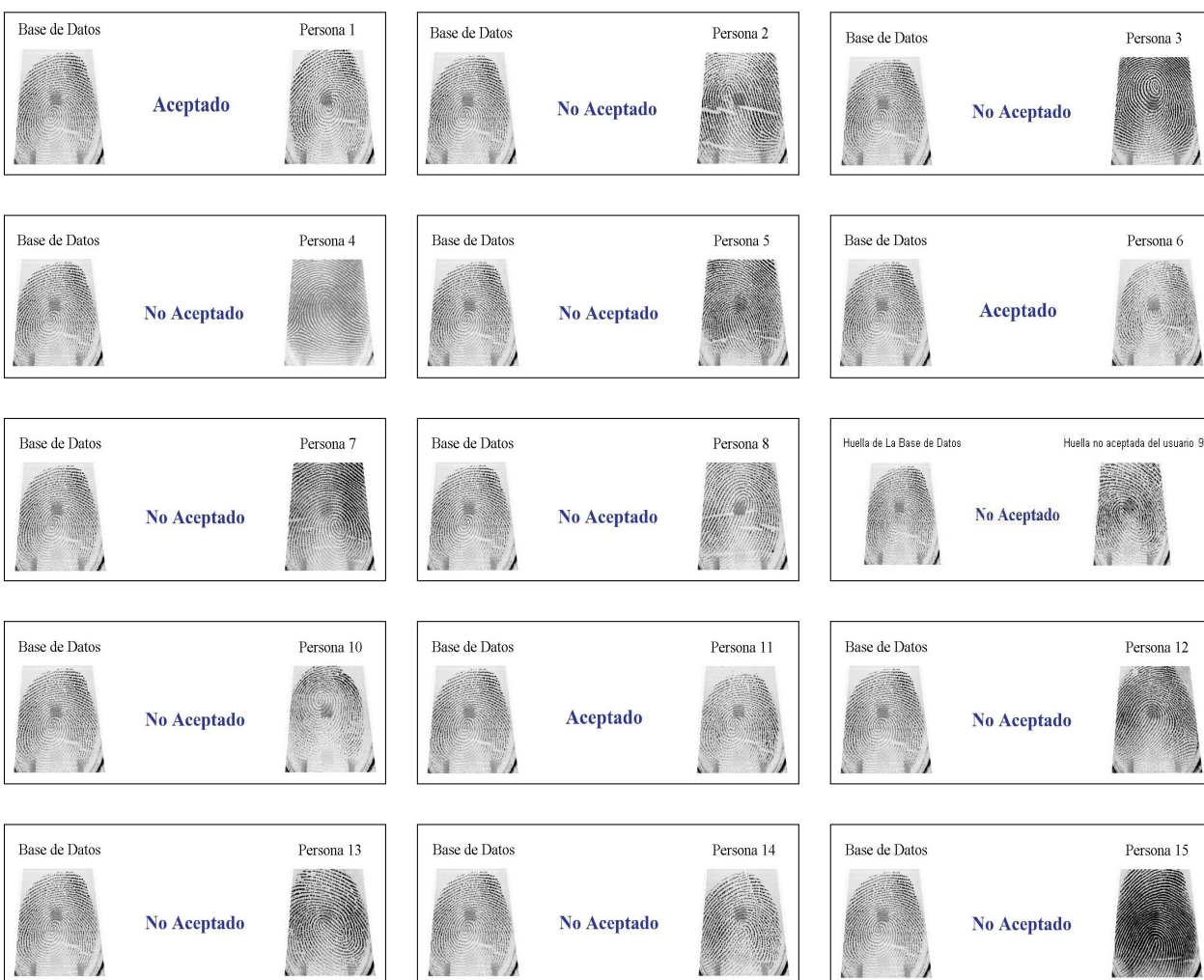


Figura 6.31: Resultados gráficos de MatLab obtenidos de la comparación del histograma de la huella registrada versus los histogramas de los tres conjuntos de muestras.

En la figura 6.31 se pueden observar que la comparación con el umbral de discriminación con valor de 16 000, fue suficiente para realizar la discriminación de las personas que no se encuentran registradas y reconocer al usuario que si esta registrado. En este caso las personas 1, 6 y 11 corresponden al mismo individuo, el cual, es el usuario registrado, por tal motivo fue aceptado.

6.3. Problema de identificación de múltiples usuarios

Después de que se realizaron y mostraron los resultados de las pruebas para comprobar, la efectividad de utilizar la solución propuesta, al problema de la verificación de un usuario en los sistemas biométricos con base en huellas dactilares, lo siguiente es atender el segundo problema, por lo que, a continuación se mencionará, cual es este problema y se mostraran, los resultados de las pruebas que se realizaron para comprobar la efectividad de la solución

propuesta en esta tesis para la identificación de múltiples usuarios en los sistema biométricos.

6.3.1. Definición del problema de identificación de múltiples usuarios

El segundo problema que aborda este trabajo de tesis es *la identificación de múltiples usuarios* por medio de sus huellas dactilares. El término múltiples usuarios, se refiere a que el sistema biométrico identifica a un conjunto de usuarios registrados en la base de datos, este tipo de situaciones se presenta cuando el mismo servicio se presta a diversas personas, un ejemplo podría ser si se aplica en una casa habitación, donde sólo los miembros de la familia puedan tener acceso a ella, otros ejemplos serían el acceso a la entrada de un club, el acceso a la entrada a un edificio, el acceso a la entrada de una comunidad en la red, etc.

En las situaciones mencionadas, un servicio se presta a varias personas (usuarios) que fueron registradas con anterioridad para tener la autorización del uso de un servicio y siendo estos usuarios los únicos autorizados para hacer uso de este servicio, si otra persona que no se encuentra registrada (no usuario) intentara tener acceso al servicio, ésta será rechazada por el sistema biométrico, ver figura 6.32.

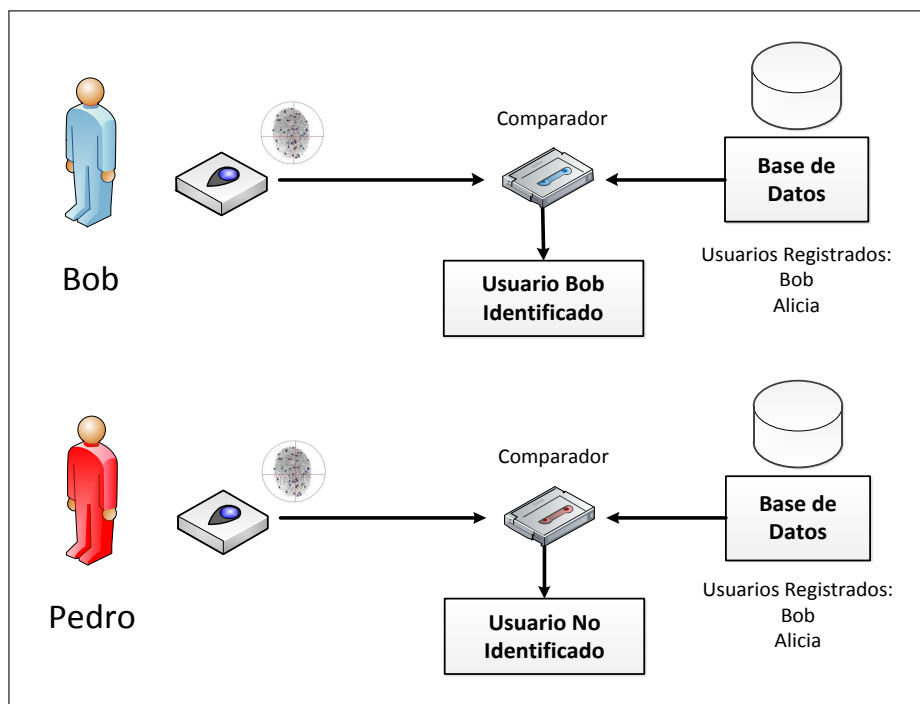


Figura 6.32: Diagrama esquemático que ilustra el problema de identificar al usuario registrado.

6.3.2. Prueba realizada para la posible solución del problema

El propósito de la prueba es comprobar la efectividad de utilizar como valor de umbral 16 000, que fue seleccionado durante las pruebas del problema de verificación de un sólo usuario. Para la realización de la prueba se seleccionaron 5 usuarios, a los cuales, se les tomó una imagen de su pulgar izquierdo, que posteriormente se almacenó en la base de datos, después con el objetivo de comprobar, que independientemente del número de veces que se obtenga el histograma de la misma imagen dará como resultado exactamente el mismo histograma y por consiguiente, la comparación de estos histogramas dará como resultado 0. Se procedió a obtener en dos ocasiones el histograma de la misma imagen del usuario 1 almacenado en la base de datos, para realizar la comparación de estos histogramas. El mismo procedimiento se realizó para cada uno de los restantes usuarios registrados en la base de datos. En la figura 6.33 se observa las imágenes de las huellas que serán registradas en la base de datos, mientras que en las figuras 6.34, 6.35, 6.36, 6.37 y 6.38, se pueden observar los histogramas de las huellas que se encuentra almacenadas en la base de datos, mientras que en la figura 6.39 se puede observar el resultado que se obtiene en la comparación de histogramas de la misma imagen de la huella almacenada en la base de datos.

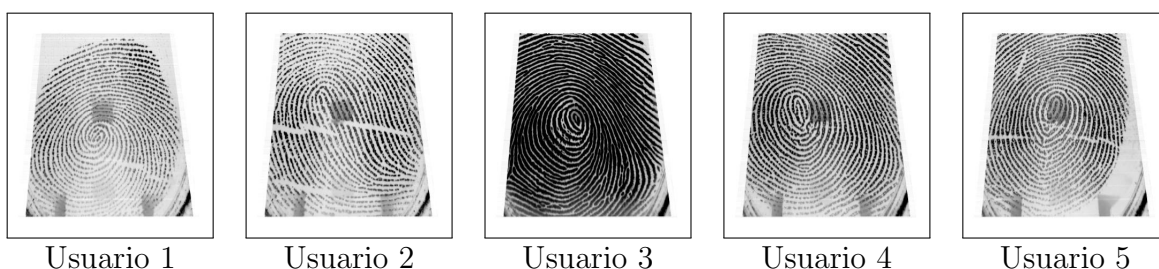


Figura 6.33: Imágenes de las muestras de las cinco personas que se registraron en la base de datos.

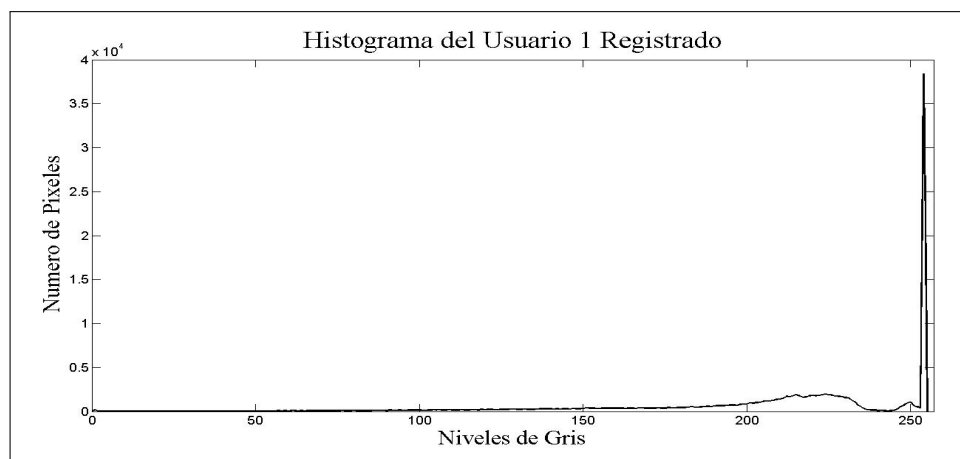


Figura 6.34: Histograma del usuario 1 registrado en la base de datos del problema 2

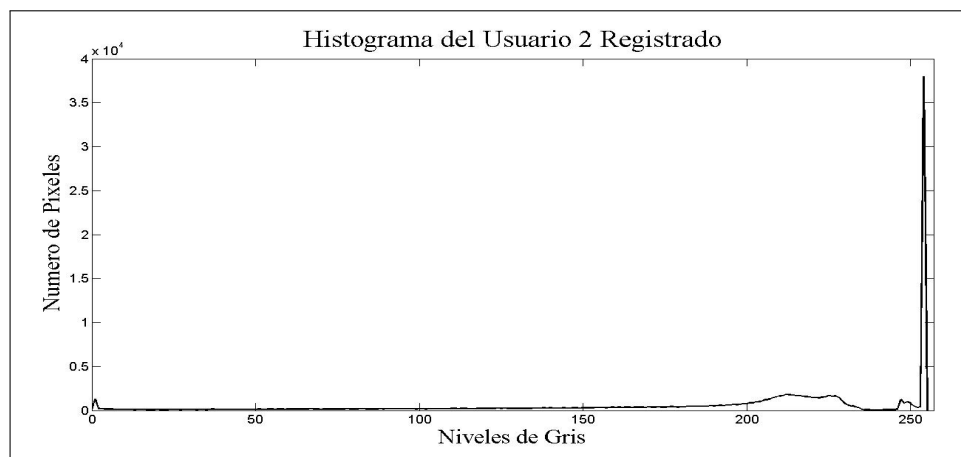


Figura 6.35: Histograma del usuario 2 registrado en la base de datos del problema 2

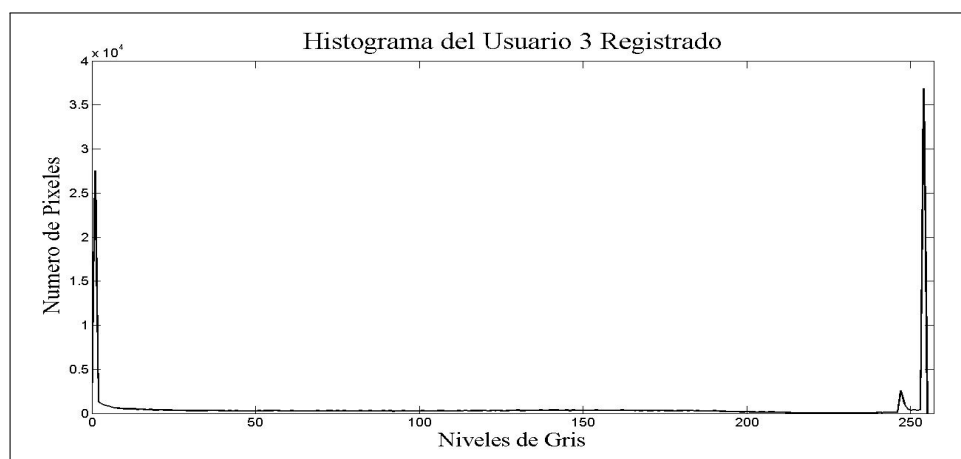


Figura 6.36: Histograma del usuario 3 registrado en la base de datos del problema 2

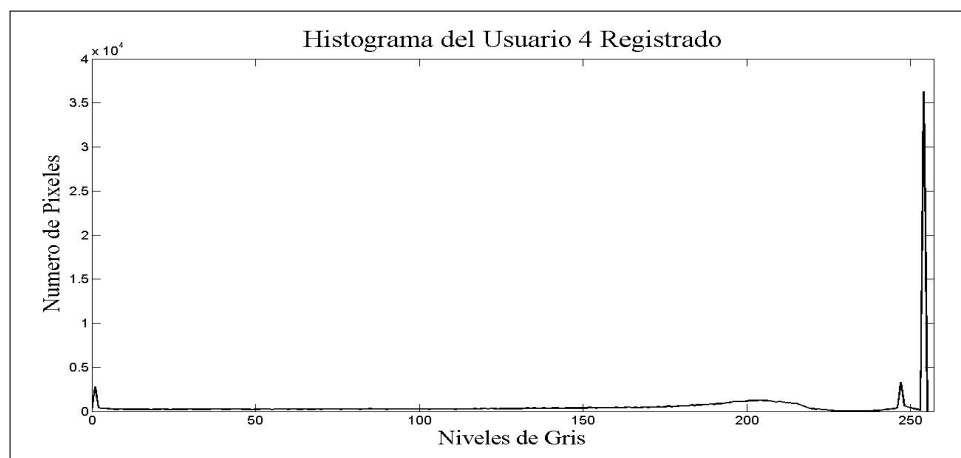


Figura 6.37: Histograma del usuario 4 registrado en la base de datos del problema 2

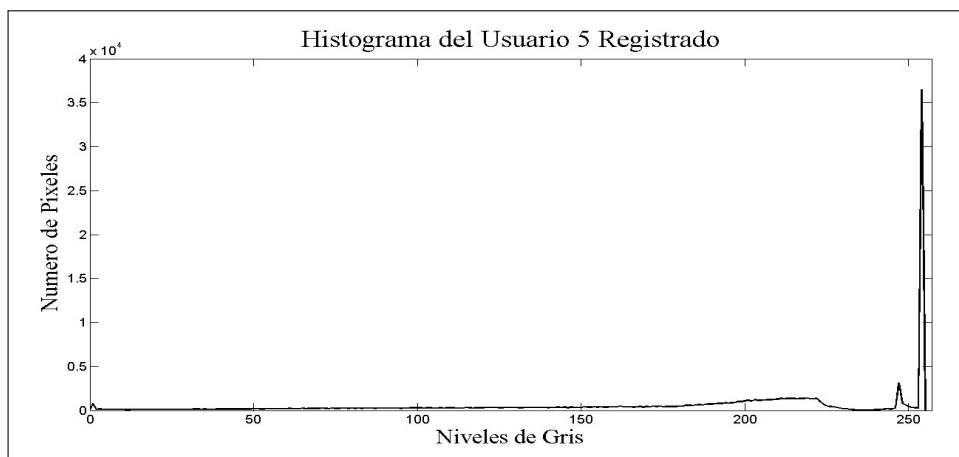


Figura 6.38: Histograma del usuario 5 registrado en la base de datos del problema 2

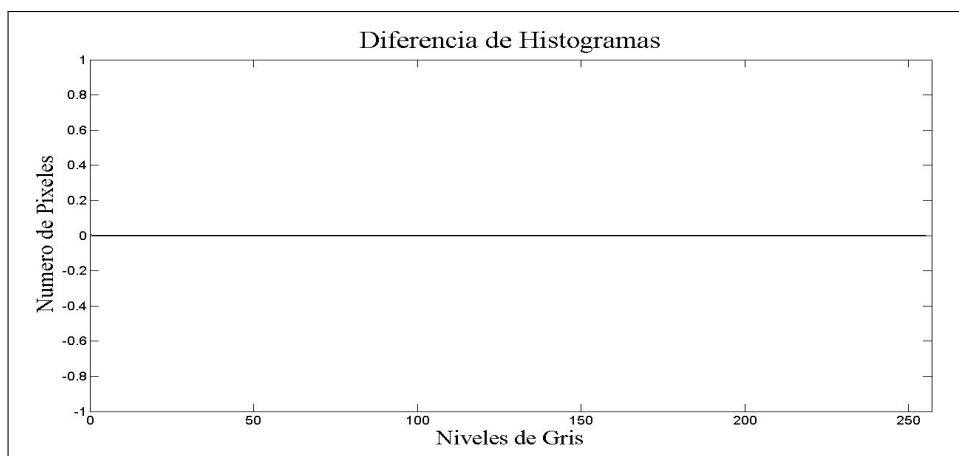


Figura 6.39: Histograma obtenido en la comparación con la misma imagen almacenada en la base de datos.

Comentario: Las imágenes de las huellas de los usuarios registrados en la base de datos que se mostraron en las anteriores figuras, se tomaron en las condiciones lo más similares posibles. Esto se realizó limpiando el lector en cada lectura de muestra, limpiando el dedo de la persona a quien se le tomaba la muestra, colocando el dedo de la persona en la misma posición para todos los usuarios y teniendo la misma iluminación para todas la tomas de muestras.

Antes de mostrar los resultados obtenidos de la prueba realizada, se debe mencionar el proceso que se siguió para realizarla, como sigue: considerando a una persona que solicita ser reconocido, éste deberá colocar su dedo sobre el lector de huellas dactilares para tomar la imagen de entrada, el siguiente paso es obtener su histograma y posteriormente compararlo con el histograma de cada uno de los usuarios registrados en la base de datos, esto dará como resultado la diferencia de los niveles de gris debido a la comparación de los histogramas, lo cual, determina cuantos pixeles son distintos en la imagen de entrada con respecto a la

imagen almacenada en la base de datos. Siendo este valor el que se dará como resultado. Este proceso se podrá entender mejor observando el diagrama que se muestra en la figura 6.40.

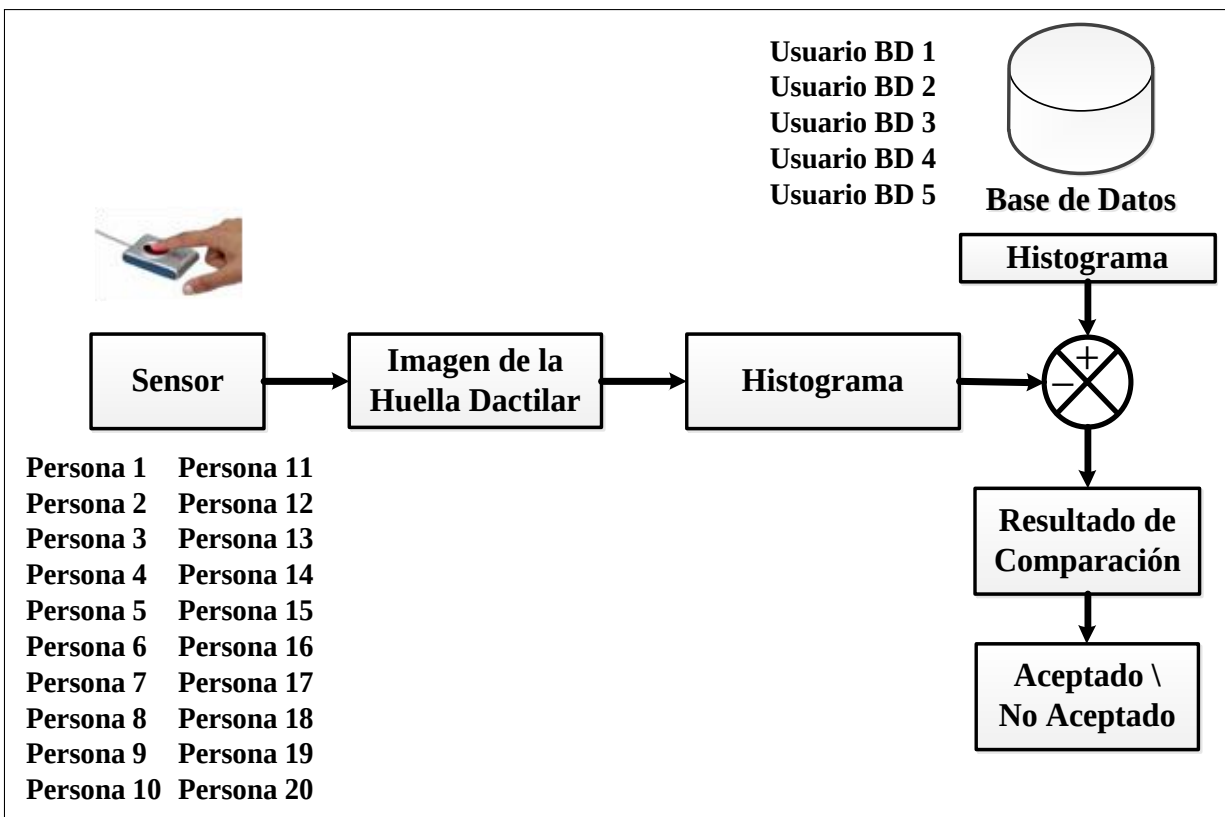


Figura 6.40: Diagrama de proceso de comparación para la identificación de múltiples usuarios.

Después de registrar a cinco usuarios en la base de datos, lo que siguió para realizar la prueba, fue tomar imágenes adicionales del pulgar izquierdo de los individuos registrados en la base de datos. Además, se tomaron imágenes del pulgar izquierdo de 15 individuos no registrados. Estas imágenes se tomaron en condiciones lo más parecidas posibles con las que se tomaron las registradas. El objetivo fue obtener un conjunto de 20 imágenes de huellas en total, con las cuales se realizó la prueba. Las 20 muestras se pueden ver en la figura 6.41.

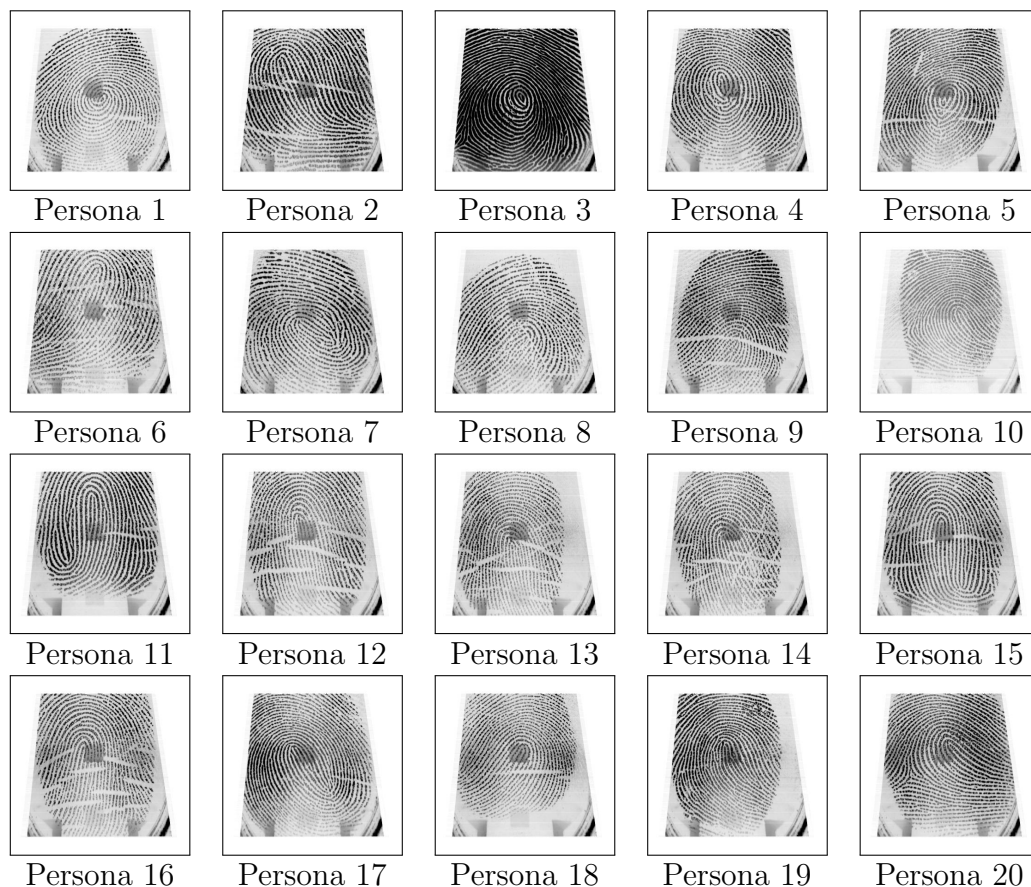


Figura 6.41: Muestras del pulgar izquierdo de 20 personas que solicitan acceso, las cuales, son utilizadas en la identificación de múltiples usuarios.

Las imágenes de las huellas de las personas que se muestran en la figura 6.41, se tomaron en las condiciones lo más similares posibles con respecto a las imágenes que se encuentran en la base de datos, esto se realizó limpiando el lector en cada lectura de muestra, limpiando el dedo de cada persona que se le tomaba la muestra, colocando el dedo de cada persona en la misma posición para todos los individuos y manteniendo la misma iluminación para todas las tomas de muestras. Se hace la observación que las personas del 1 al 5 se encuentran registradas en la base de datos (es decir, son usuarios) y siguiendo el mismo orden.

Los histogramas de las 20 imágenes de huellas dactilares se muestran en dos figuras, en la figura 6.42 se pueden observar los histogramas de las primeras 10 muestras (1 a la 10), mientras que en la figura 6.43 se observan los histogramas de las 10 muestras restantes (11 a la 20).

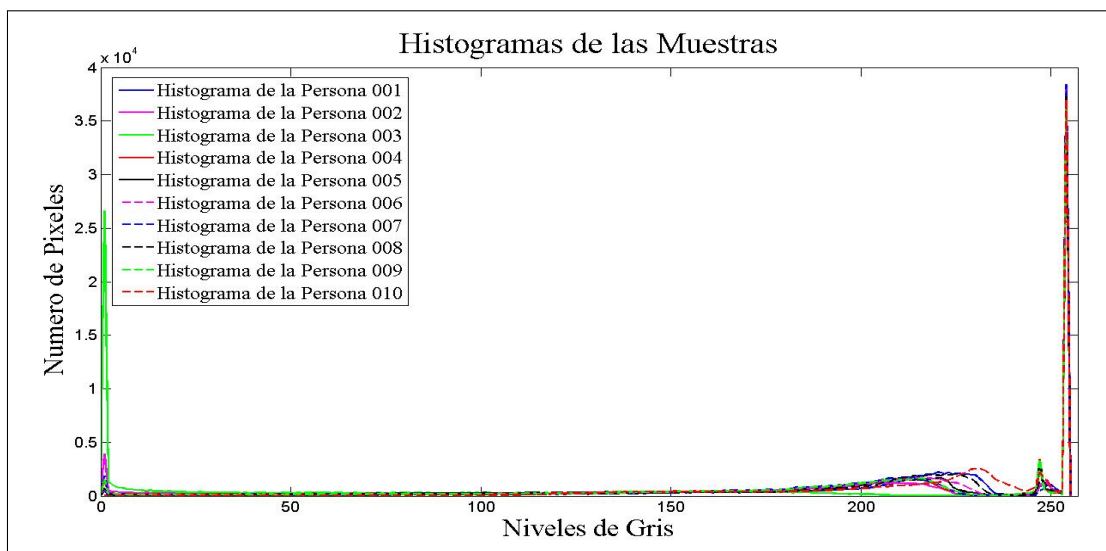


Figura 6.42: Histogramas de las muestras de las personas del 1 al 10.

En la figura 6.42 se puede observar la distribución de niveles de gris de un histograma con respecto a otros, en algunas ocasiones son similares y en otras su diferencia es muy notable. El ejemplo más claro es el histograma de la muestra 3 que difiere bastante de los demás histogramas.

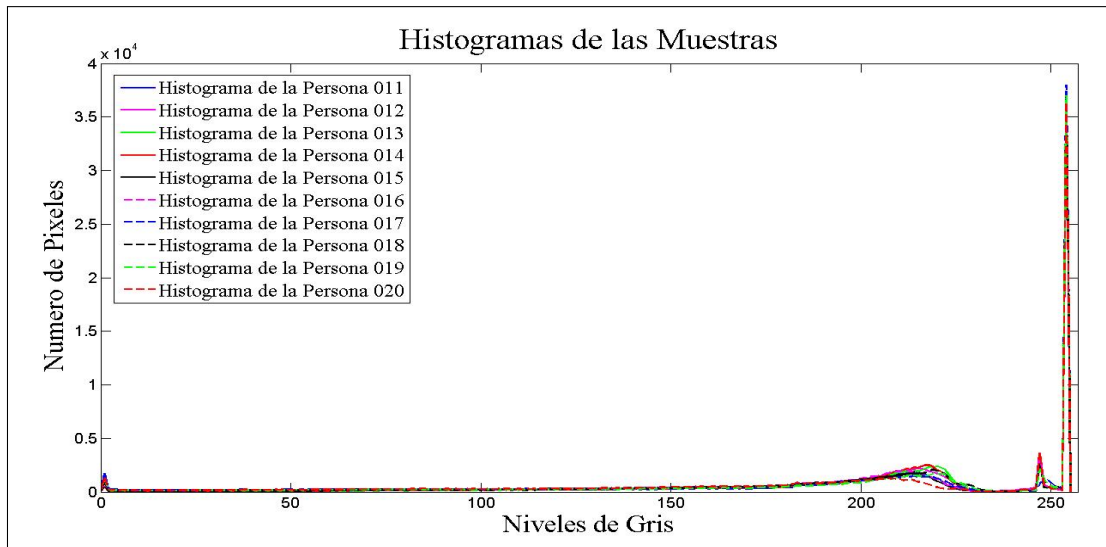


Figura 6.43: Histogramas de las muestras de las personas del 11 al 20.

En la figura 6.43 se puede observar la distribución de niveles de gris de un histograma con respecto a los otros, que son muy parecidos y la diferencia entre ellos es menos notoria en comparación con los histogramas mostrados en la figura 6.42. Lo que podría deducirse

viendo estos histogramas es que hay muchas posibilidades de presentarse falsas aceptaciones en la comparación, con algunos de estos histogramas.

Resultados de prueba realizada

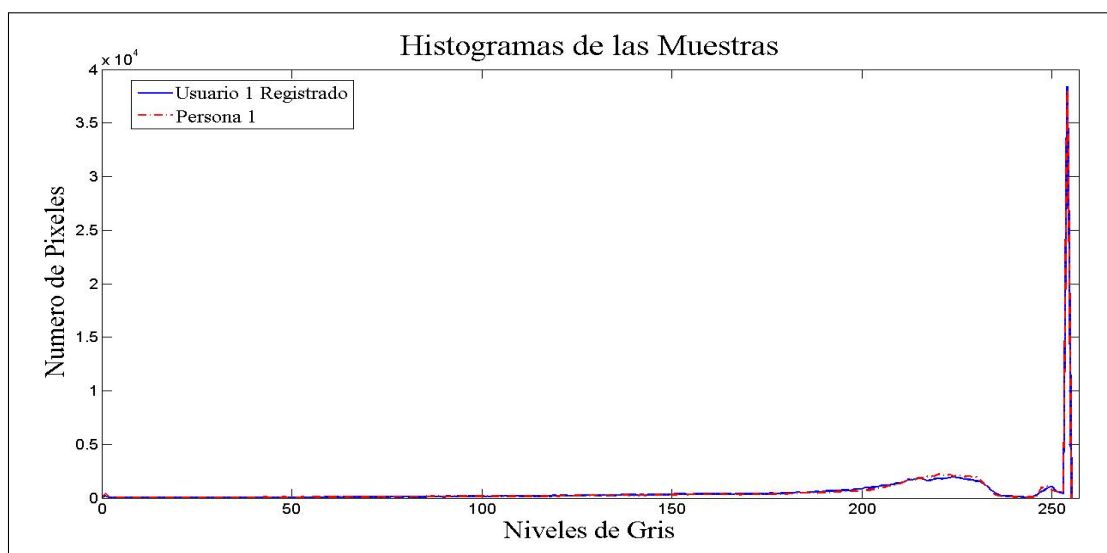
Después de mostrar lo que se requirió para la realización de la prueba, lo siguiente es mostrar los resultados obtenidos. En la tabla 6.5 se pueden observar los resultados de la comparación de histogramas, los cuales, se obtuvieron con la ayuda de un programa implementado en MatLab (ver anexo B), que realiza el proceso mostrado en la figura 6.40, el cual, utiliza las operaciones que indican las expresiones matemáticas (5.1) y (5.2).

<i>Histogramas_{BD}</i> \ <i>Histogramas_{Personas}</i>	UBD 1	UBD 2	UBD 3	UBD 4	UBD 5
Persona 1	14486	26366	121546	80140	54252
Persona 2	58646	42900	79768	21316	28138
Persona 3	123110	103900	8920	72814	88558
Persona 4	54644	37404	91178	24060	16730
Persona 5	40984	28822	96326	35860	11514
Persona 6	29016	23090	112406	50096	28484
Persona 7	47056	32848	104238	37992	29722
Persona 8	18228	17782	116528	70178	43788
Persona 9	51520	35806	100042	28386	20210
Persona 10	44274	56054	131910	87464	69124
Persona 11	50438	31498	91330	32726	21236
Persona 12	40668	35336	111740	45770	26728
Persona 13	36538	35774	116816	52202	31406
Persona 14	41040	38556	116476	51422	31800
Persona 15	52108	36484	93288	26442	15730
Persona 16	50712	40394	107548	35916	26278
Persona 17	44152	29122	95944	33386	15776
Persona 18	33702	29686	112320	48072	28030
Persona 19	37050	22054	91306	39502	16532
Persona 20	68208	52262	83430	15296	21868

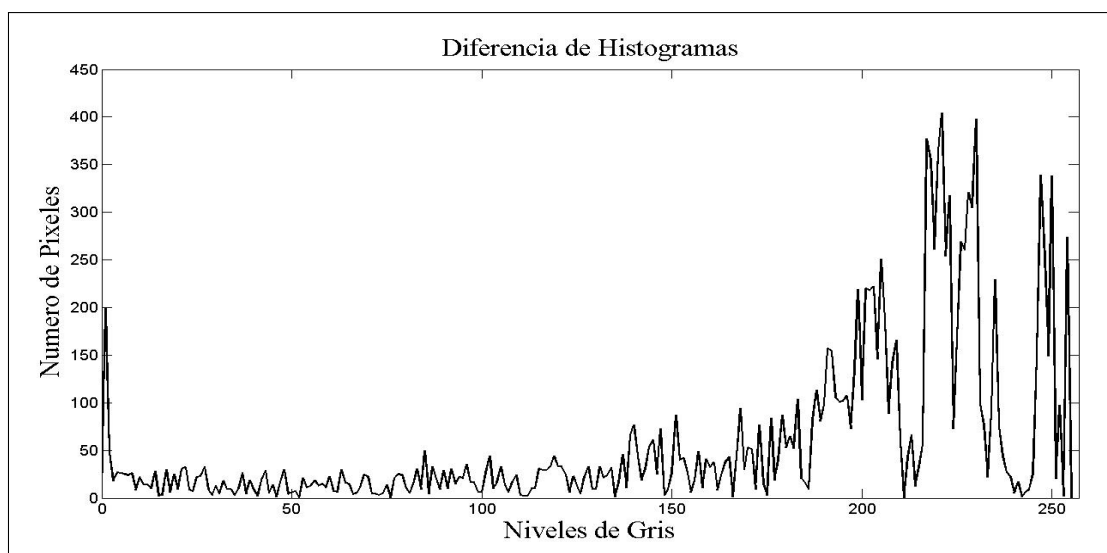
Tabla 6.5: Resultados obtenidos de la comparación de histogramas para el problema de identificación de múltiples usuarios.

A partir de los resultados mostrados en la tabla 6.5, se puede observar que si se toma el valor del umbral que se utilizó en las pruebas del problema anterior, identificación de un usuario, que fue de 16 000, hay algunas personas que serán rechazadas erróneamente y algunas que serán aceptadas erróneamente.

A continuación en las siguientes figuras se mostrarán los histogramas de las imágenes que se encuentran almacenadas en la base de datos, las imágenes de ingreso correspondientes a cada una de las huellas registradas y los histogramas obtenidos de la diferencia.



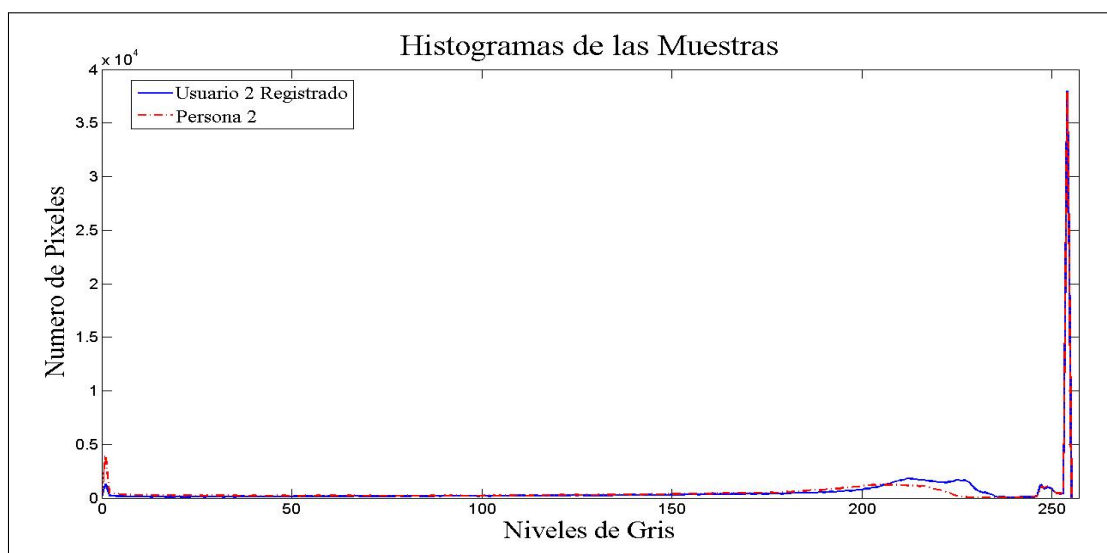
a)



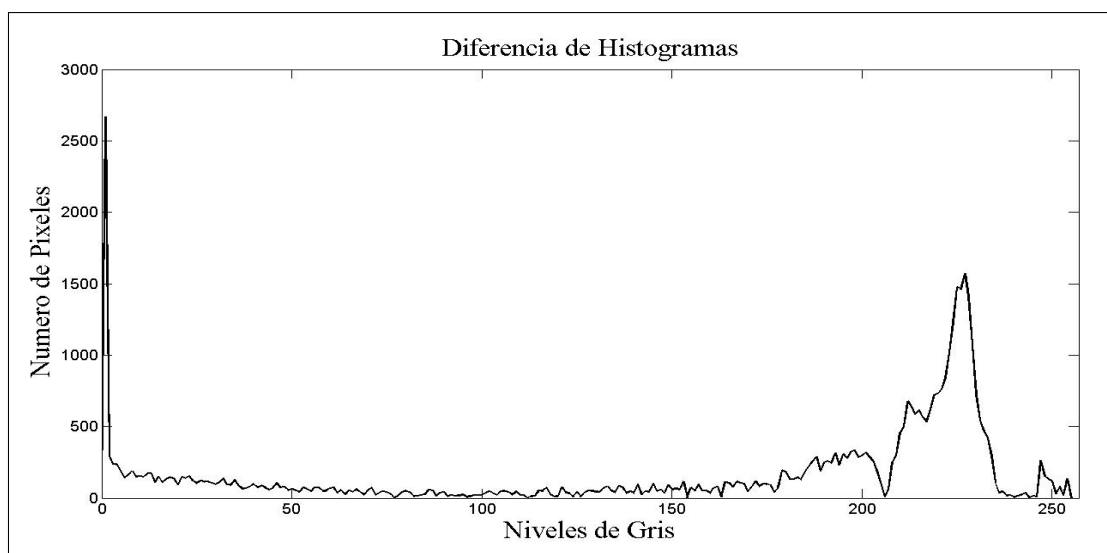
b)

Figura 6.44: En la figura se muestra: a) los histogramas comparados y b) el histograma resultante de la comparación de la persona 1 de ingreso versus el usuario 1 registrado.

En la figura 6.44 se puede observar que la distribución de los niveles de gris que resultaron de la comparación de las muestras, están por debajo de los 450 pixeles en los niveles de gris.



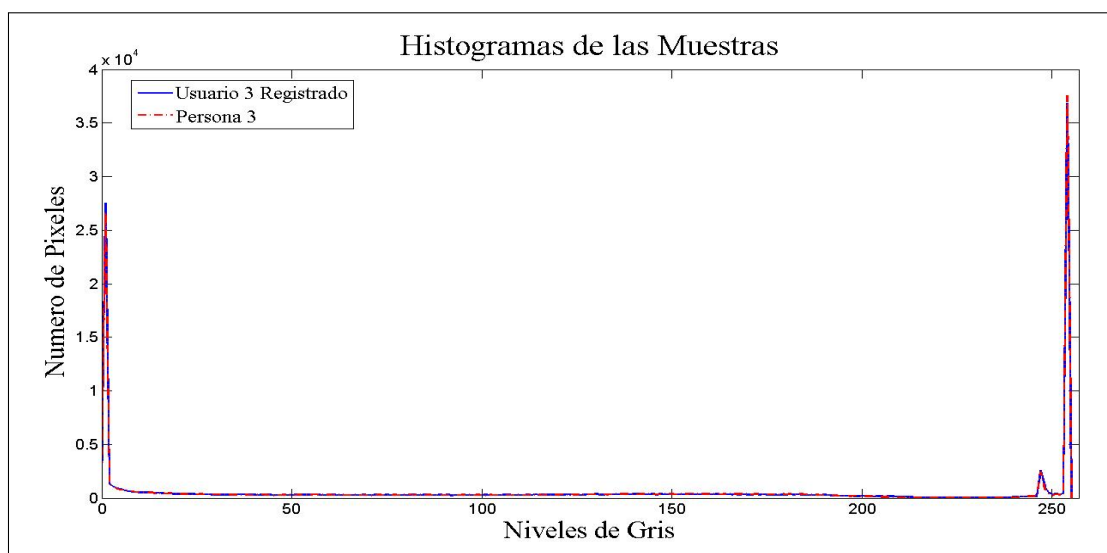
a)



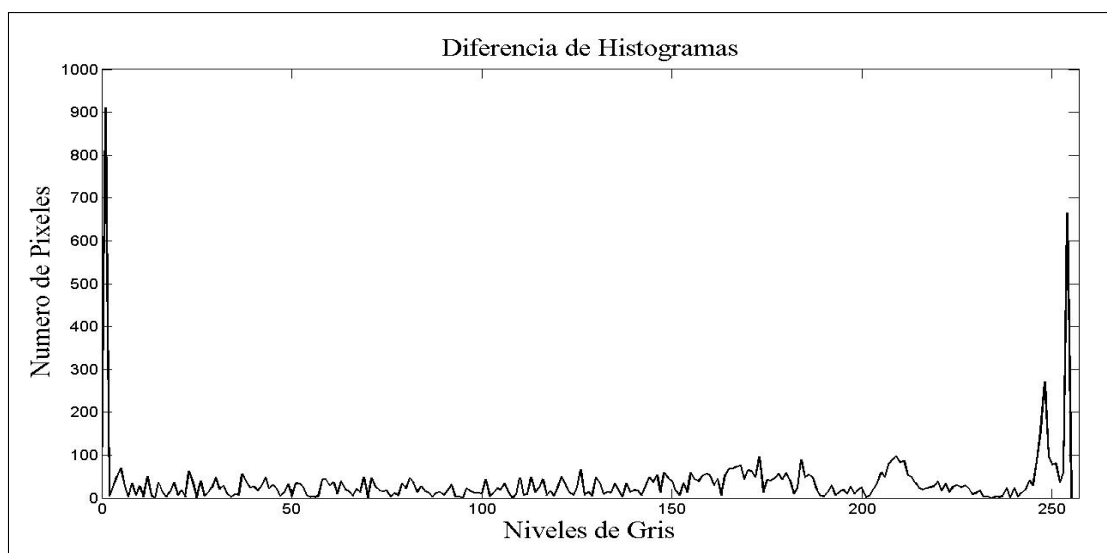
b)

Figura 6.45: En la figura se muestra: a) los histogramas comparados y b) el histograma resultante de la comparación de la persona 2 de ingreso versus el usuario 2 registrado.

En la figura 6.45 se puede observar que la distribución de los niveles de gris que resultaron de la comparación de las muestras, la mayoría están por debajo de los 1 500 pixeles en los niveles de gris.



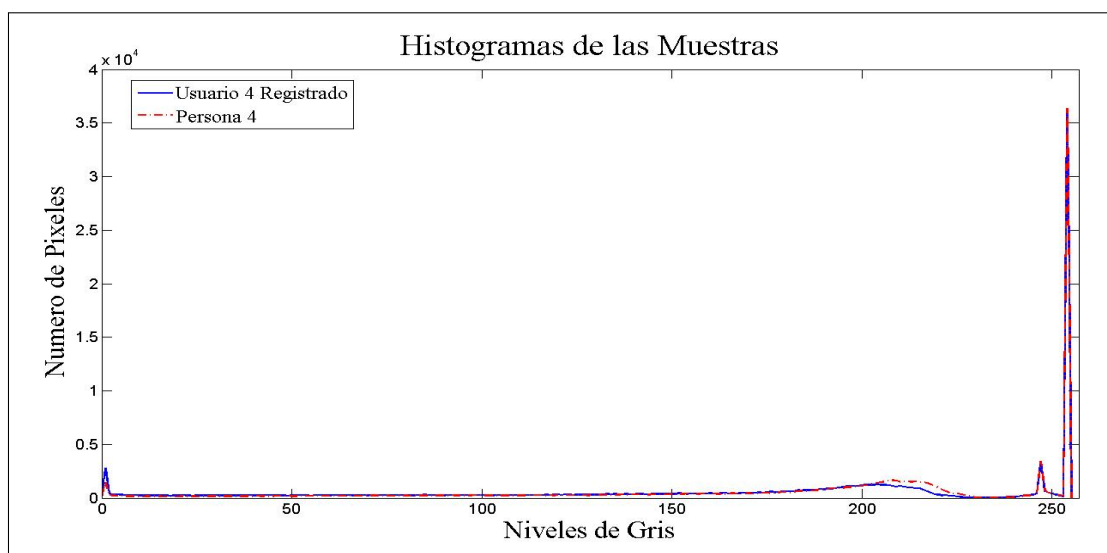
a)



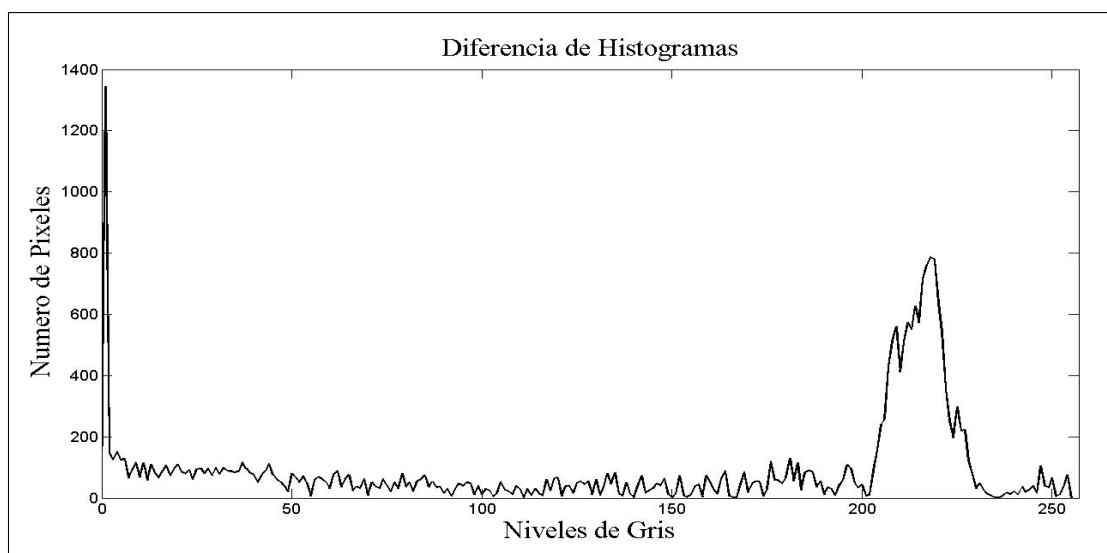
b)

Figura 6.46: En la figura se muestra: a) los histogramas comparados y b) el histograma resultante de la comparación de la persona 3 de ingreso versus el usuario 3 registrado.

En la figura 6.46 se puede observar que la distribución de los niveles de gris que resultaron de la comparación de las muestras, la mayoría están por debajo de los 900 pixeles en los niveles de gris.



a)



b)

Figura 6.47: En la figura se muestra: a) los histogramas comparados y b) el histograma resultante de la comparación de la persona 4 de ingreso versus el usuario 4 registrado.

En la figura 6.47 se puede observar que la distribución de los niveles de gris que resultaron de la comparación de las muestras, la mayoría están por debajo de los 900 pixeles en los niveles de gris.

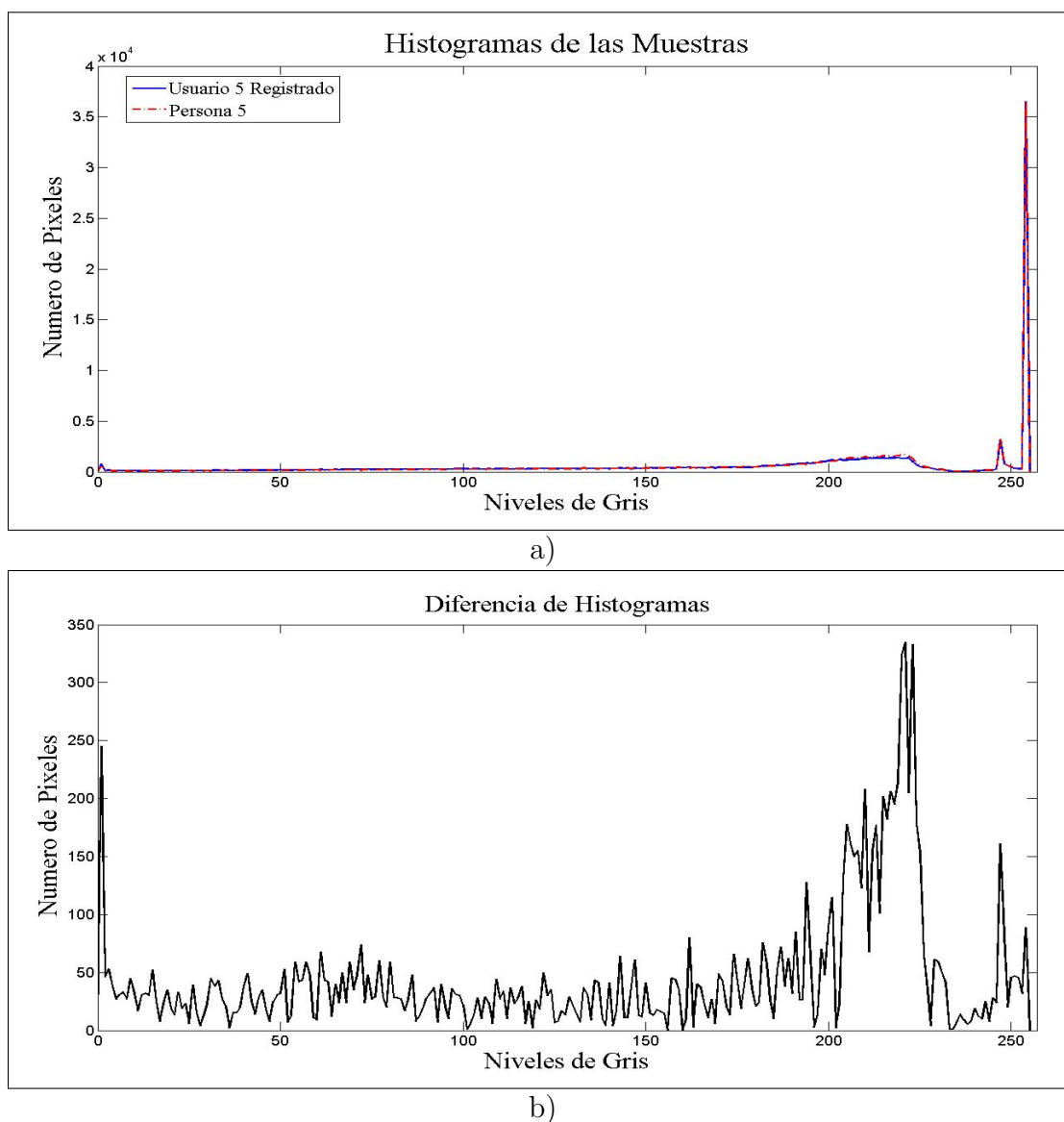


Figura 6.48: En la figura se muestra: a) los histogramas comparados y b) el histograma resultante de la comparación de la persona 5 de ingreso versus el usuario 5 registrado.

En la figura 6.48 se puede observar que la distribución de los niveles de gris que resultaron de la comparación de las muestras, la mayoría están por debajo de los 350 píxeles en los niveles de gris.

Para complementar la información mostrada anteriormente, a continuación se muestra los resultados obtenidos de un programa de comparación de histogramas implementado en MatLab (ver anexo B), en el cual, se ha utilizado un valor de umbral de discriminación de 16 000. Las siguientes figuras corresponden a los resultados de la comparación de las muestras que fueron reconocidas correctamente con respecto a las huellas registradas en la base de datos, pero que fueron tomadas en distintos tiempos.

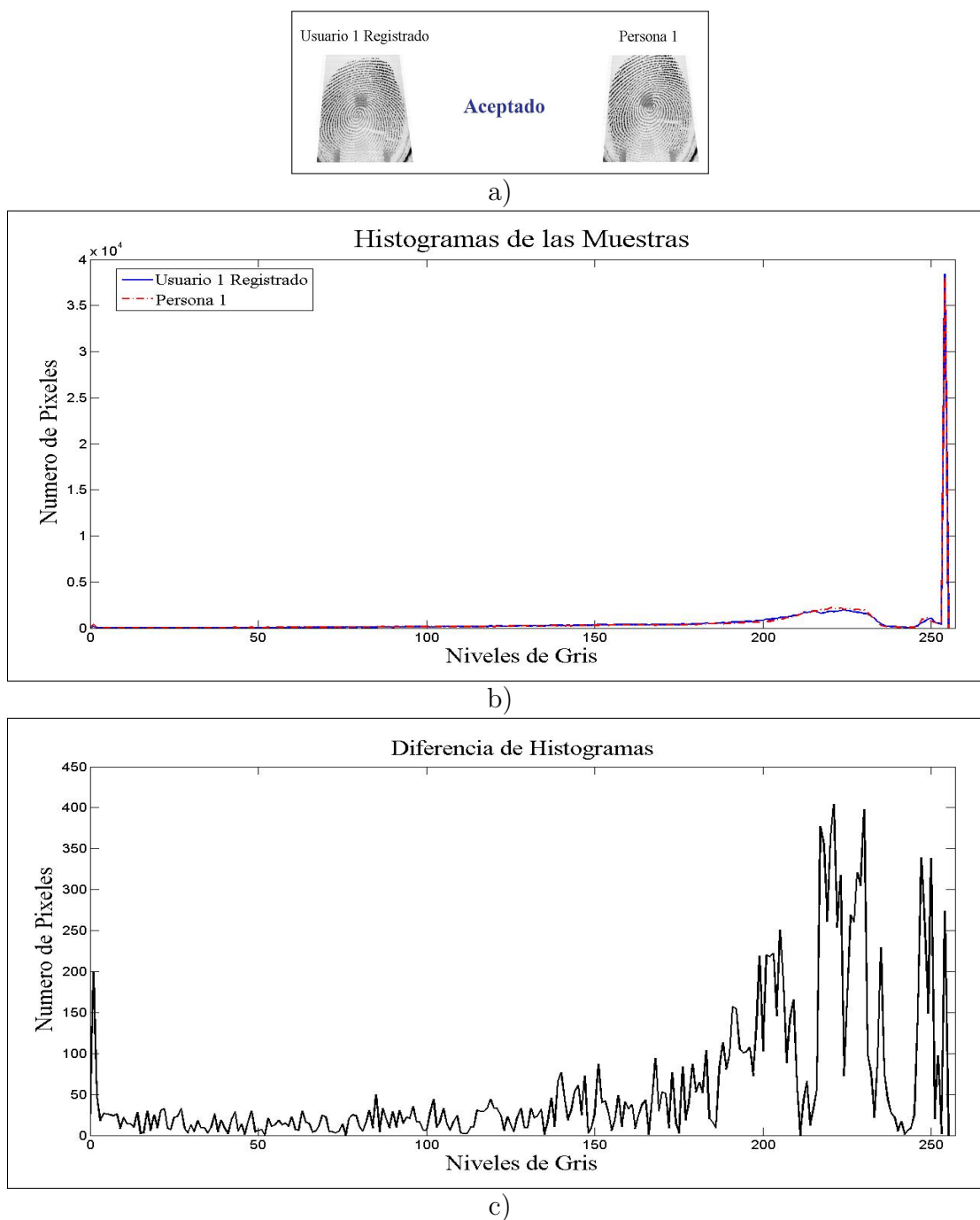


Figura 6.49: Información obtenida del programa en MatLab: a) resultado gráfico de la comparación, b) histogramas comparados y c) histograma resultante de la diferencia.

Comentario: Esta huella fue aceptada correctamente y no presentó ninguna falsa aceptación.

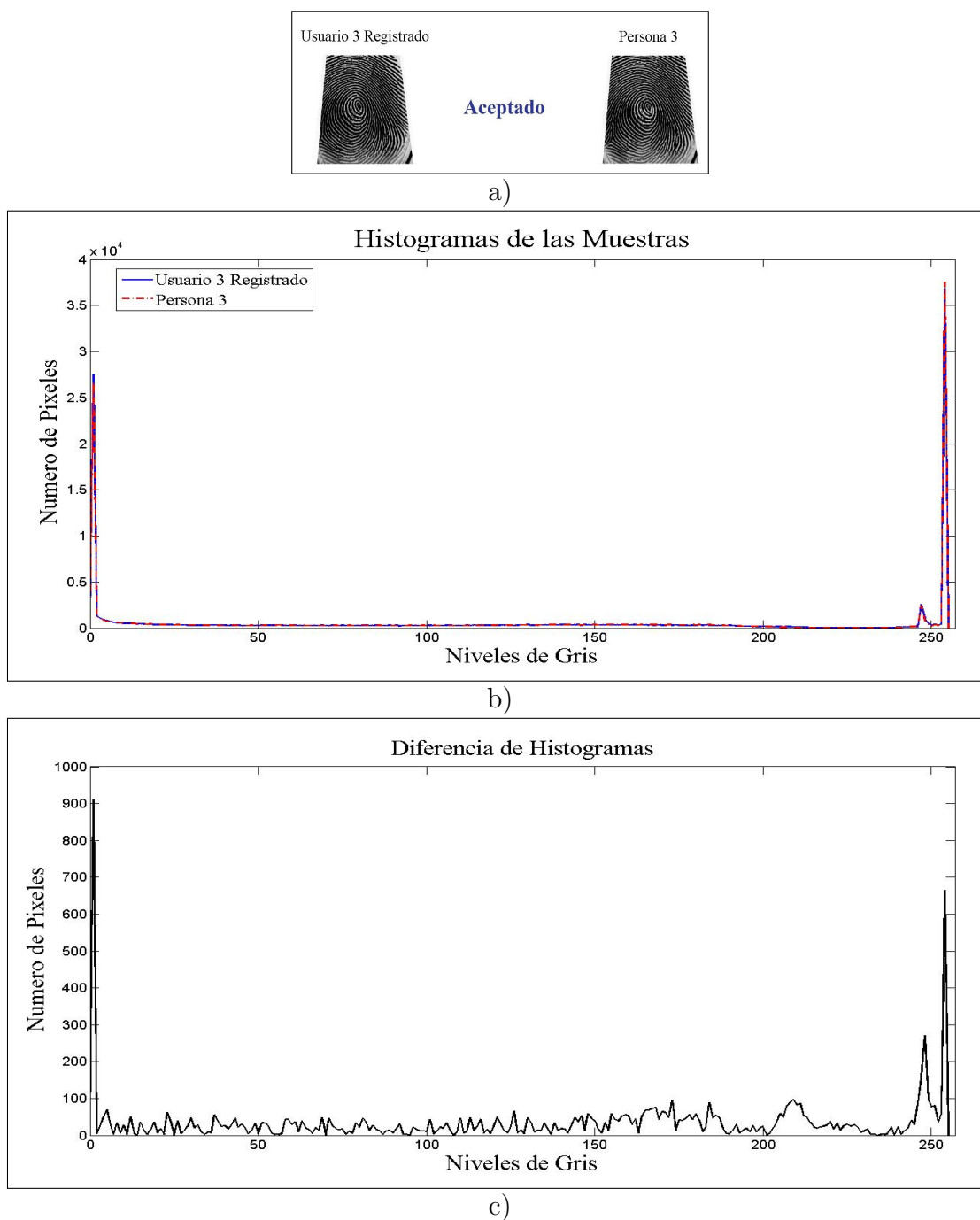


Figura 6.50: Información obtenida del programa en MatLab: a) resultado gráfico de la comparación, b) histogramas comparados y c) histograma resultante de la diferencia.

Comentario: Esta huella fue aceptada correctamente y no presentó ninguna falsa aceptación.

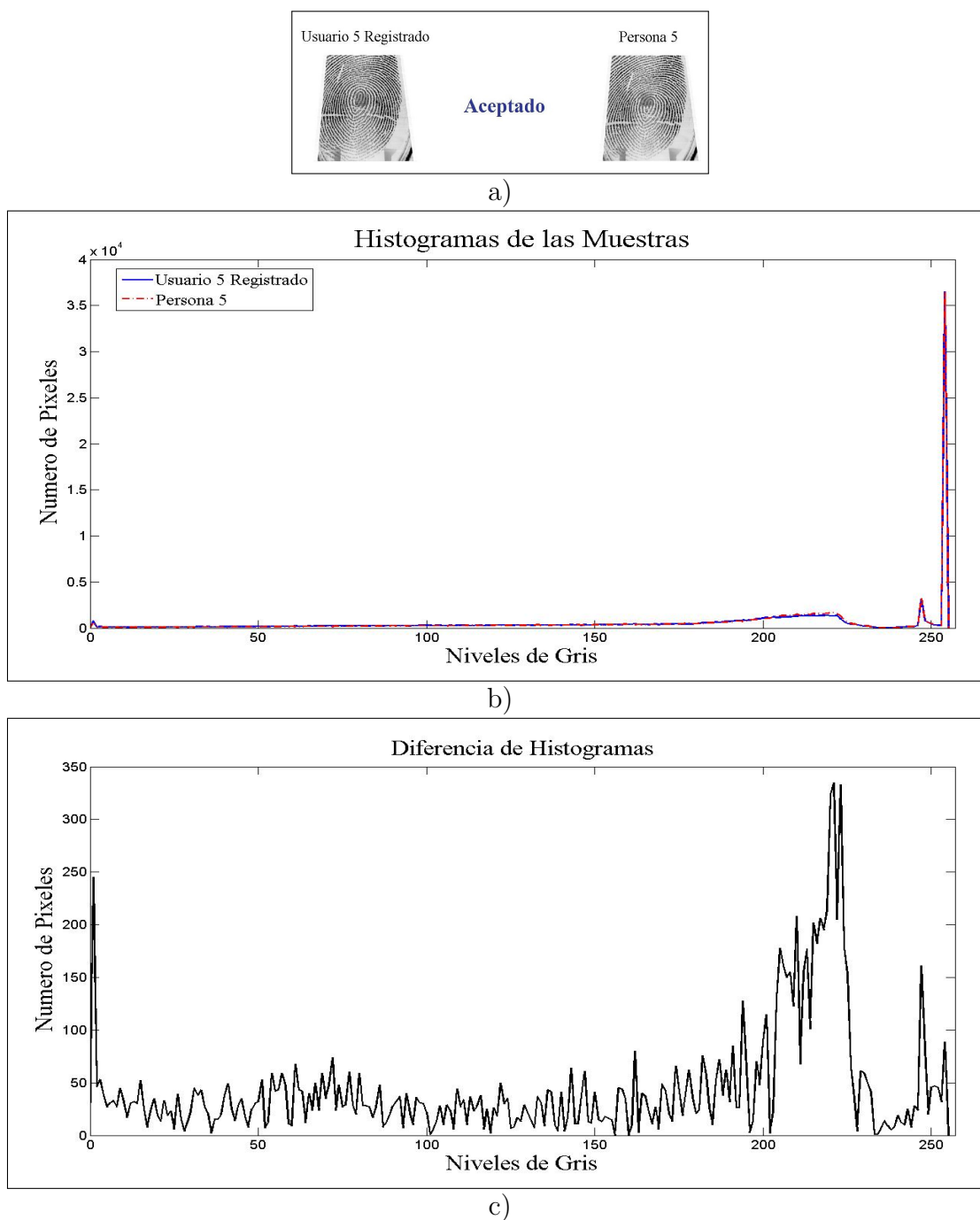


Figura 6.51: Información obtenida del programa en MatLab: a) resultado gráfico de la comparación, b) histogramas comparados y c) histograma resultante de la diferencia.

Comentario: Esta huella fue aceptada correctamente, pero presentó dos falsas aceptaciones, que más adelante se mostrarán.

Debido a que el número de figuras correspondientes a los resultados de comparación de las muestras, que fueron rechazadas correctamente son demasiadas para mostrar en este documento, se optó por no mostrarlas y en su lugar presentar la tabla 6.6, la cual, muestra los resultados numéricos mostrados en la tabla 6.5, pero en esta ocasión las celdas están coloreadas para distinguir a las personas aceptadas correctamente, las personas rechazadas correctamente, las personas aceptadas erróneamente y las personas rechazadas erróneamente.

Donde:



Representa a las personas aceptadas correctamente.

Representa a las personas aceptadas erróneamente.

Representa a las personas rechazadas correctamente.

Representa a las personas rechazadas erróneamente.

<i>Histogramas_{BD}</i> \ <i>Histogramas_{Personas}</i>	UBD 1	UBD 2	UBD 3	UBD 4	UBD 5
Persona 1	14486	26366	121546	80140	54252
Persona 2	58646	42900	79768	21316	28138
Persona 3	123110	103900	8920	72814	88558
Persona 4	54644	37404	91178	24060	16730
Persona 5	40984	28822	96326	35860	11514
Persona 6	29016	23090	112406	50096	28484
Persona 7	47056	32848	104238	37992	29722
Persona 8	18228	17782	116528	70178	43788
Persona 9	51520	35806	100042	28386	20210
Persona 10	44274	56054	131910	87464	69124
Persona 11	50438	31498	91330	32726	21236
Persona 12	40668	35336	111740	45770	26728
Persona 13	36538	35774	116816	52202	31406
Persona 14	41040	38556	116476	51422	31800
Persona 15	52108	36484	93288	26442	15730
Persona 16	50712	40394	107548	35916	26278
Persona 17	44152	29122	95944	33386	15776
Persona 18	33702	29686	112320	48072	28030
Persona 19	37050	22054	91306	39502	16532
Persona 20	68208	52262	83430	15296	21868

Tabla 6.6: Resultados de la comparación de histogramas para múltiples usuarios.

En la tabla 6.6 se puede observar que las personas que no se encuentran registradas en la base de datos fueron rechazadas en la mayoría de las veces. Sin embargo, también se puede observar que en algunas ocasiones se presentan falsas aceptaciones y falsos rechazos.

En las siguientes figuras se mostrarán los resultados de la comparación por medio de histogramas, correspondientes a las personas que fueron rechazadas erróneamente.

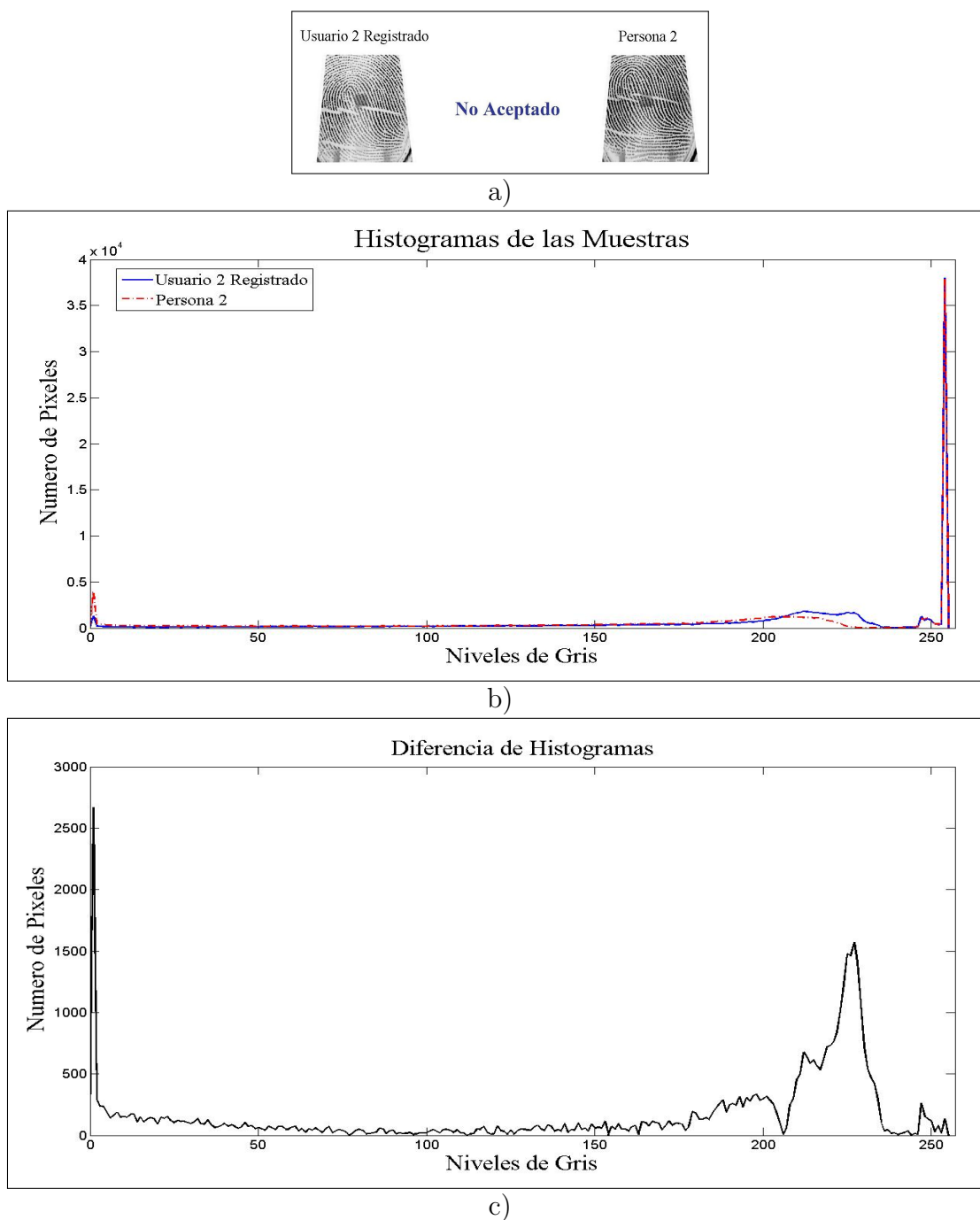


Figura 6.52: Información obtenida del programa en MatLab: a) resultado gráfico de la comparación, b) histogramas comparados y c) histograma resultante de la diferencia.

Comentario: Esta huella fue rechazada erróneamente y no presentó ninguna falsa aceptación.

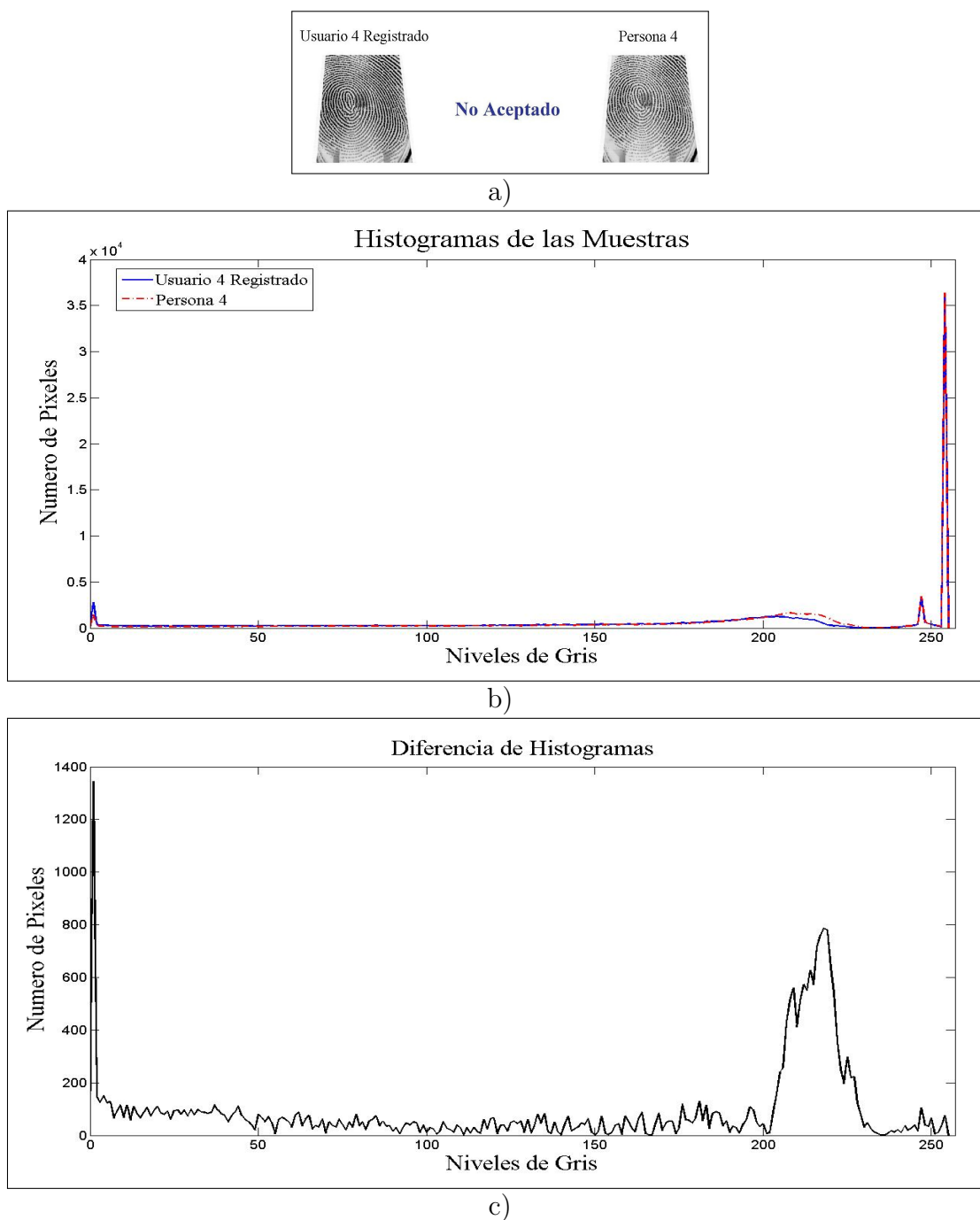


Figura 6.53: Información obtenida del programa en MatLab: a) resultado gráfico de la comparación, b) histogramas comparados y c) histograma resultante de la diferencia.

Comentario: Esta huella fue rechazada erróneamente y presentó una falsa aceptación.

En las siguientes figuras se mostrarán los resultados de la comparación por medio de histogramas, correspondientes a las personas que fueron aceptadas erróneamente.

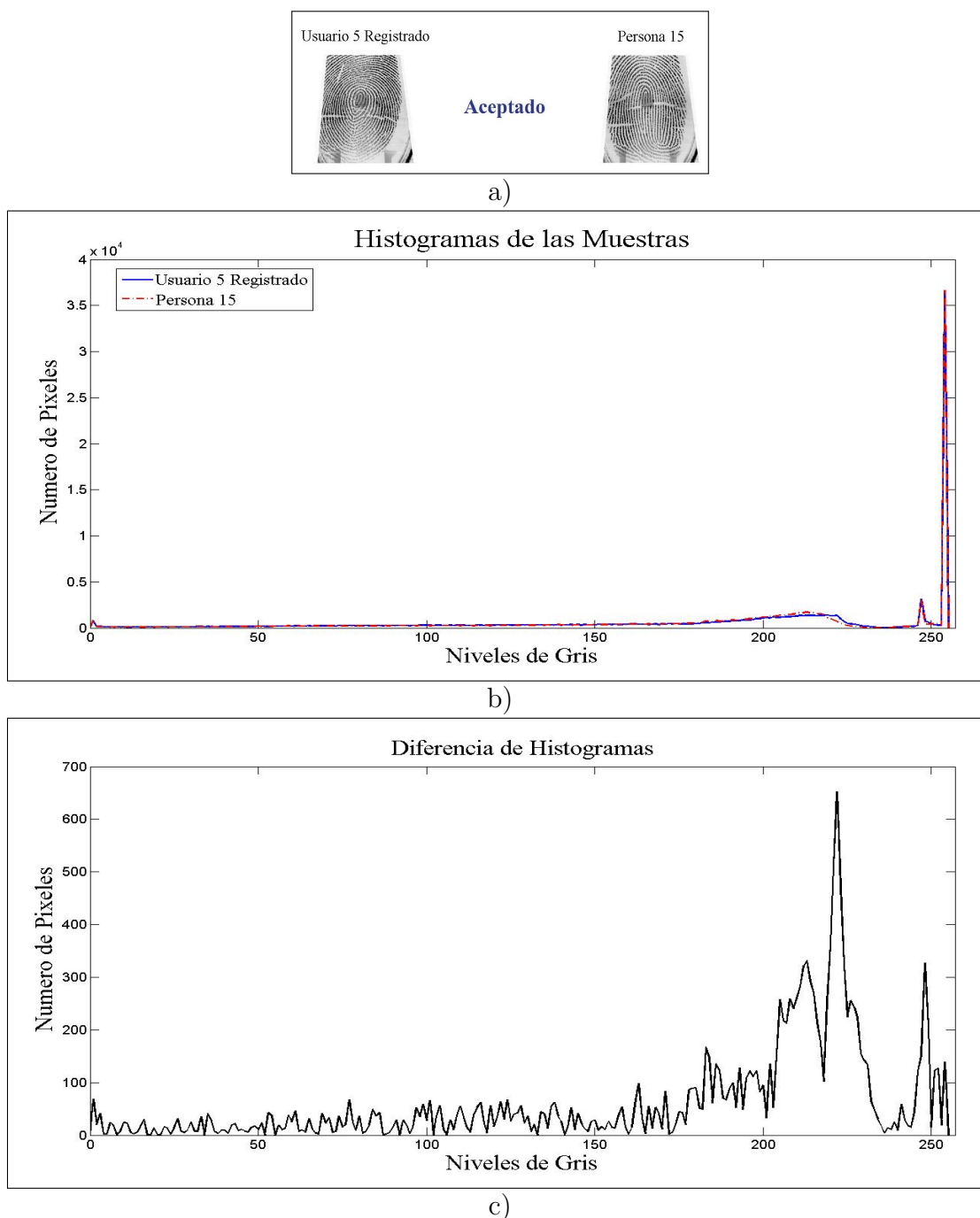


Figura 6.54: Información obtenida del programa en MatLab: a) resultado gráfico de la comparación, b) histogramas comparados y c) histograma resultante de la diferencia.

En la figura 6.54 se puede ver que la persona 15 fue confundida con el usuario 5 de la base de datos, en otras palabras, la imagen de la huella de entrada fue aceptada erróneamente. Este resultado puede ser debido a que la distribución de los niveles de grises de la imagen sea muy parecida y debido a esto sus histogramas resultan ser muy similares.

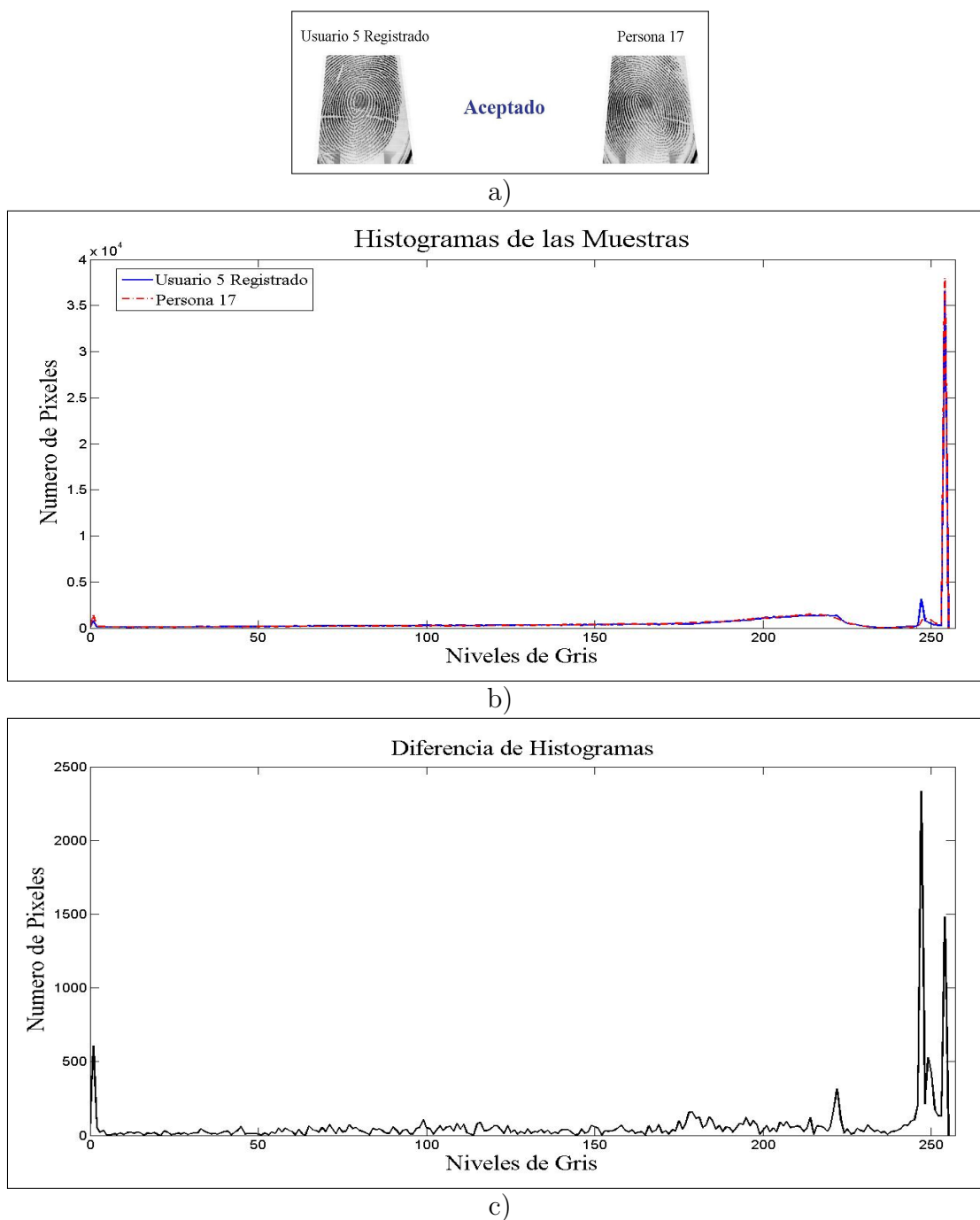


Figura 6.55: Información obtenida del programa en MatLab: a) resultado gráfico de la comparación, b) histogramas comparados y c) histograma resultante de la diferencia.

En la figura 6.55 se puede ver que la persona 17 fue confundida con el usuario 5 de la base de datos, en otras palabras, la imagen de la huella de entrada fue aceptada erróneamente. Hay que notar que es la segunda ocasión donde se confunde una huella de entrada con el usuario 5 que se encuentra registrado en la base de datos.

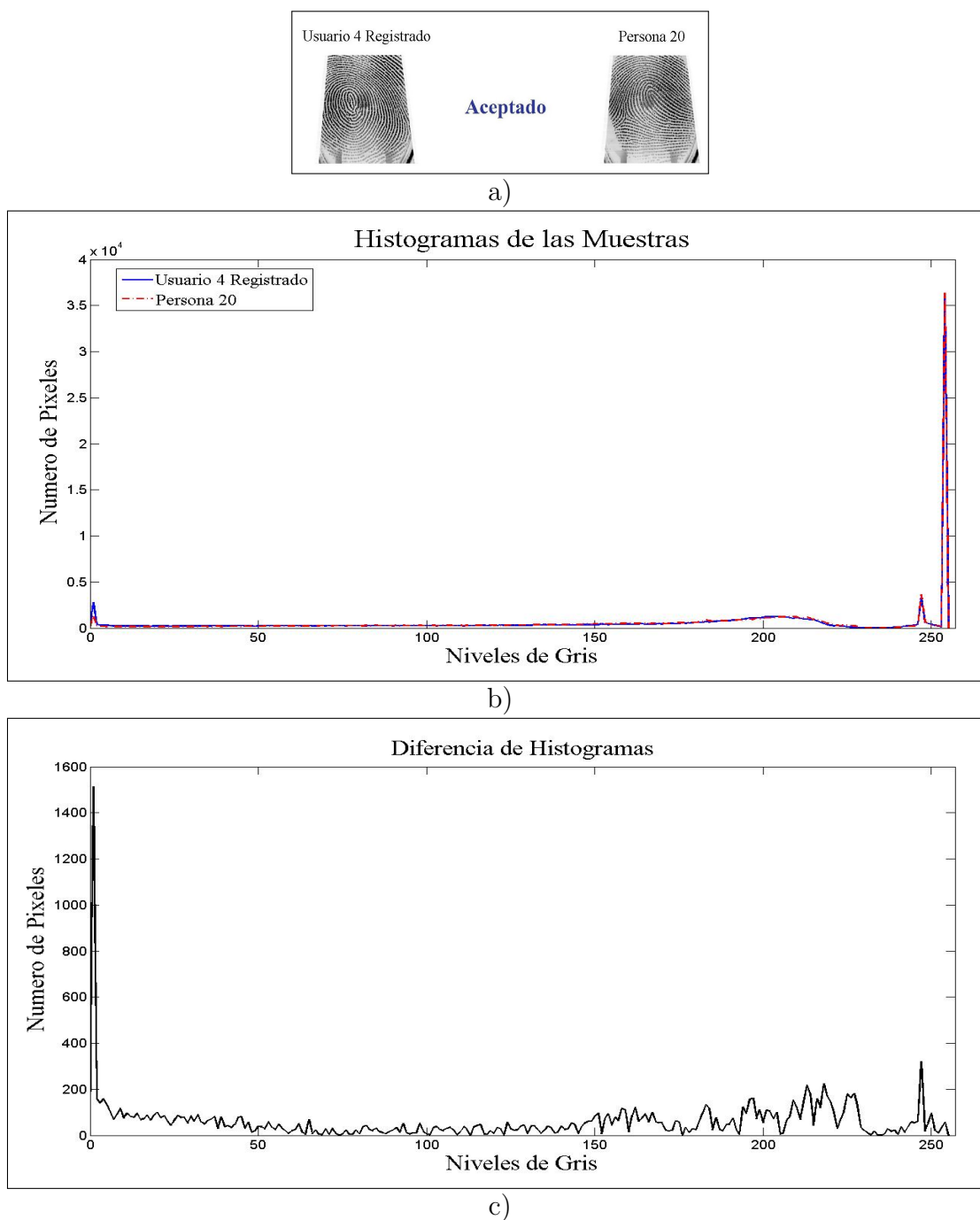


Figura 6.56: Información obtenida del programa en MatLab: a) resultado gráfico de la comparación, b) histogramas comparados y c) histograma resultante de la diferencia.

En la figura 6.56 se puede ver que la persona 20 fue confundida con el usuario 4 de la base de datos, en otras palabras, la imagen de la huella de entrada fue aceptada erróneamente.

6.4. Conclusiones

- Observando los resultados obtenidos en las etapas de calibración y verificación se puede observar por medio de las tablas 6.1, 6.2 y 6.3 que seleccionando el umbral de discriminación de un valor de 16 000 los resultados mejoran al momento de identificar al usuario que se encuentra registrado en la base de datos y discrimina a las personas que no se encuentren en la base de datos. Siendo este valor de umbral el más adecuado por el momento, de acuerdo con las pruebas realizadas.
- Observando los resultados obtenidos por la comparación de histogramas realizados por los programas en MatLab. Se puede notar que el reconocimiento de la huella dactilar resulta más efectivo, si la imagen de entrada del usuario se asemeja más a la que se encuentra registrada en la base de datos.
- Si se observa la tabla 6.6 con los resultados de la comparación de histogramas podemos observar claramente que si se toma un umbral de discriminación de 15 000 en lugar de uno de 16 000, las situaciones donde se presentaron las falsas aceptaciones disminuirían, sin embargo los falsos rechazos se podrían presentar más seguido.
- Tomando en cuenta las situaciones de falso rechazo y falsa aceptación, se tendría que considerar cual de las dos situaciones ocasionaría menor molestia al usuario, para tomar una decisión de si se debe aumentar, disminuir ó dejar el umbral, para así tener el menor número de falsas aceptaciones ó falsos rechazos según la decisión que se tome.

Capítulo 7

Encriptado, desencriptado y comparación de imágenes

7.1. Introducción

La finalidad de estas pruebas realizadas, fue para comprobar la efectividad de utilizar el método de encriptado que se propone en esta tesis, se realizaron tres pruebas la primera fue el encriptado donde además se generó un archivo “txt”, la segunda fue el desencriptado de la imagen utilizando el archivo “txt”, la tercera fue comparar la imagen encriptada versus la imagen recuperada, a continuación se muestran los resultados obtenidos durante estas pruebas y los programas implementados en Matlab que se desarrollaron al finalizar estas últimas pruebas.

7.2. Encriptado de imágenes

El proceso de encriptado de la imagen consiste en seis etapas principales, la primera etapa es para tomar la lectura de una imagen almacenada en la base de datos, con motivo de generar una variable de imagen en el programa, en la segunda etapa se convierte la variable imagen en un vector que contiene cada uno de los bytes de la imagen, en la tercera etapa los bytes de la imagen se convierten en 8 bits con valores de 0 o 1, los cuales, se dividen entre 1 000 y se acomodan en un vector columna, al cual, se le llamará vector de datos. En la cuarta etapa, se generará un vector columna con los datos resultantes del estado caótico x_1 del sistema de Rössler (5.3). En la quinta etapa, se suman los dos vectores mencionados anteriormente para generar un vector columna que contenga los datos ocultos de la imagen. En la última etapa, el vector resultante de la suma se guarda como un archivo “txt”. Para ayudar a la comprensión de este proceso se puede observar la figura 7.1.

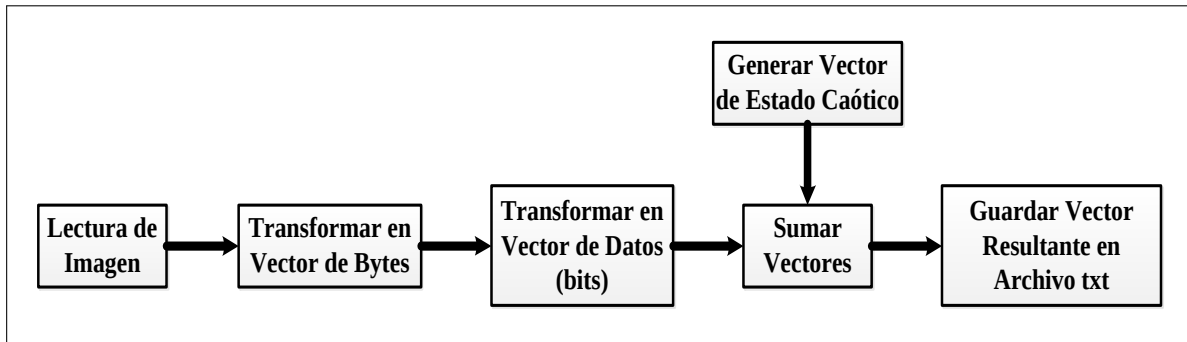


Figura 7.1: Diagrama de bloques del proceso de encriptado de imágenes de huellas dactilares.

La imagen del pulgar izquierdo que se utilizó en la prueba se muestra en la figura 7.2, posteriormente se muestran los resultados generados por el programa implementado en MatLab versión 2008a (ver anexo C.1). En la figura 7.3 a) se muestra una gráfica de los valores del 100500 al 101000 del vector de bits resultantes de la imagen, en la figura 7.3 b) se muestran los valores del 100500 al 101000 del estado caótico x_1 del sistema de Rössler (5.3), en la figura 7.3 c) se muestran los valores del 100500 al 101000 de los datos encriptados con el estado caótico, la última información que proporciona el programa es la figura 7.4.



Figura 7.2: Imagen de la huella dactilar capturada por el sensor que se encriptará.

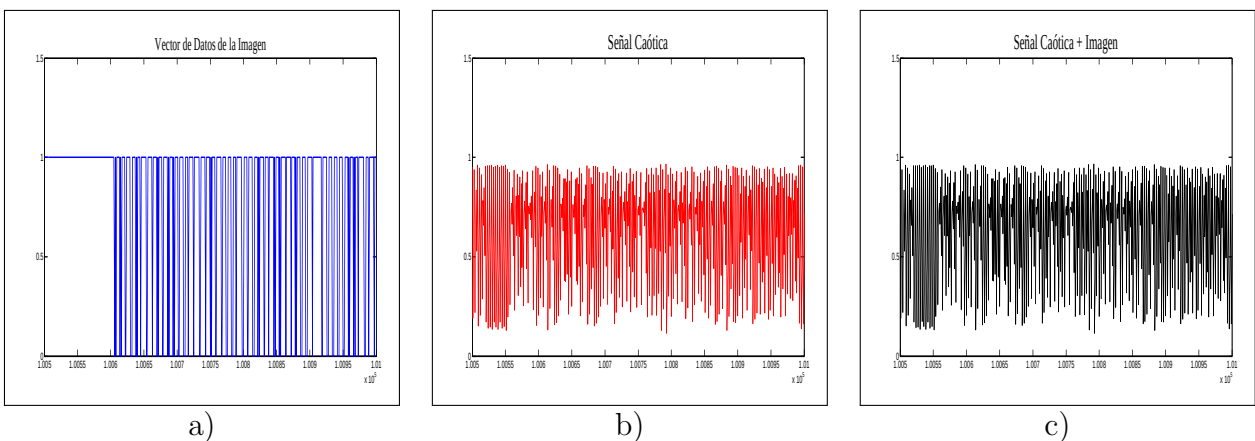


Figura 7.3: La figura contiene la siguiente información: a) vector de bits, b) estado caótico y c) datos encriptados.

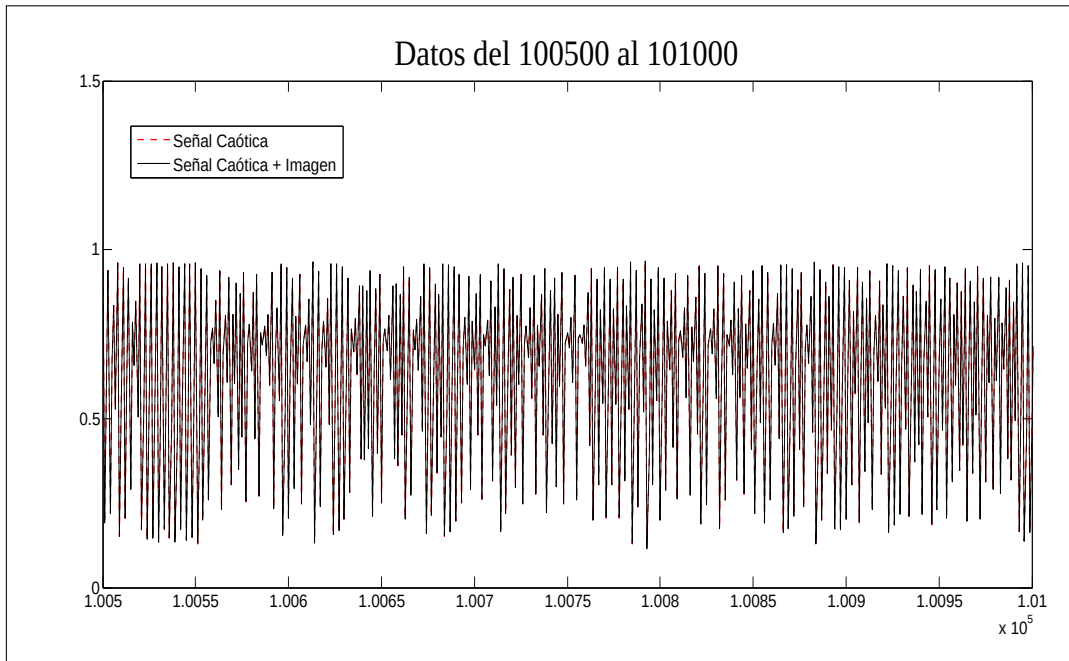


Figura 7.4: La imagen muestra los datos encriptados y el estado caótico.

En la figura 7.4 se puede observar que los datos de la imagen y de la señal caótica prácticamente son iguales, a tal punto que no se pueden distinguir una de la otra, siendo este hecho lo que se buscaba en este trabajo de tesis.

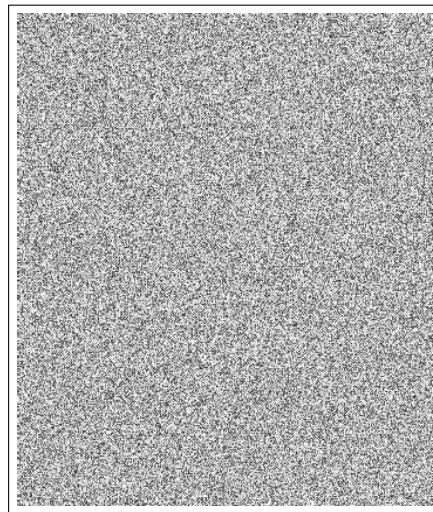


Figura 7.5: Imagen de la huella dactilar encriptada.

En la figura 7.5 se puede observar la imagen de la huella dactilar que se obtendría si alguna persona efectuara un intento de observar la imagen encriptada.

7.3. Descriptado de imágenes

El proceso de descriptado está compuesto por seis etapas principales, las cuales, se muestran en la figura 7.6. En la primera etapa se lee el archivo “txt” desde una dirección especificada, en la segunda etapa se generará un vector columna con los datos resultantes del estado caótico x_1 del mapeo de Rössler (5.3), con las mismas condiciones iniciales y parámetros que se utilizaron en el proceso de encriptado, en la tercera etapa se realiza la operación de restar el vector del estado caótico al vector de datos, el nuevo vector obtenido se multiplica por 1 000, dando como resultado el vector que contiene los bits de la imagen, en la cuarta etapa se reconstruye un vector de bytes utilizando el vector de bits obtenido en la etapa anterior, lo que se realiza es reacomodar los bits en conjuntos de 8 para reconstruir el vector de bytes que componen la imagen, en la quinta etapa se reacomodan los bytes en una matriz de dimensión 355×390 , con la cual, se obtiene la imagen reconstruida, en la última etapa se guarda la imagen recuperada. Para mejor comprensión del proceso observar la figura 7.6.

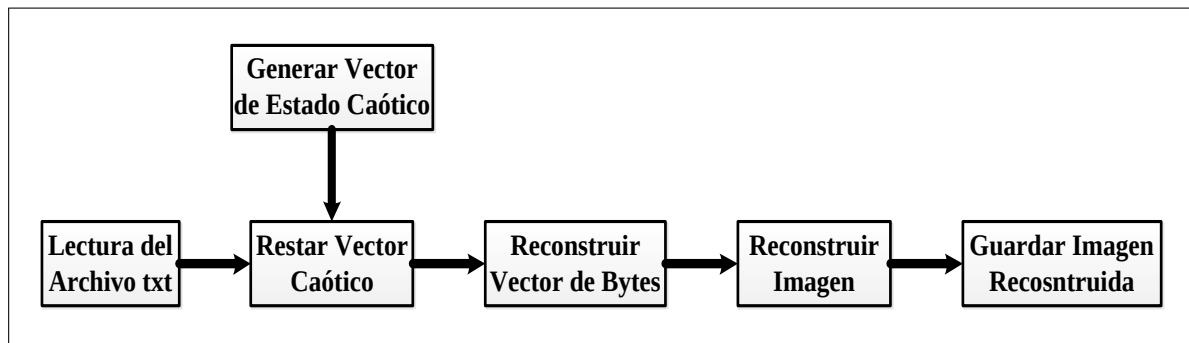


Figura 7.6: Diagrama de bloques del proceso de descriptado de imágenes de huellas dactilares.

A continuación en la figura 7.7 se muestra la imagen del pulgar izquierdo, que se obtiene después de utilizar el programa de descriptado implementado en MatLab (ver anexo C.2).



Figura 7.7: Imagen de la huella dactilar recuperada.

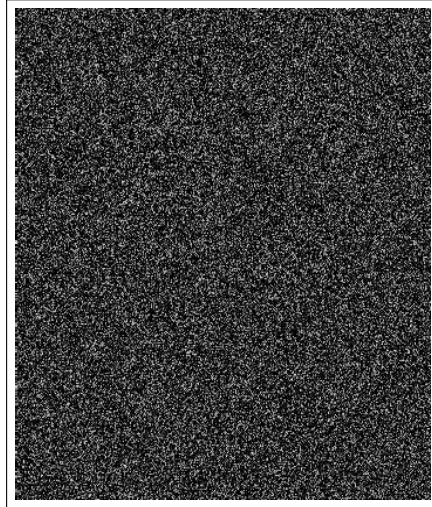


Figura 7.8: Imagen de la huella dactilar recuperada con condiciones iniciales distintas.

En la figura 7.8 se puede observar la imagen de la huella dactilar que se obtendría si alguna persona efectuará un descriptado de la imagen, con condiciones iniciales distintas a las utilizadas en el encriptado.

7.4. Comparación de las imágenes

La prueba de comparación por medio de histogramas que se realizó, tuvo como objetivo determinar el número de píxeles de la imagen, que podrían ser afectados durante el proceso de encriptado y descriptado. La prueba de comparación consiste en comparar una imagen del pulgar izquierdo previo al proceso de encriptado (ver figura 7.2) versus la imagen del pulgar izquierdo después de los procesos de encriptado y descriptado, dicho de otra manera la imagen recuperada (ver figura 7.7). El proceso que se realizó en la prueba se muestra en la figura 7.9.

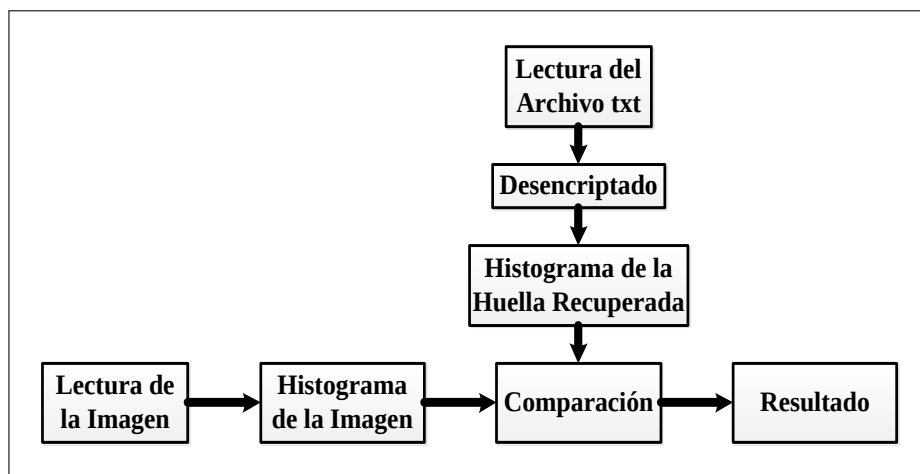


Figura 7.9: Diagrama del proceso de comparación de imágenes de huellas dactilares.

La comparación de histogramas de la huella recuperada versus la huella antes del proceso de encriptado, se realizó en un programa de MatLab versión R2008a (ver anexo C.3). A continuación se muestran en las siguientes figuras lo que se obtiene como resultado del programa. En la figura 7.10 se pueden observar los histogramas de las imágenes que se comparan, en la figura 7.11 se muestra el resultado de la comparación de histogramas y en la figura 7.12 se muestra la decisión que se toma de la comparación.

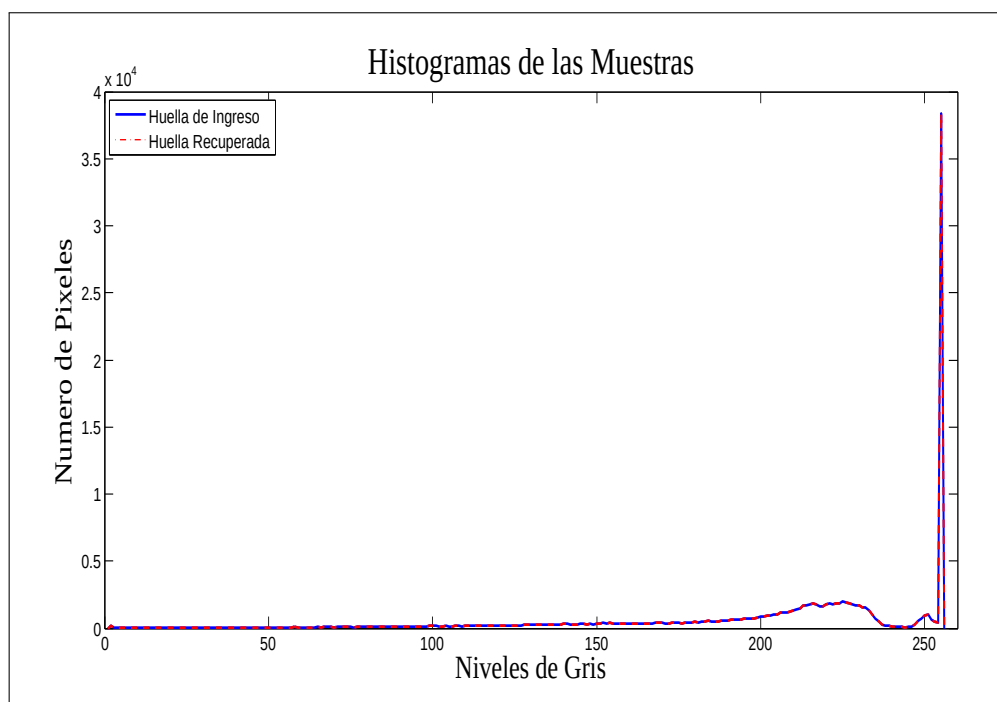


Figura 7.10: Histogramas utilizados en el programa del anexo C.3

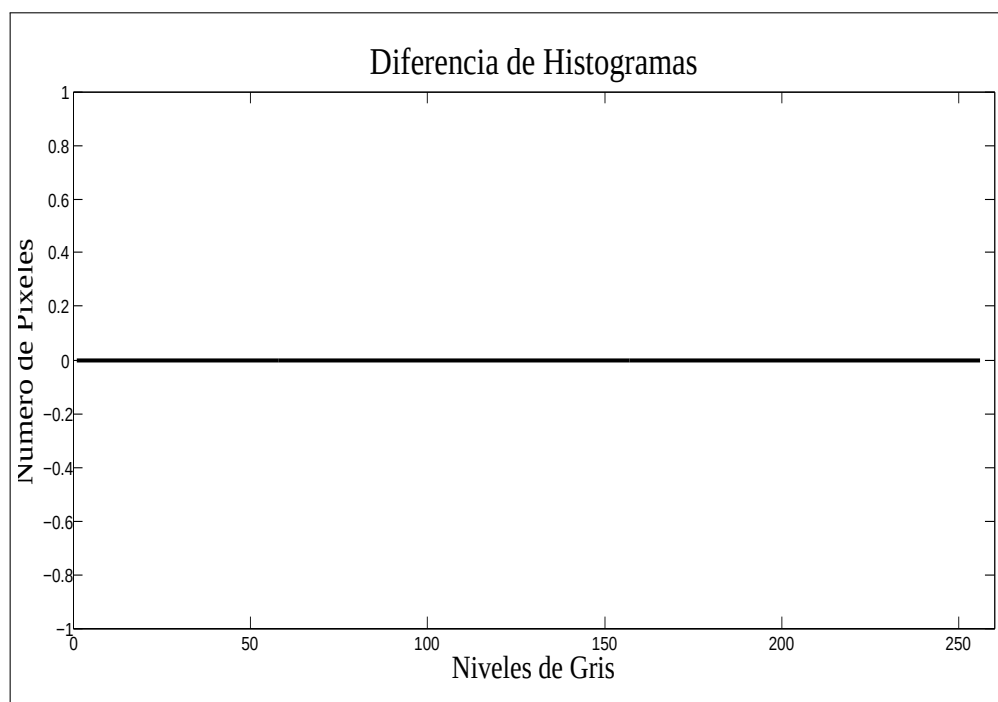


Figura 7.11: Histograma resultante de la comparación

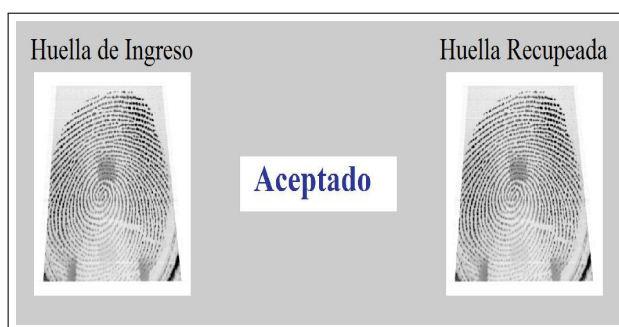


Figura 7.12: Decisión de la comparación del programa del anexo C.3.

7.5. Programas finales del sistema biométrico

A continuación se mostrarán tres programas que se desarrollaron con un entorno gráfico, los cuales, simulan las etapas de registro, verificación y identificación de un sistema biométrico de control de acceso con base a huellas dactilares. La realización de estos programas se decidió después de observar los resultados obtenidos de las pruebas anteriores, en las cuales, se puede observar que la comparación por medio de histogramas podría funcionar, así como también el proceso de encriptado y desencriptado mostrado en este capítulo.

7.5.1. Programa de registro

El objetivo del programa de registro es crear un archivo “txt”, en el cual, contendrá la información requerida para realizar la comparación en el programa de identificación o verificación. Lo que realiza este programa es tomar la imagen de una huella dactilar, para convertirla en un archivo “txt”, siendo este archivo utilizado para reconstruir la imagen. En la figura 7.13 se muestra la ventana con la que se inicia el programa, en la cual, se pueden observar cinco botones, con el primer botón se obtiene la imagen del usuario que pretende ser registrado, lo que ocurre al presionar el botón es que se despliega una ventana en donde se muestran las imágenes de las huellas de las personas que se registraran, posteriormente de seleccionar una huella, ésta se mostrará en alguna sección de la ventana de registro, con el segundo botón se transforma la imagen en un vector de datos, para posteriormente mostrar algunos de sus datos en forma gráfica, el tercer botón genera un vector, el cual, contiene los valores proporcionados por el estado x_1 del mapeo de Rössler (5.3), posterior a generar el vector se muestran algunos de sus valores, el cuarto botón realiza la acción de ocultar el vector de datos de la imagen en el vector de la señal caótica, después de realizar esta acción se muestran algunos de los valores obtenidos, con el quinto botón se despliega una ventana (ver figura 7.16), en la cual, se indicará la carpeta y el nombre con el que se guardará el vector resultante de la operación anterior. En la figura 7.15 se muestra la ventana del programa después de utilizar cada uno de los botones.

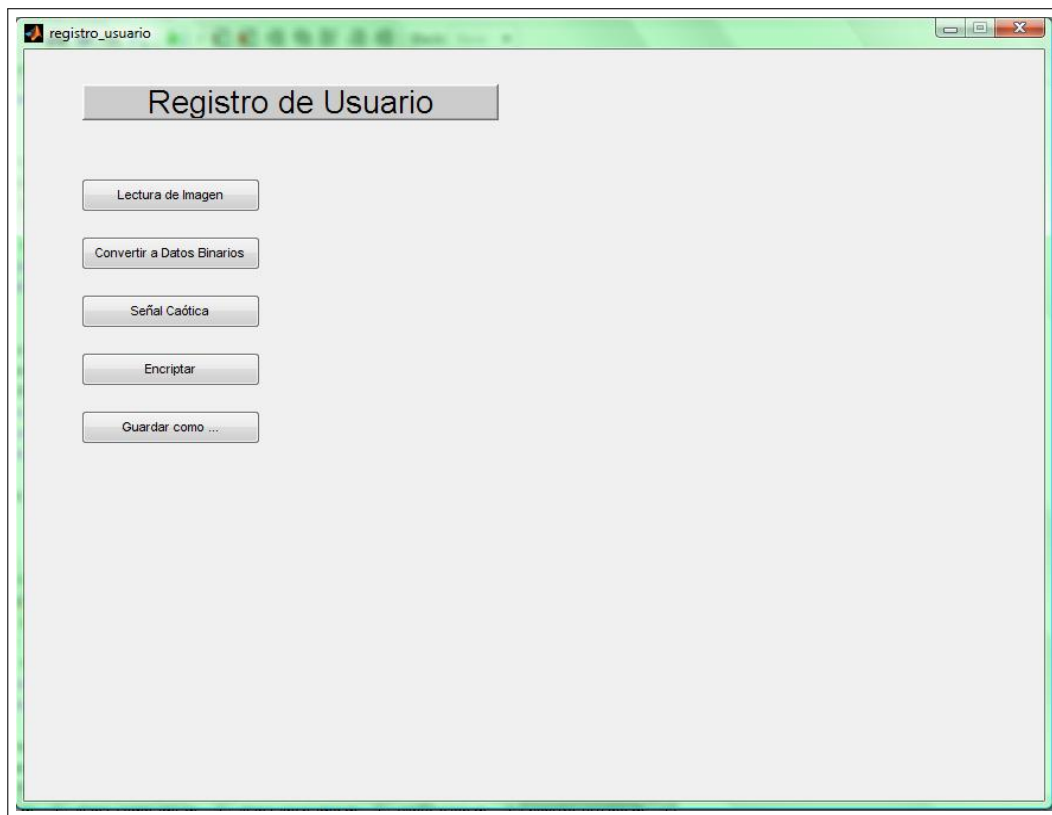


Figura 7.13: Ventana del programa al iniciar el proceso de registro.

Después de pulsar alguno de los botones, en la parte superior derecha de la ventana del programa, se desplegará una leyenda que dice “**Esperar**” con letras color rojo, la cual, se mantendrá de esa forma hasta terminar el proceso que se realizará al pulsar el botón, una vez concluido el proceso, la leyenda cambiará a letras color verde que dice “**Continuar**”.

El motivo de que la imagen de la huella se obtenga de una carpeta en donde previamente se almacenó después de adquirirla del lector, en lugar de adquirirla directamente del lector de huellas dactilares, es debido a que no fue posible utilizar el lector de huellas dactilares directamente con el programa de MatLab, por tal motivo al momento de presionar el botón de “**Lectura de imagen**” se despliega la ventana que se muestra en la figura 7.14.

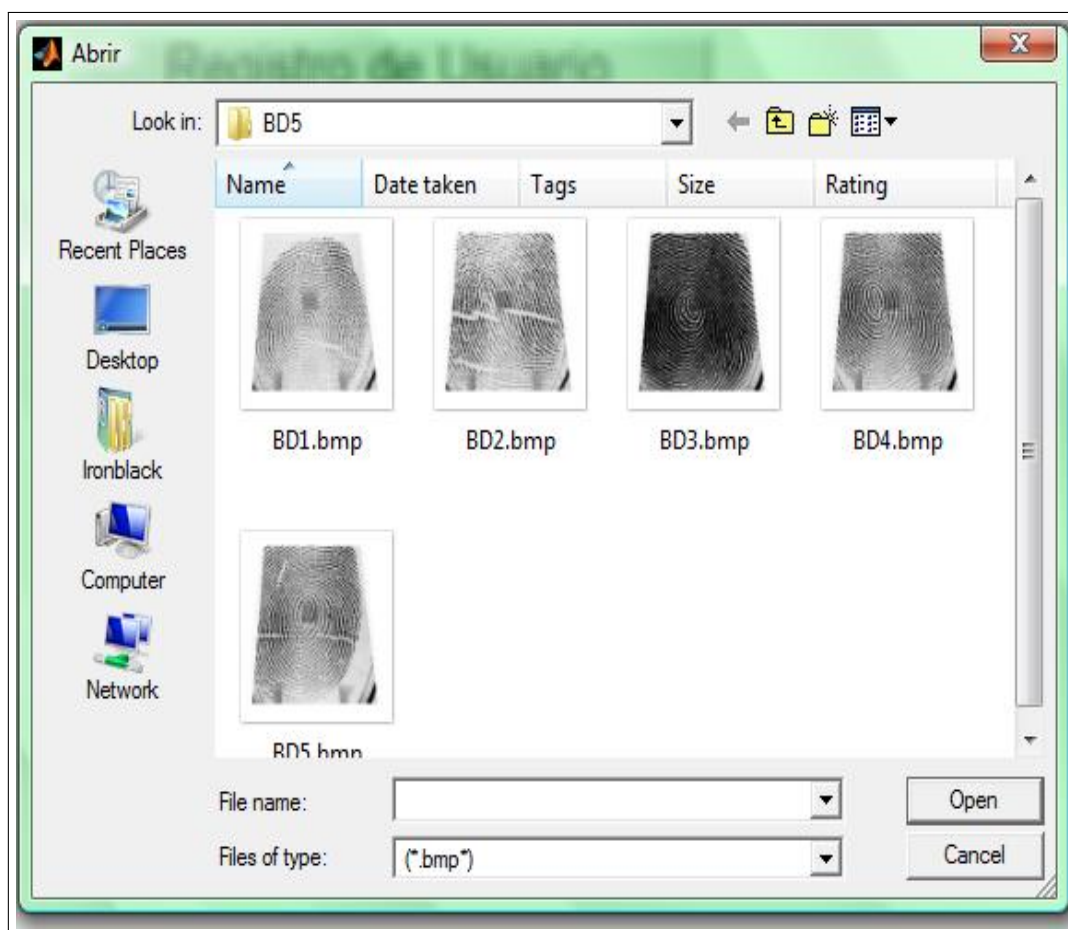


Figura 7.14: Ventana que se despliega al pulsar el botón “Lectura de imagen”.

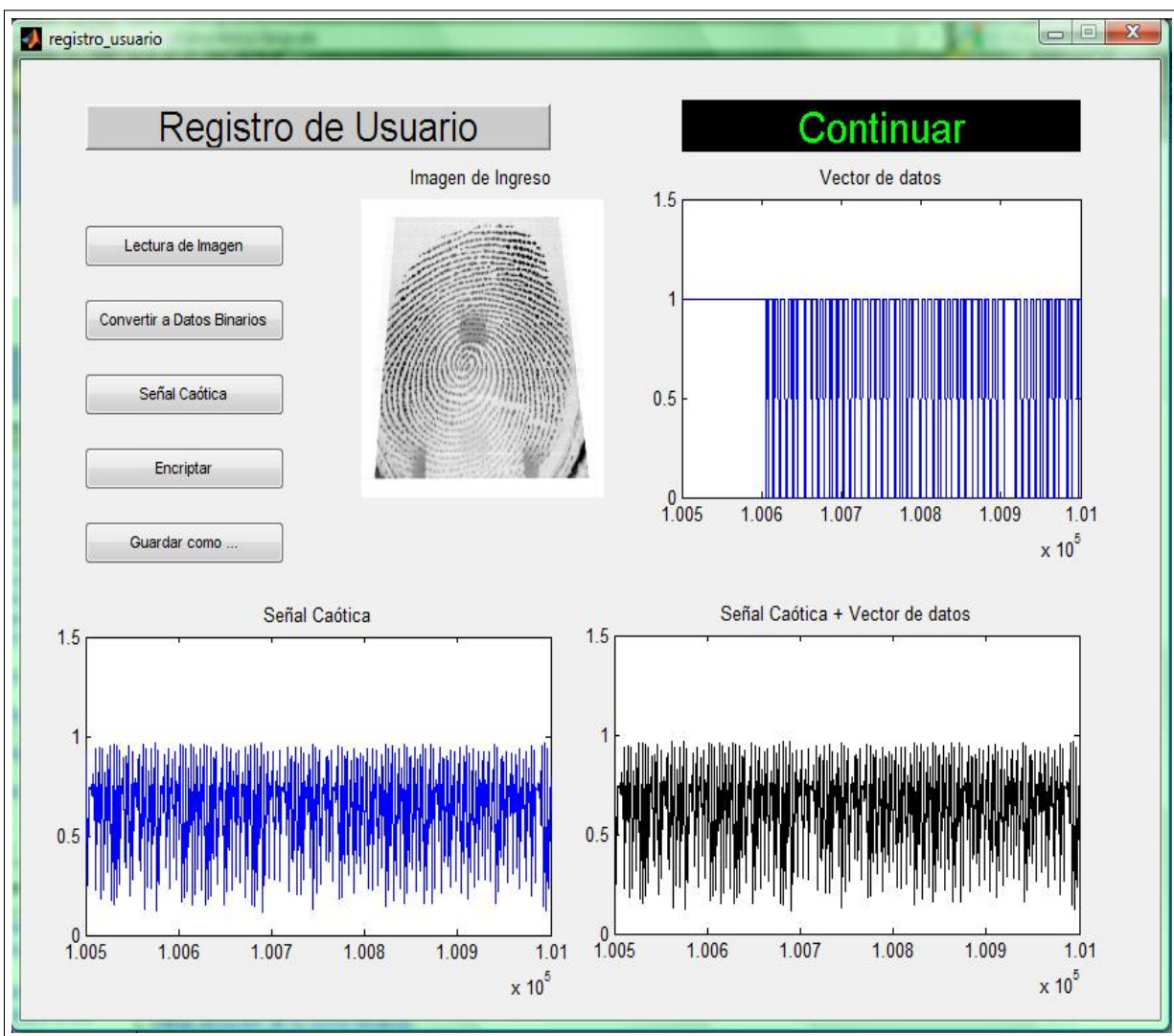


Figura 7.15: Ventana del programa de registro al finalizar el proceso.

En la figura 7.15 se puede observar que las gráficas están en la misma escala en el eje horizontal, esto fue con el motivo de que al momento de observar las gráficas se pueda apreciar, que información es la que se está ocultando. El código del programa se puede consultar en el anexo D.1.

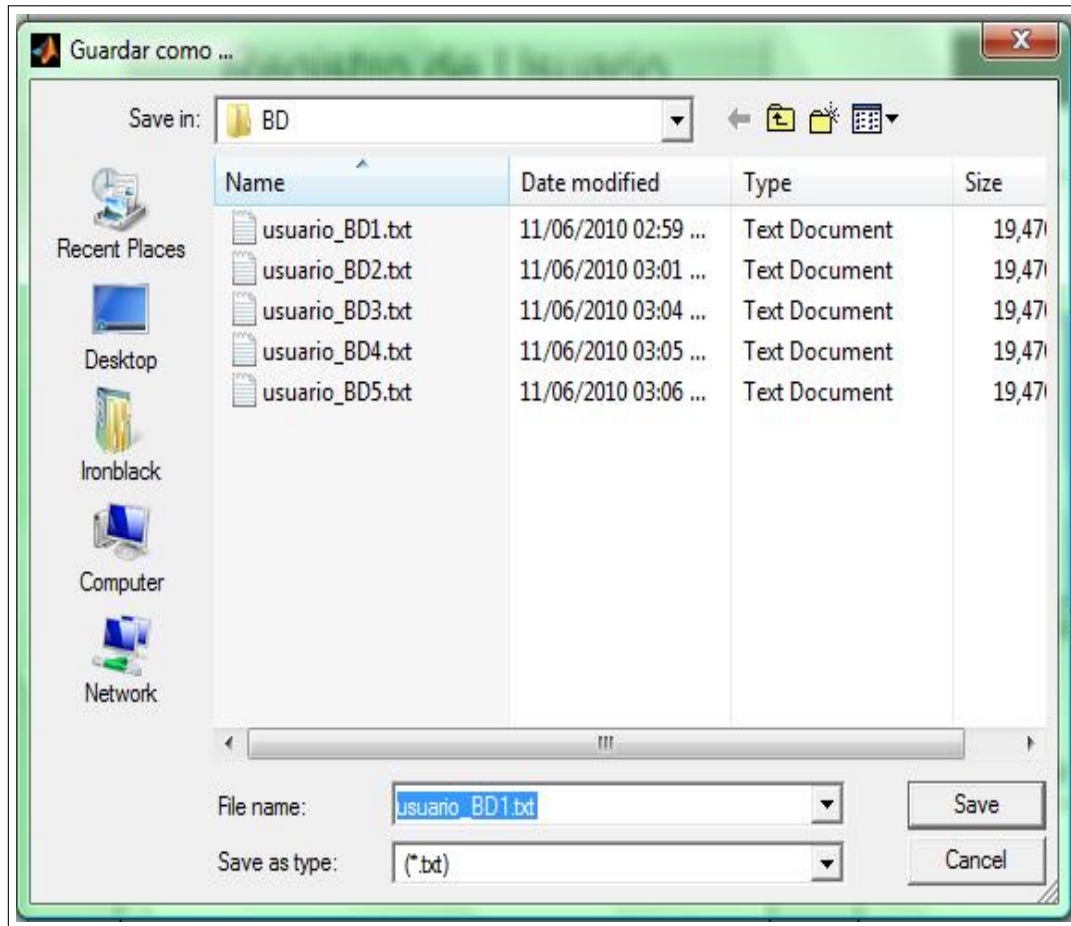


Figura 7.16: Ventana que se despliega al pulsar el botón “Guardar como ...”, en la cual, se puede observar que se encuentran registrados 5 usuarios.

7.5.2. Programa de verificación

El objetivo del programa, es verificar la existencia del registro de un usuario específico. Lo que realiza el programa es comparar el histograma de la imagen de ingreso versus el histograma de la imagen, de sólo uno de los usuarios registrados. En la figura 7.19 se muestra la ventana que se despliega al correr el programa, en esta ventana se pueden observar tres botones. Con el primer botón se obtiene la imagen del usuario registrado, lo que sucede al presionar el primer botón es que se despliega una ventana en donde se muestran los archivos “txt” de los usuarios registrados, al seleccionar un archivo el programa lo decodificará para convertirlo en una imagen, después la despliega en una sección de la ventana del programa. Con el segundo botón se obtiene la imagen de la huella del usuario que pretende ser reconocido por el programa, lo que sucede al presionar el segundo botón es que se despliega una ventana en donde se muestran las imágenes de la huellas de los posibles usuarios que pretenden ser reconocidos, después de seleccionar el programa lo muestra en una sección de la ventana de inicio, con el tercer botón se realiza la comparación de histogramas de las imágenes seleccionadas, con esta acción se obtienen como resultados los histogramas de las imágenes

y una leyenda que puede decir “**No Aceptado**” con letras color rojo o “**Aceptado**” con letras color verde. En la figura 7.20 se muestra la ventana del programa al finalizar todos los pasos que se realizan al presionar los tres botones.

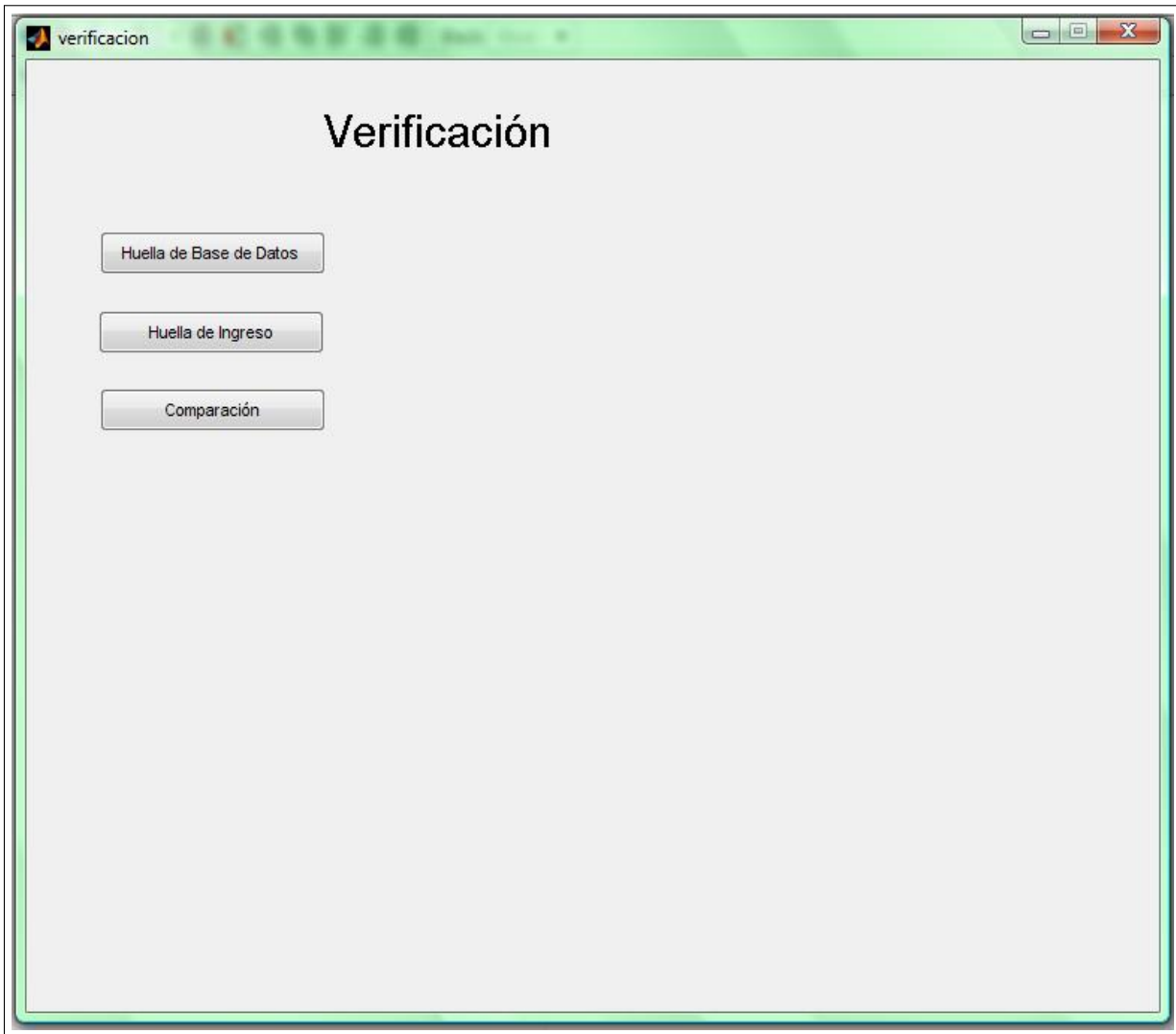


Figura 7.17: Ventana al iniciar el programa de verificación

Comentario: La ventana que se despliega al oprimir el botón “**Huella de Base de Datos**”, es similar a la ventana que se muestra en la figura 7.16, pero con la diferencia que en esta ocasión sirve para leer un archivo “txt”. También en la ventana de verificación aparece la leyenda “**Procesando**” en la parte superior derecha, cuando el programa se encuentra realizando alguna operación.

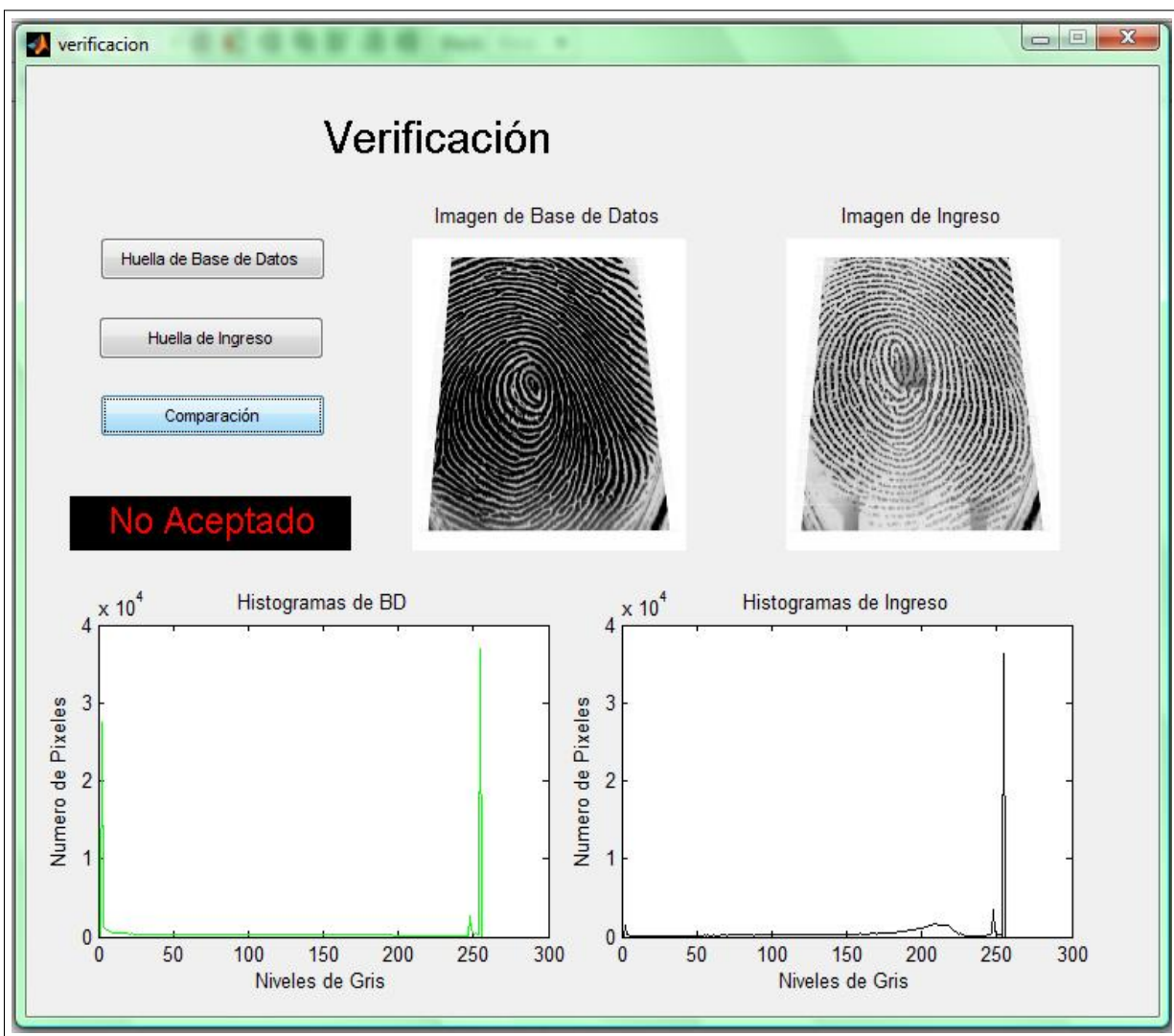


Figura 7.18: Ventana del programa de verificación al finalizar el proceso.

7.5.3. Programa de identificación

El objetivo del programa, es identificar al usuario que pretende ser reconocido por el sistema biométrico. Lo que realiza el programa es comparar el histograma de la imagen de ingreso versus los histogramas de las imágenes de cada uno de los usuarios registrados. En la figura 7.19 se muestra la ventana que se despliega al correr el programa, en esta ventana se pueden observar tres botones, el primer botón realiza la acción de cargar las variables que contendrán las imágenes de las huellas registradas, siendo estas variables indispensables para que el programa realice la comparación. Con el segundo botón se obtiene la imagen del usuario que pretende ser reconocido por el programa, lo que sucede al presionar el segundo botón, es que se despliega una ventana en donde se muestran las imágenes de la huellas de los posibles usuarios que pretenden ser reconocidos, después de seleccionar el programa lo muestra en la parte superior derecha de la ventana de inicio, con el tercer botón se realiza la

comparación de histogramas con el motivo de obtener como resultado una respuesta de si se encontró al usuario de ingreso, si el usuario es identificado se desplegaran los histogramas de las imágenes comparadas, las imágenes comparadas y una leyenda que dice “***Identificado***”, en caso contrario sólo se desplegará la leyenda que dice “***No identificado***”, en la figura 7.20 se puede observar la ventana del programa al ocurrir una identificación.

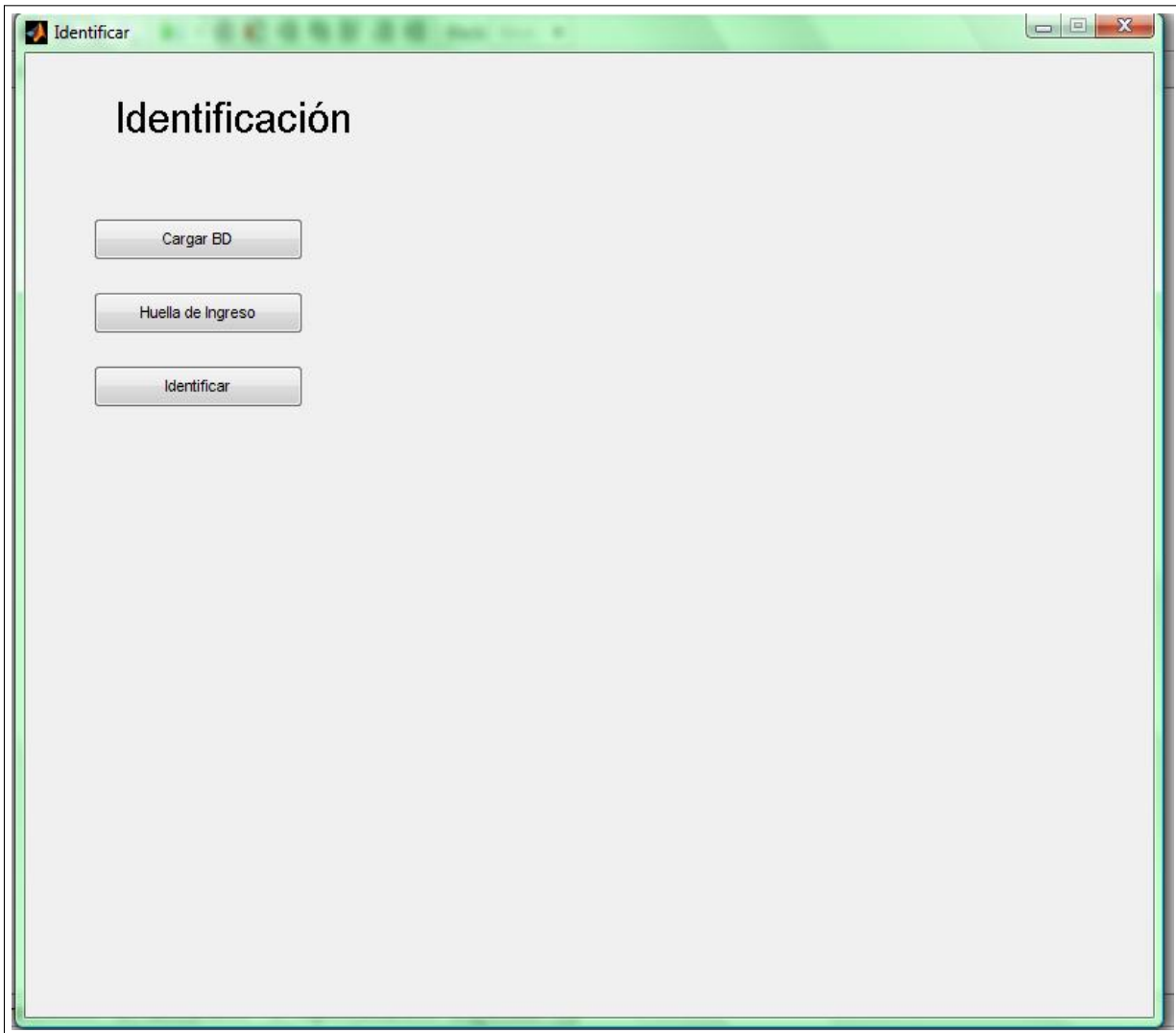


Figura 7.19: Ventana al iniciar el programa de identificación.



Figura 7.20: Ventana del programa de identificación al finalizar el proceso.

7.6. Conclusiones

- La acción de guardar la imagen encriptada como un archivo “txt”, le brida más seguridad a comparación de guardarla como un formato de imagen, por ejemplo supóngase que alguien adquiriera el archivo, pero al momento que intente averiguar que contiene el archivo lo único que encontrará en él serán números que para el individuo no tendrán significado, y no podrá identificar que se trata de una imagen.
- Los programas de registro, identificación y verificación, son lentos en el momento de convertir la imagen a vector de datos o viceversa, lo cual, podría causar incomodidad al usuario que utilice estos programas. Esto podría ser corregido si se buscara un método más rápido al utilizado en este trabajo de tesis.

Capítulo 8

Conclusiones generales y trabajo futuro

8.1. Conclusiones generales

El presente trabajo de tesis proporciona una contribución más a las comunicaciones seguras, en esta ocasión, se atendió la problemática de la seguridad en algunas líneas de comunicación de un sistema biométrico de control de accesos basado en huellas dactilares. Para dicho trabajo se consideró un sistema biométrico de control de accesos basado en huellas dactilares, tomando en consideración el escenario, en el cual, la etapa de registro, la etapa de almacenamiento (base de datos) y la etapa de identificación/verificación se encuentran separadas en sitios remotos. También, se omitieron algunos procesos debido a que su implementación en el programa de MatLab tomaría un largo tiempo, con el cual no se disponía, por tal motivo se propusieron algunas alternativas a estos procesos. Considerando lo anterior los resultados que nos proporcionó el trabajo de tesis son los siguientes:

- Se observó que considerando las ecuaciones de Rössler con las siguientes condiciones iniciales $x(0) = (0,3,0,0,05)$ y valores de parámetros $\alpha = 3,8$, $\beta = 0,05$, $\gamma = 0,35$, $\delta = 3,78$, $\varsigma = 0,2$, $\eta = 0,1$, $\theta = 1,9$. Presenta una dinámica hipercaótica, lo cual, fue benéfico para el propósito de cifrar la información biométrica con dicha dinámica. Esto se debe al hecho, que al ocultar la información en esta dinámica hipercaótica es menos propensa a ser visible a un observador no deseado.
- La alternativa de comparación, que se presentó en el trabajo de tesis, proporcionó buenos resultados al identificar las huellas dactilares, sin embargo estos resultados podrían mejorarse tomando en consideración dos o más parámetros en la identificación de huellas, por ejemplo, en las figuras del capítulo 6 donde se muestran los histogramas resultantes de la comparación, se mencionan que no sobrepasan cierta cantidad en el eje vertical de las gráficas, ese hecho podría considerarse para proporcionar una nueva condición en el proceso de la comparación, lo cual, podría mejorar los resultados.

- El encriptado y desencriptado de imágenes de huellas dactilares por medio de un programa implementado en MatLab, proporcionó buenos resultados. El cifrado y descifrado no afectó al proceso de comparación, siendo este hecho de gran importancia debido que esto nos indica que el cifrado presentado en esta tesis podría no afectar a los procesos que se realizan los sistemas biométricos convencionales.

8.2. Trabajos futuros

Las actividades futuras que podrían realizarse con respecto al tema principal del trabajo realizado en la tesis serían las siguientes:

- Realizar investigaciones más profundas con respecto a la alternativa de comparación por medio de histogramas, para analizar la viabilidad de su implementación en dispositivos comerciales.
- Realizar una investigación con respecto a los falsos rechazos y las falsas aceptaciones, utilizando la alternativa de comparación propuesta en esta tesis.
- En esta ocasión se realizó el encriptado por medio de adición, debido a esto se podría realizar una investigación con diferentes métodos de encriptado caóticos, con el motivo de analizar, cual es, más conveniente en los sistemas biométricos de control de accesos.
- Realizar el encriptado de la información biométrica, sin excluir ningún proceso del sistema biométrico convencionales. Es decir, con base a los patrones de las huellas. Esto haría consumir menos espacio de memoria.
- Realizar investigaciones con respecto a los ataques que sufren los sistemas biométricos comerciales. Pero en esta ocasión utilizando el encriptado propuesto en esta tesis, para analizar su efectividad a los posibles ataques que pueda sufrir.
- Llevar a cabo la implementación del encriptado caótico en un sistema biométrico comercial.

Anexo A

Programa en MatLab - Problema 1

Código del programa de comparación de histogramas para un usuario implementado en MatLab versión R2008a.

A.1. Resultado numérico

El siguiente programa proporciona como resultado los histogramas de las muestras y los resultados numéricos de la comparación de histogramas para un sólo usuario.

```
function problema_1
%% Declaración de variables
close all
N=5;
x=0:255;

%% Imagen de la Base de Datos.
IB=imread('E:\muestras\BD\usuarioregistrado1.bmp');
IBD=double(IB);

%% Cálculo del histograma de la huella de la base de datos
%% Inicialización de la variable de histograma
[r c]=size(IBD);
for i=1:256
    h(i)=0;
end

%% Cálculo de histogramas
for i=1:r
    for j=1:c
        k=IB(i,j);
        h(k+1)=h(k+1)+1;
    end
end
```

```
%% Graficado del histograma de la huella de la base de datos
```

```
figure;plot(x,h);
xlabel('\Niveles de gris')
ylabel('\Número de pixeles')
title('\Histograma de la base de datos')
legend('\Huella de la base de datos')
```

```
%% Lectura de imágenes de las huellas de las personas que solicitan la verificación
```

```
for i=1:N
    D=strcat('E:\muestras\musuarios\usuario',int2str(i));
    direccion=strcat(D,'\.bmp');    % Concatena dos cadenas las cuales.
                                    % Forman la dirección de la imagen.
    eval('img=imread(direccion);'); % Lee la imagen en la direccion str.
    im(:, :, i)=img;
    imD(:, :, i)=double(img);
end
```

```
%% Cálculo de histogramas de las muestras de las personas
```

```
%% Inicialización de la variable de histogramas
```

```
[r c]=size(imD(:, :, 1));
for i=1:256
    for j=1:N
        h1(i,j)=0;
    end
end
```

```
%% Cálculo de histogramas
```

```
for L=1:N
    for i=1:r
        for j=1:c
            k1=im(i,j,L);
            h1(k1+1,L)=h1(k1+1,L)+1;
        end
    end
end
```

```
%% Graficado del conjunto de muestras a comparar
```

```
figure;plot(x,h1(:,1));
hold on
    plot(x,h1(:,2),\k-')
    plot(x,h1(:,3),\r')
    plot(x,h1(:,4),\b-.)
    plot(x,h1(:,5),\k')
hold off
xlabel('\Niveles de gris')
```

```
ylabel('Numero de pixeles')
title('Histogramas de las muestras')
legend('Persona 1','Persona 2','Persona 3','Persona 4','Persona 5')
```

```
%% Proceso para el cálculo de error
```

```
% Cálculo de diferencia de histogramas
```

```
for i=1:N
    for p=1:256
        if h1(p,i)>h(p)
            E1(p,i)=h1(p,i)-h(p);
        else
            E1(p,i)=h(p)-h1(p,i);
        end
    end
end
end
```

```
% Calculo del Error
```

```
for i=1:N
    sume1(i)=(sum(E1(:,i)));
end
```

```
%% Despliega la variable sume1 en command window
```

```
disp('Diferencia')
sume1
```

A.2. Resultado gráfico

Al agregar el siguiente código en el programa, se generan como resultado un cuadro de figura, el cual, contiene las huellas comparadas y la leyenda de aceptado o no aceptado. Además, genera otros dos cuadros de figura en uno contiene los histogramas que se comparan y en el otro el histograma resultante de la diferencia.

```
%% Comparación
```

```
for i=1:N
    if sume1(i)<16000
        % Despliega en command window la leyenda “Aceptado persona # ”
        disp(strcat('Aceptado persona # ',int2str(i)))
        % Despliega una ventana con las huellas comparadas y una leyenda de Aceptado
        figure;subplot(1,3,1);imshow(IB)
        title('Huella de la base de datos')
        subplot(1,3,2);imshow(imread('C:\Users\Ironblack\Pictures\aceptado.jpg'))
        subplot(1,3,3);imshow(im(:, :, i))
        title(strcat('Huella aceptada de la persona ',int2str(i)))
        % Despliega una ventana con los histogramas comparados
        figure;plot(x,h,'b')
```

```

hold on
plot(x,h1(:,i),'r-')
hold off
xlabel('Niveles de gris')
ylabel('Número de pixeles')
title('Histogramas de las muestras')
legend('Huella de base de datos',strcat(' Persona ',int2str(i)));
% Despliega una ventana con el histograma de la diferencia de comparación
figure;plot(x,E1(:,i),'k')
xlabel('Niveles de gris')
ylabel('Número de pixeles')
title('Diferencia de histogramas')
else
% Despliega en command window la leyenda "No aceptado persona #"
disp(strcat('No aceptado persona # ',int2str(i)))
% Despliega una ventana con las huellas comparadas y una leyenda de No aceptado
figure;subplot(1,3,1);imshow(IB)
title('Huella de la base de datos')
subplot(1,3,2);imshow(imread('C:\Users\Ironblack\Pictures\noaceptado.jpg'))
subplot(1,3,3);imshow(im(:, :, i))
title(strcat('Huella no aceptada de la persona ',int2str(i)))
% Despliega una ventana con los histogramas comparados
figure;plot(x,h,'b')
hold on
plot(x,h1(:,i),'r-')
hold off
xlabel('Niveles de gris')
ylabel('Número de pixeles')
title('Histogramas de las muestras')
legend('Huella de base de datos',strcat('Persona ',int2str(i)));
% Despliega una ventana con el histograma de la diferencia de comparación
figure;plot(x,E1(:,i),'k')
xlabel('Niveles de gris')
ylabel('Número de pixeles')
title('Diferencia de histogramas')
end
end
end

```

Anexo B

Programa en MatLab - Problema 2

Código del programa de comparación de histogramas desarrollado para el problema de comparación de múltiples usuarios, el cual se realizó en MatLab versión R2008a.

B.1. Resultado numérico

El siguiente código del programa sólo proporciona como resultado los histogramas de las muestras que se utilizaron, también como resultado de la comparación de histogramas, proporciona de forma numérica el total de los pixeles distintos en las imágenes.

```
function problema_2
%% % Declaración de variables
close all
NBD=5;
NM=20;
x=0:255;
%% % Usuarios registrados en la base de datos
% Lectura de La Base de Datos
for i=1:NBD
    img=imread(strcat('E:\muestras\BD5\BD',int2str(i),'.bmp'));
    im(:, :, i)=img;
end
% Inicialización de la variable de histogramas
[r c]=size(im(:, :, 1));
for i=1:256
    for j=1:NBD
        h(i,j)=0;
    end
end

% Cálculo de histogramas
```

```

for L=1:NBD
    for i=1:r
        for j=1:c
            k=im(i,j,L);
            h(k+1,L)=h(k+1,L)+1;
        end
    end
end
% Graficado de histogramas de la base de datos
figure;plot(x,h);
xlabel('Niveles de gris')
ylabel('Número de pixeles')
title('Histogramas de la base de datos')
% Leyenda de los gráficos
for i=1:NBD
    numero=strcat('Usuario ',int2str(i));
    his(i,:)=strcat(numero,'Registrado');
end
legend(his)
%% Huellas de las personas que solicitan la identificación
for i=1:NM
    img=imread(strcat('E:\muestras\musuarios5\usuario5_',int2str(i),'.bmp'));
    im1(:, :, i)=img;
    imD1(:, :, i)=double(img);
end
% Inicialización de la variable de histogramas
[r c]=size(imD1(:, :, 1));
for i=1:256
    for j=1:NM
        h1(i,j)=0;
    end
end
% Cálculo de histogramas
for L=1:NM
    for i=1:r
        for j=1:c
            k=im1(i,j,L);
            h1(k+1,L)=h1(k+1,L)+1;
        end
    end
end
end

% Gráfica de los histogramas de las personas del 1 - 10

```

```

figure;plot(x,h1(:,1));
hold on
plot(x,h1(:,2),'m')
plot(x,h1(:,3),'g')
plot(x,h1(:,4),'r')
plot(x,h1(:,5),'k')
plot(x,h1(:,6),'m- -')
plot(x,h1(:,7),'b- -')
plot(x,h1(:,8),'k- -')
plot(x,h1(:,9),'g- -')
plot(x,h1(:,10),'r- -')
hold off
xlabel('\Niveles de gris')
ylabel('\Numero de pixeles')
title('\Histogramas de las muestras')
% Leyenda de las gráficas
for i=1:10
    if i<10
        numero=strcat(' 00',int2str(i));
    else
        numero=strcat(' 0',int2str(i));
    end
    his1(i,:)=strcat(' Persona ',numero);
end
legend(his1)
% Gráfica de los histogramas de las personas del 11 - 20
figure;plot(x,h1(:,11));
hold on
plot(x,h1(:,12),'m')
plot(x,h1(:,13),'g')
plot(x,h1(:,14),'r')
plot(x,h1(:,15),'k')
plot(x,h1(:,16),'m- -')
plot(x,h1(:,17),'b- -')
plot(x,h1(:,18),'k- -')
plot(x,h1(:,19),'g- -')
plot(x,h1(:,20),'r- -')
hold off
xlabel('\Niveles de gris')
ylabel('\Número de pixeles')
title('\Histogramas de las muestras')

```

```

% Leyenda de las gráficas

```



```

for i=11:NM
    if i<10
        numero=strcat(' 00',int2str(i));
    else
        numero=strcat(' 0',int2str(i));
    end
    his2(i-10,:)=strcat(' Persona ',numero);
end
legend(his2)
%% Cálculo de error
% Cálculo de pixeles distintos en los niveles de gris
for i=1:NBD
    for j=1:NM
        for p=1:256
            if h1(p,j)>h(p,i)
                E(p,i,j)=h1(p,j)-h(p,i);
            else
                E(p,i,j)=h(p,i)-h1(p,j);
            end
        end
    end
end
% Cálculo de pixeles distintos totales
for i=1:NBD
    for j=1:NM
        sume(i,j)=(sum(E(:,i,j)));
    end
end
%% Desplegar resultados numéricos en Command Windows
sume'
```

B.2. Resultado gráfico

Al agregar el siguiente código en el programa, se generan como resultado una ventana de figura, la cual, contiene las huellas comparadas y la leyenda de aceptado o no aceptado. También se generan ventanas con los histogramas a comparar y el histograma resultante de la comparación.

```

%% Comparación
for i=1:NBD
    for j=1:NM
        if sume(i,j)<16000
% Código para las personas aceptadas
            figure;subplot(1,3,1);imshow(im(:, :, i))
            title(strcat('Huella de la base de datos ',int2str(i)))
            subplot(1,3,2);imshow(imread('C:\Users\Ironblack\Pictures\aceptado.jpg'))
```

```

subplot(1,3,3);imshow(im1(:,:,j))
title(strcat('Huella aceptada de la persona ',int2str(j)))
% histogramas de comparación
figure;plot(x,h(:,i),'b')
hold on
plot(x,h1(:,j),'r-')
hold off
xlabel('Niveles de gris')
ylabel('Número de pixeles')
title('Histogramas de las muestras')
legend(strcat('Huella de base de datos ',int2str(i)),strcat(' Persona ',int2str(j)));
% Histograma de diferencia
figure;plot(x,E1(:,i,j),'k')
xlabel('Niveles de gris')
ylabel('Número de pixeles')
title('Diferencia de histogramas')
else
% Código para personas no aceptadas
figure;subplot(1,3,1);imshow(im(:,:,i))
title(strcat('Huella de la base de datos ',int2str(i)))
subplot(1,3,2);imshow(imread('C:\Users\Ironblack\Pictures\noaceptado.jpg'))
subplot(1,3,3);imshow(im1(:,:,j))
title(strcat('Huella no aceptada del usuario ',int2str(j)))
% histogramas de comparación
figure;plot(x,h(:,i),'b')
hold on
plot(x,h1(:,j),'r-')
hold off
xlabel('Niveles de gris')
ylabel('Número de pixeles')
title('Histogramas de las muestras')
legend(strcat('Huella de base de datos ',int2str(i)),strcat(' Persona ',int2str(j)));
% Histograma de diferencia
figure;plot(x,E1(:,i,j),'k')
xlabel('Niveles de gris')
ylabel('Número de pixeles')
title('Diferencia de histogramas')
end
end
end

```

Nota: Cuando se agrega el código completo se generan más de 100 ventanas de figuras por lo que MatLab al llegar al número 100 deja de desplegar las ventanas y manda un mensaje de error. Por tal motivo es recomendable sólo desplegar a los usuarios aceptados.

Anexo C

Programas en MatLab de encriptado, descriptado y comparación de imágenes

Los siguientes códigos de programas en conjunto encriptan, descriptan y comparan la imagen de un pulgar izquierdo.

C.1. Código de encriptado

Código del programa en MatLab que encripta una imagen, la cual, guarda como un archivo “.txt”, sin embargo este programa requiere dos subprogramas, los cuales, se mostraran por separado.

```
function encri_img
%% Lectura de la imagen
imagen=imread('E:\muestras\BD\usuarioregistrado1.bmp');
figure; imshow(imagen)
title('Imagen a Encriptar');
%% Transformar en vector de bits
[im]=Convertirtobinario2(imagen);
[x y]=size(im);
imb=im';
tx=1:x;
figure;plot(tx,imb)
title('Vector de Datos de la Imagen');
axis([100500,101000,0,1.5])
%% Sistema caótico
[xa xb xc]=senal_caos2(0.3,0,0.05);
caos=xa';
figure;plot(tx,caos,'r')
```

```

title('Señal Caótica');
axis([100500,101000,0,1.5])
%% Operación de encriptado
im_encrip=(imb/1000)+caos;
figure;plot(tx,im_encrip,'k')
title('Señal Encriptada');
axis([100500,101000,0,1.5])
% Despliega la señal encriptada junto a la señal caótica
figure;plot(tx,caos,'r')
hold on
plot(tx,im_encrip,'k')
hold off
title('Señal Caótica y Señal Encriptada');
axis([100500,101000,0,1.5])
legend('Señal Caótica','Señal Encriptada')
%% Almacenamiento del archivo
save('E:\muestras\BD\im_encripBD.txt','im_encrip','-ASCII')

```

C.2. Código de desencriptado

Código del programa en MatLab que desencripta una imagen, la cual, se encuentra guarda como un archivo “.txt”.

```

function Imagen=etapa_desencri_img
%% Lectura del archivo “txt”
im_encr=load('E:\muestras\BD\im_encripBD.txt');
%% Generar vector de estado caótico
[xa xb xc]=senal_caos2(0.3,0,0.05);
caos=xa';
%% Desencriptado de la imagen
vector_datos=((im_encr-caos)*1000);
%% Conversión del vector de datos a imagen
Imagen=conv_dat_to_im2(vector_datos);

```

C.3. Código de comparación de las imágenes

El siguiente programa compara el histograma de una imagen recuperada después del proceso de desencriptado versus la misma imagen antes del proceso de encriptado.

```

function etapa_comp_imgs
%% Declaracion de variables

```

```

x=0:255;
hBD= [1:256]*0;
hrec= [1:256]*0;
%% Imagen de la Base de Datos.
IBD=imread('E:\muestras\BD\yo.bmp');
%% Cálculo del histograma de la huella de la base de datos
[r c] =size(IBD);
for i=1:r
    for j=1:c
        k=IBD(i,j);
        hBD(k+1)=hBD(k+1)+41;
    end
end
%% Imagen recuperada
% Lectura del archivo txt
im_encr=load('E:\muestras\BD\im_encripBD.txt');
% Generar vector de estado caótico
[xa xb xc] =senal_caos2(0.3,0,0.05);
caos=xa';
% Desencryptado de la imagen
vdatos=((im_encr-caos)*1000);
% Covertir vector de datos a imagen
im_desencr=conv_dat_to_im2(vdatos);
%% Cálculo del histograma de la huella recuperada
[r c] =size(im_desencr);
for i=1:r
    for j=1:c
        k=im_desencr(i,j);
        hrec(k+1)=hrec(k+1)+1;
    end
end
%% Obtencion de la diferencia de histogramas
for p=1:256
    if hBD(p)>hrec(p)
        dif(p)=hBD(p)-hrec(p);
    else
        dif(p)=hrec(p)-hBD(p);
    end
end
diferencia=sum(dif);
%% Comparación por medio de un umbral de 16000
if diferencia<16000
    figure;subplot(1,3,1);imshow(IBD)
    title('Huella de La Base de Datos')
    subplot(1,3,2);imshow(imread('C:\Users\Ironblack\Pictures\aceptado.jpg'))
    subplot(1,3,3);imshow(im_desencr)

```

```

title('Huella aceptada')
figure;plot(x,hBD,'b')
hold on
plot(x,hrec,'r-.')
hold off
xlabel('Niveles de Gris')
ylabel('Número de Pixeles')
title('Histogramas de las Muestras')
legend('Huella de Base de Datos','Huella Recuperada');
figure;plot(x,dif,'k')
xlabel('Niveles de Gris')
ylabel('Numero de Pixeles')
title('Diferencia de Histogramas')
end

```

Anexo D

Programas finales del sistema biométrico

D.1. Programa de registro de usuarios

```
function varargout = registro_usuario(varargin)
% REGISTRO_USUARIO M-file for registro_usuario.fig
% REGISTRO_USUARIO, by itself, creates a new REGISTRO_USUARIO or raises the existing
% singleton*.
%
% H = REGISTRO_USUARIO returns the handle to a new REGISTRO_USUARIO or the handle
% to the existing singleton*.
%
% REGISTRO_USUARIO('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in REGISTRO_USUARIO.M with the given input arguments.
%
% REGISTRO_USUARIO('Property','Value',...) creates a new REGISTRO_USUARIO or raises
% the existing singleton*. Starting from the left, property value pairs are applied to the GUI
% before registro_usuario_OpeningFcn gets called. An unrecognized property name or invalid
% value makes property application stop.All inputs are passed to registro_usuario_OpeningFcn
% via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one instance to
% run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help registro_usuario
% Last Modified by GUIDE v2.5 08-Jun-2010 00:17:28

% Begin initialization code – DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
```

```

    'gui_OpeningFcn', @registro_usuario_OpeningFcn, ...
    'gui_OutputFcn', @registro_usuario_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout1:nargout] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code – DO NOT EDIT

% — — — Executes just before registro_usuario is made visible.
function registro_usuario_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved – to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to registro_usuario (see VARARGIN)
% Choose default command line output for registro_usuario
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% — — — Outputs from this function are returned to the command line.
function varargout = registro_usuario_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;

% — — — Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
[nom,path] = uigetfile('*.*bmp*', 'Abrir', 'E:\muestras\BD5\');
if isequal(nom,0)
    return
end
imagen=imread([path,nom]);
imshow(imagen,'parent',handles.axes1)

```



```
handles.imagen=imagen;
guidata(hObject,handles);
title(handles.axes1,'Imagen de Ingreso')
```

```
% --- Executes on button press in pushbutton3.
```

```
function pushbutton3_Callback(hObject, eventdata, handles)
set(handles.text3,'String','Esperar','fontsize',22,'backgroundcolor',[0 0 0],'foregroundcolor',...
    [1 0 0],'visible','on')
pause(2)
imagen=handles.imagen;
datos=Convertirtobinario(imagen);
plot(datos,'parent',handles.axes2)
axis(handles.axes2,[100500,101000,0,1.5])
title(handles.axes2,'Vector de datos')
set(handles.text3,'String','Continuar','fontsize',22,'backgroundcolor',[0 0 0],'foregroundcolor',...
    [0 1 0],'visible','on')
handles.datos=datos;
guidata(hObject,handles);
```

```
% --- Executes on button press in pushbutton5.
```

```
function pushbutton5_Callback(hObject, eventdata, handles)
set(handles.text3,'String','Esperar','fontsize',22,'backgroundcolor',[0 0 0],'foregroundcolor',...
    [1 0 0],'visible','on')
pause(2)
[caosx1 caosx2 caosx3]=senal_caos2(0.3,0,0.05);
plot(caosx1,'parent',handles.axes3)
axis(handles.axes3,[100500,101000,0,1.5])
title(handles.axes3,'Señal Caótica')
set(handles.text3,'String','Continuar','fontsize',22,'backgroundcolor',[0 0 0],'foregroundcolor',...
    [0 1 0],'visible','on')
handles.caosx1=caosx1;
guidata(hObject,handles);
```

```
% --- Executes on button press in pushbutton6.
```

```
function pushbutton6_Callback(hObject, eventdata, handles)
set(handles.text3,'String','Esperar','fontsize',22,'backgroundcolor',[0 0 0],'foregroundcolor',...
    [1 0 0],'visible','on')
pause(2)
num=1000; huelladat=handles.datos;
scaos=handles.caosx1;
caos=scaos';
im_encrip=(huelladat/num)+caos;
plot(im_encrip,'k','Parent',handles.axes6)
axis(handles.axes6,[100500,101000,0,1.5])
title('Señal Caótica + Vector de datos')
```

```

set(handles.text3,'String','Continuar','fontsize',22,'backgroundcolor',[0 0 0],'foregroundcolor',...
    [0 1 0],'visible','on')
handles.im_encrip=im_encrip;
guidata(hObject,handles);

% --- Executes on button press in pushbutton7.
function pushbutton7_Callback(hObject, eventdata, handles)
usuario_BD=handles.im_encrip;
[nom, dir] = uiputfile('E:\huellas encriptadas\BD\usuario_BD1.txt','Guardar como ...');
if isequal(nom,0)
    return
end
save([dir nom],\usuario_BD','-ASCII')

```

D.2. Programa de verificación de usuario

```

function varargout = verificacion(varargin)
% VERIFICACION M-file for verificacion.fig
% VERIFICACION, by itself, creates a new VERIFICACION or raises the existing singleton*.
%
% H = VERIFICACION returns the handle to a new VERIFICACION or the handle
% to the existing singleton*.
%
% VERIFICACION('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in VERIFICACION.M with the given input arguments.
%
% VERIFICACION('Property','Value',...) creates a new VERIFICACION or raises
% the existing singleton*. Starting from the left, property value pairs are applied to the GUI
% before verificacion_OpeningFcn gets called. An unrecognized property name or invalid
% value makes property application stop.All inputs are passed to verificacion_OpeningFcn
% via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one instance to
% run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help verificacion
% Last Modified by GUIDE v2.5 12-Jun-2010 16:49:59

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @verificacion_OpeningFcn, ...

```

```

    'gui_OutputFcn', @verificacion_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if narginout
    [varargout1:nargout] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code – DO NOT EDIT

% — — — Executes just before verificacion is made visible.
function verificacion_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved – to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to verificacion (see VARARGIN)
% Choose default command line output for verificacion
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% — — — Outputs from this function are returned to the command line.
function varargout = verificacion_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;

% — — — Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
set(handles.text3,'String',' ', 'fontsize',18, 'backgroundcolor',[0.941 0.941 0.941], 'foregroundcolor',...
    [0.941 0.941 0.941], 'visible', 'on')
set(handles.text6,'String',' ', 'fontsize',18, 'backgroundcolor',[0.941 0.941 0.941], 'foregroundcolor',...
    [0.941 0.941 0.941], 'visible', 'on')
[nom,path] = uigetfile('*.txt*', 'Abrir', 'E:\huellas encriptadas\BD\');

```

```

if isequal(nom,0)
    return
end
set(handles.text6,'String','Procesando','fontsize',22,'backgroundcolor',[0 0 0],'foregroundcolor',...
    [1 0 0],'visible','on')
pause(2)
im_encr=load([path nom]);
[caosx1 caosx2 caosx3]=senal_caos2(0.3,0,0.05);
caos=caosx1';
im_desencr=((im_encr-caos)*1000);
imagen_r=conv_dat_to_im2(im_desencr);
imshow(imagen_r,'parent',handles.axes1)
title(handles.axes1,'Imagen de Base de Datos')
set(handles.text6,'String','Continuar','fontsize',22,'backgroundcolor',[0 0 0],'foregroundcolor',...
    [0 1 0],'visible','on')
handles.imagen_r=imagen_r;
guidata(hObject,handles);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
set(handles.text3,'String',' ','fontsize',18,'backgroundcolor',[0.941 0.941 0.941],'foregroundcolor',...
    [0.941 0.941 0.941],'visible','on')
[nom,path]=uigetfile('*.bmp*','Abrir','E:\muestras\musuarios5\');
if isequal(nom,0)
    return
end
set(handles.text6,'String','Procesando','fontsize',22,'backgroundcolor',[0 0 0],'foregroundcolor',...
    [1 0 0],'visible','on')
pause(2)
imagen=imread([path,nom]);
imshow(imagen,'parent',handles.axes2)
title(handles.axes2,'Imagen de Ingreso')
set(handles.text6,'String','Continuar','fontsize',22,'backgroundcolor',... [0 0 0],'foregroundcolor',...
    [0 1 0],'visible','on')
handles.imagen=imagen;
guidata(hObject,handles);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
im_ingreso=handles.imagen;
im_BD=handles.imagen_r;
set(handles.text6,'String','Procesando','fontsize',22,'backgroundcolor',[0 0 0],'foregroundcolor',...
    [1 0 0],'visible','on')
pause(2)
[h_BD h_ingreso R c]=compa_veri2(im_BD,im_ingreso);
x=1:256;

```

```

set(handles.text6,'String',' ','fontsize',18, 'backgroundcolor',[0.941 0.941 0.941], 'foregroundcolor',...
    [0.941 0.941 0.941], 'visible','on')
set(handles.text3,'String',R,'fontsize',18, 'backgroundcolor',[0 0 0], 'foregroundcolor',...
    c, 'visible','on')
plot(x,h_BD,'g','parent',handles.axes3)
xlabel(handles.axes3,'Niveles de Gris')
ylabel(handles.axes3,'Número de Pixeles')
title(handles.axes3,'Histogramas de BD')
plot(x,h_ingreso,'k','parent',handles.axes4)
xlabel(handles.axes4,'Niveles de Gris')
ylabel(handles.axes4,'Número de Pixeles')
title(handles.axes4,'Histogramas de Ingreso')

```

D.3. Programa de identificación de usuario

```

function varargout = Identificar(varargin)
% IDENTIFICAR M-file for Identificar.fig
% IDENTIFICAR, by itself, creates a new IDENTIFICAR or raises the existing singleton*.
%
% H = IDENTIFICAR returns the handle to a new IDENTIFICAR or the handle
% to the existing singleton*.
%
% IDENTIFICAR('CALLBACK',hObject,eventData,handles,...) calls the local
% function named CALLBACK in IDENTIFICAR.M with the given input arguments.
%
% IDENTIFICAR('Property','Value',...) creates a new IDENTIFICAR or raises
% the existing singleton*. Starting from the left, property value pairs are applied to the GUI
% before Identificar_OpeningFcn gets called. An unrecognized property name or invalid
% value makes property application stop. All inputs are passed to Identificar_OpeningFcn
% via varargin.
%
% *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one instance to
% run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES
% Edit the above text to modify the response to help Identificar
% Last Modified by GUIDE v2.5 19-Jun-2010 18:11:35

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name', mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @Identificar_OpeningFcn, ...
    'gui_OutputFcn', @Identificar_OutputFcn, ...

```

```

    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout1:nargout] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code – DO NOT EDIT

% — — — Executes just before Identificar is made visible.
function Identificar_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved – to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% varargin    command line arguments to Identificar (see VARARGIN)
% Choose default command line output for Identificar
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% — — — Outputs from this function are returned to the command line.
function varargout = Identificar_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)
% Get default command line output from handles structure
varargout{1} = handles.output;

% — — — Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
set(handles.text3,'String',' ','fontsize',18,'backgroundcolor',[0.941 0.941 0.941],'foregroundcolor',...
    [0.941 0.941 0.941],'visible','on')
set(handles.text2,'String','Procesando','fontsize',18,'backgroundcolor', [0 0 0],'foregroundcolor',...
    [1 0 0],'visible','on')
pause(2)
[caosx1 caosx2 caosx3]=senal_caos2(0.3,0,0.05);
caos=caosx1';
for i=1:5

```

```

    im_encr=load(strcat('E:\huellas encriptadas\BD\usuario_BD',int2str(i),'.txt'));
    im_desencr=((im_encr-caos)*1000);
    imagen=conv_dat_to_im2(im_desencr);
    im_BD(:,i)=imagen;
end
set(handles.text2,'String','Continuar','fontsize',18,'backgroundcolor',[0 0 0],'foregroundcolor',...
    [0 1 0],'visible','on')
handles.im_BD=im_BD;
guidata(hObject,handles);

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
set(handles.axes1,'visible','off')
set(handles.axes3,'visible','off')
set(handles.axes4,'visible','off')
cla(handles.axes1)
cla(handles.axes3)
cla(handles.axes4)
title(handles.axes1,' ')
set(handles.text3,'String',' ','fontsize',18,'backgroundcolor',[0.941 0.941 0.941],'foregroundcolor',...
    [0.941 0.941 0.941],'visible','on')
[nom,path]=uigetfile('*.bmp*','Abrir','E:\muestras\musuarios5\');
if isequal(nom,0)
    return
end
set(handles.text2,'String','Procesando','fontsize',18,'backgroundcolor',[0 0 0],'foregroundcolor',...
    [1 0 0],'visible','on')
pause(2)
imagen=imread([path nom]);
imshow(imagen,'parent',handles.axes2)
title(handles.axes2,'Imagen de Ingreso')
set(handles.text2,'String','Continuar','fontsize',18,'backgroundcolor',[0 0 0],'foregroundcolor',...
    [0 1 0],'visible','on')
handles.imagen=imagen;
guidata(hObject,handles);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
im_ingreso=handles.imagen;
im_BD=handles.im_BD;
x=1:256;
set(handles.text2,'String','Procesando','fontsize',18,'backgroundcolor',[0 0 0],'foregroundcolor',...
    [1 0 0],'visible','on')
for i=1:5
    im=im_BD(:,i);
    [h_BD h_ingreso R c]=compa_veri2(im,im_ingreso);

```

```

S=strcmp(R,'Aceptado')
if S==1
    imshow(im_BD(:,:,i),'parent',handles.axes1);
    title(handles.axes1,'Imagen de BD')
    set(handles.text2,'String','Identificado','fontsize',18,'backgroundcolor',[0 0 0],'foregroundcolor',...
        [0 1 0],'visible','on')
    set(handles.text2,'String','Continuar','fontsize',18,'backgroundcolor',[0 0 0],'foregroundcolor',...
        [0 1 0],'visible','on')
    plot(x,h_BD,'g','parent',handles.axes3)
    xlabel(handles.axes3,'Niveles de Gris')
    ylabel(handles.axes3,'Número de Pixeles')
    title(handles.axes3,'Histogramas de BD')
    plot(x,h_ingreso,'k','parent',handles.axes4)
    xlabel(handles.axes4,'Niveles de Gris')
    ylabel(handles.axes4,'Número de Pixeles')
    title(handles.axes4,'Histogramas de Ingreso')
    return
else
    set(handles.text2,'String','No identificado','fontsize',18,'backgroundcolor',[0 0 0],'foregroundcolor',...
        [0 1 0],'visible','on')
    end
end
set(handles.text2,'String','Continuar','fontsize',18,'backgroundcolor',[0 0 0],'foregroundcolor',...
    [0 1 0],'visible','on')

```


Anexo E

Programas complementarios

E.1. Conversión de imagen a vector de datos

Código del programa que convierte la imagen a un vector de datos que contiene los bits de la imagen.

```
function [Y3]=Convertirtobinario2(imagen)
%% %% Variables
A=imagen;
[vector_x vector_y]=size(A);
%% %% Etapa para crear vector de Bytes
A1=A(:,1);
for i=2:vector_y
    A1=[A1;A(:,i)]; % Asignacion de cada valor de pixel de la imagen a A1
end
%% %% Etapa para crear vector de bits
% Los valores son convertidos de decimal a binario
Y=dec2bin(A1); % Conversión de decimal a binario del arreglo A1,
% Proceso para reacomodar los bits
p=10;
dato=Y(1,:)' ;
dato1=Y(2,:)' ;
dato2=Y(3,:)' ;
dato3=Y(4,:)' ;
dato4=Y(5,:)' ;
dato5=Y(6,:)' ;
dato6=Y(7,:)' ;
dato7=Y(8,:)' ;
dato8=Y(9,:)' ;
dato9=Y(10,:)' ;
Y1=[dato;dato1;dato2;dato3;dato4;dato5;dato6;dato7;dato8;dato9];
rr=((vector_y*vector_x)/10)-1;
for pp=1:rr
```

```

dato=Y(p+1,:);
dato1=Y(p+2,:);
dato2=Y(p+3,:);
dato3=Y(p+4,:);
dato4=Y(p+5,:);
dato5=Y(p+6,:);
dato6=Y(p+7,:);
dato7=Y(p+8,:);
dato8=Y(p+9,:);
dato9=Y(p+10,:);
Y1=[Y1;dato;dato1;dato2;dato3;dato4;dato5;dato6;dato7;dato8;dato9];
p=p+10;
end
% Crear un vector Y3 de tipo double que contega los datos
Y3=str2num(Y1);

```

E.2. Sistema caótico

Código del programa que genera los vectores con los estados caóticos del mapeo de Rössler.

```

function [x2a x2b x2c] =senal_caos2(x0,y0,z0)
%% %% Declaración de variables
% variables del sistema caótico
alfa=3.8;
beta=0.05;
gamma=0.35;
delta=3.78;
dseta=0.2;
eta=0.1;
theta=1.9;
% Variables requeridas en el programa
x1=x0;
x2=y0;
x3=z0;
%% %% Parte de las operaciones matemáticas
for bits=1:8
    for vector_x=1:355
        for vector_y=1:390
            xa(vector_y)=(alfa*x0)*(1-x0)-(beta*(z0+gamma)*(1-2*y0));
            xb(vector_y)=(delta*y0)*(1-y0)+(dseta*z0);
            xc(vector_y)=eta*(((z0+gamma)*(1-(2*y0)))-1)*(1-theta*y0);
            x0=xa(vector_y);
            y0=xb(vector_y);
            z0=xc(vector_y);
        end
        x1=[x1 xa];
    end
end

```

```

        x2=[x2 xb];
        x3=[x3 xc];
    end
end
% selecciona los valores del vector del 1 – 1107600, el cual es, la dimensión del vector de datos de
la imagen.
x2a=x1(1:1107600);
x2b=x2(1:1107600);
x2c=x3(1:1107600);

```

E.3. Conversión del vector de datos a imagen

El objetivo del programa es reacomoda los bits del vector de datos en una matriz para regenerar la imagen de la huella dactilar.

```

function [I] =conv_dat_to_im2(Y3)
% Variables
vector_y=390;
vector_x=355;
% % ***** Acomodar el conjunto de 8 bits de 1 byte de la imagen *****
%***** Este ciclo es para obtener un arreglo de bytes de tipo DOUBLE *****
p=80;
dato01=Y3(1:8,1)';
dato02=Y3(9:16,1)';
dato03=Y3(17:24,1)';
dato04=Y3(25:32,1)';
dato05=Y3(33:40,1)';
dato06=Y3(41:48,1)';
dato07=Y3(49:56,1)';
dato08=Y3(57:64,1)';
dato09=Y3(65:72,1)';
dato10=Y3(73:80,1)';
Ynueva= [dato01;dato02;dato03;dato04;dato05;dato06;dato07;dato08;dato09;dato10];
rr=((vector_x*vector_y)/10)-1;
for pp=1:rr
    dato01=Y3(p+1:p+8,1)';
    dato02=Y3(p+9:p+16,1)';
    dato03=Y3(p+17:p+24,1)';
    dato04=Y3(p+25:p+32,1)';
    dato05=Y3(p+33:p+40,1)';
    dato06=Y3(p+41:p+48,1)';
    dato07=Y3(p+49:p+56,1)';
    dato08=Y3(p+57:p+64,1)';
    dato09=Y3(p+65:p+72,1)';
    dato10=Y3(p+73:p+80,1)';
    Ynueva= [Ynueva;dato01;dato02;dato03;dato04;dato05;dato06;dato07;dato08;dato09;dato10];
end

```

```

    p=p+80;
end
tic
%% ***** Reconstrucción de vector de bytes *****
Entero=uint8(Ynueva); % Conversión a entero sin signo de Ynueva
Cadena=int2str(Entero); % Conversión a cadena de Entero
Y1=bin2dec(Cadena); % Conversión a decimal de Cadena
t=toc; %% Reconstrucción de la matriz de imagen
h=Y1(1:vector_y,1);
renglon=vector_y-1;
cr=1;
for i=1:(vector_x-1)
    h=[h,Y1(cr:cr+renglon,1)];
    cr=vector_y+cr;
end
I = uint8(h);
%% Proceso de desplegado y guardado de la imagen
figure;imshow(I);
title('Imagen Recuperada')
imwrite(I,'E:\huellas_desencriptadas\imrecup.bmp');

```

E.4. Comparación de imágenes

```

function [h_BD h_ingreso R c]=compa_veri2(im_BD,im_ingreso)
%% Obtención de los histogramas de las imágenes
% inicializacion de las variables de histogramas
for i=1:256
    h_ingreso(i)=0;
    h_BD(i)=0;
end
% obtencion de la dimensión de la imagen
[r c] =size(im_BD);
[r1 c1] =size(im_ingreso);
% Proceso de obtención de los histogramas
for i=1:r
    for j=1:c
        % histograma de la imagen de base de datos
        k=im_BD(i,j);
        h_BD(k+1)=h_BD(k+1)+1;
        % histograma de la imagen de ingreso
        k1=im_ingreso(i,j);
        h_ingreso(k1+1)=h_ingreso(k1+1)+1;
    end
end
%% obtención de la diferencia de histogramas
for p=1:256

```

```

    if h_BD(p)>h_ingreso(p)
        dif(p)=h_BD(p)-h_ingreso(p);
    else
        dif(p)=h_ingreso(p)-h_BD(p);
    end
end
diferencia=sum(dif);
%% Comparación por medio de un umbral de 16000
if diferencia<16000
    R='Aceptado';
    c= [0 1 0];
else
    R='No Aceptado';
    c= [1 0 0];
end

```

Bibliografía

- [AA07] Javier Areitio Bertolín y María Teresa Areitio Bertolín. Análisis en torno a la tecnología biométrica para los sistemas electrónicos de identificación y autenticación. *Revista española de electrónica*, ISSN 0482-6396, 630:52–67. 2007. http://www.redeweb.com/_txt/630/52.pdf.
- [AB08] Javier Areitio Bertolín. *Seguridad de la Información: redes, informática y sistemas de información*. Paraninfo, CENGAGE Learning. 2008.
- [AC09] Aguilar Bustos A.Y., Cruz Hernández C. *Synchronization of discrete-time hyperchaotic systems: an application in communications*, *Chaos, Solitons & Fractals* 41:1301–1310. 2009
- [AC+10] Aguilar Bustos A.Y., Cruz Hernández C., López Gutiérrez R.M., Tlelo Cuatle E., Posadas Castillo C. *Hyperchaotic Encryption for Secure E-Mail Communication*, en: Emergent Web Intelligence, Ed. Springer AI & KP 2010.
- [ACLP08] Aguilar Bustos A. Y., Cruz Hernández C., López Gutiérrez R. M. y Posadas Castillo C. Synchronization of Different Hyperchaotic Maps for Encryption. *Nonlinear Dynamics and Systems Theory*, 8(3):221–236, 2008.
- [ALB09] Coto Alberto. *Entrenamiento Mental*. Capítulo Algo más que números. 95–96. EDAF México. 2009.
- [ANDO09] A. Balanov, N. Janson, D. Postnov, O. Sosnovtseva. *Synchronization: from simple to complex*, Springer-Verlag, Berlin-Heidelberg, 2009. UABC.
- [ANI08a] K. Jain Anil. Biometric Recognition: A new Paradigm for Security. *Website*, 2008. http://biometrics.cse.msu.edu/Presentations/AnilJain_BiometricsParadigm_Illinois_Oct07.pdf.
- [ANI08b] K. Jain Anil. Biometric System Security. *Website*, 2008. http://biometrics.cse.msu.edu/Presentations/AnilJain_BiometricsSecurity_Japan_Jan06.pdf.
- [BOL04] Ruud Bolle, Jonathan Connell y Sharathchandra Pankanti. *Guide to biometrics*. Springer professional computing. Springer-Verlag New York. 2004.
- [BRA92] Nicholas B. Tufillaro, Jeremiah Reilly y Tyler Abbott Addison-Wesley. *An experimental approach to nonlinear dynamics and chaos*, 1992.
- [CMC04] Chen G., Mao Y.B., Chui C.K. *A symmetric image encryption scheme based on 3D chaotic cat maps*, *Chaos, Solitons & Fractals*. 21:749–761. 2004

- [CM10] Cruz Hernández C., Martynyuk A.A. *Advances in chaotic dynamics with applications*. Series on Stability, Oscillations and Optimization of Systems, Cambridge Scientific Publishers, London, Vol. 4, 2010.
- [CRU04] Cruz Hernández C. Synchronization of time-delay Chua's oscillator with application to secure communication, *Nonlinear Dyn Syst Theory*, 4:1–13, 2004.
- [CRU08] Cruz Hernández C., Romero-Haros N. *Communicating via synchronized time-delay Chua's circuits*. *Commun Nonlinear Sci Numer Simul*, 13:645–659, 2008.
- [CS03] Julien Clinton Sprott, *Chaos and time-series analysis*. Oxford University Press, New York, 2003.
- [CS05] Cruz Hernández C., Serrano Guerrero H. *Cryptosystems based on synchronized Chua's circuits*. En: *Proceedings of the 16th IFAC World Congress*, Prague, Czech Republic, 2005.
- [DC05] López Mancilla D., Cruz Hernández C. *Output synchronization of chaotic systems: model-matching approach with application to secure communication*. *Nonlinear Dyn Syst Theory*. 5:141-156. 2005
- [DGV93] Joan Daemen, René Govaerts y Joos Vandewalle. Weak Keys for IDEA. *Advances in Cryptology, CRYPTO 93 Proceedings*, Lecture Notes in Computer Science 773, D. R. Stinson (ed.), Springer-Verlag, Santa Barbara, California, USA. 224–231, 1993. <https://www.cosic.esat.kuleuven.be/publications/article-140.pdf>.
- [DKT07] Jean Pierre David, Kassem Kalach y Nicolas Tittley. Hardware Complexity of Modular Multiplication and Exponentiation. *IEEE Transactions on Computers*, 56(10):1308–1319. 2007.
- [DR01a] Joan Daemen y Vincent Rijmen. Advanced Encryption Standard (AES). *Federal Information Processing Standards Publication 197*, National Institute of Standards and Technology. 26 de Noviembre de 2001. <http://www.csrc.nist.gov/publications/fips/fips197/fips-197.pdf>.
- [DR01b] Joan Daemen y Vincent Rijmen. The Wide Trail Design Strategy. Bahram Honary, *Cryptography and Coding*, volumen 2260 de *Lecture Notes in Computer Science*, páginas 222–238. Springer Berlin/Heidelberg, 2001. http://dx.doi.org/10.1007/3-540-45325-3_20.
- [DZ04] David D. Zhang. *Palmprint authentication*. kluwer Academic Publishers, Norwell Massachusetts USA. 2004.
- [EE04] Adriana Elisa Espinosa. *El caos y la caracterización de series de tiempo a través de técnicas de la dinámica No-Lineal*. Tesis. Universidad Nacional Autónoma de México. 2004.
- [ELG85] Taher El Gamal . A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472. 1985.
- [EMPU08] Dominik Engel, Student Member, Elias Pschernig y Andreas Uhl. An Analysis of Light-weight Encryption Schemes for Fingerprint Images. *IEEE Transactions on Information Forensics and Security*, 2(3):173–182, 2008.

- [EO92] E. Ott. *Chaos in dynamical systems*. Cambridge University Press, Cambridge, 1992.
- [ESP92] Real Academia Española. *Diccionario de la lengua española/Real Academia Española*. Talleres Graficos de la Editorial Espasa-cape S.A. 1992.
- [FC92] F.C. Moon. *Chaotic vibration. An introduction for applied scientists and engineers*. Wiley, New York, 1992.
- [FD06] Jesús Ignacio Fernández Domingo. *La firma electrónica: (aspectos de la Ley 59/2003, de 19 de diciembre)*. Capítulo Aspectos Técnicos de la firma electrónica. 81–84. Editorial Reus. 2006.
- [FIPSP09] Federal Information Processing Standards Publication. National Institute of Standards and Technology, FIPS PUB 186-3: Digital Signature Standard (DSS). *Website*. Junio 2009. http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf.
- [FIPSP99] Federal Information Processing Standards Publication. National Institute of Standards and Technology, FIPS PUB 46-3: Data Encryption Standard (DES). *Website*. Octubre 25, 1999. <http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>
- [FLSCC06] Li Fang, Maylor K.H. Leung , Tejas Shikhare, Victor Chan y Kean Fatt Choon. Palmprint classification. *IEEE International Conference on Systems, Man and Cybernetics, 2006. SMC '06*. 4:2965–2969. 2006. <http://www3.ntu.edu.sg/SCE/labs/forse/PDF/palmprintClass.pdf>. Consultado el 20 marzo del 2010.
- [FO06] Marcos Faúndez Zanuy y Sergio Osuna Silvestre. Experimentos sobre la vulnerabilidad de sistemas biométricos. *Terceras jornadas nacionales de la ABIE. Sevilla*, Noviembre 2006. Páginas 13–22. ISBN: 84-934865-1-5. http://www.shsconsultores.es/jornadas/0203_-_experimentos_sobre_la_vulnerabilidad_de_sistemas_biometricos.pdf.
- [FS07] Armando Fernández Sarasola. *Control de los Sistemas Continuos*. Textos universitarios ediuno. Oviedo Asturias, 2007.
- [GAKD00] Shlomo Greenberg, Mayer Aladjem, Daniel Kogan y Itshak Dimitrov. Fingerprint Image Enhancement using Filtering Techniques. *15th International Conference on Pattern Recognition*. 3:322–325. 2000.
- [GCLG08] Gámez L., Cruz Hernández C., López Gutiérrez R.M., García Guerrero E.E. *Synchronization of multi-scroll chaos generators: application to private communication*, Revista Mexicana de Física 54:299–305. 2008.
- [GCLG09] Gámez L., Cruz Hernández C., López Gutiérrez R.M., García Guerrero E.E. *Synchronization of Chua's Circuits with Multi-scroll Attractors: Application to Communications*, Commun Nonlinear Sci Numer Simul, 14:2765-2775. 2009.
- [GRGM08] Krishnamurthy G.N, Dr. V. Ramaswamy, Leela G.H y Ashalatha M.E. Blow-CAST-Fish: A New 64-bit Block Cipher. *IJCSNS International Journal of Computer Science and Network Security*, 8(4):282–290, 2008. http://paper.ijcsns.org/07_book/200804/20080440.pdf.
- [HG88] H.G. Schuster. *Deterministic chaos. An Introduction*. VCH, Weinheim, 1988.

- [HIL00] Robert C. Hilborn. *Chaos and Nonlinear Dynamics: An Introduction for Scientists and Engineers*. Oxford University Press Inc., New York. 2000.
- [HKKP06] Hyun C.H., Kim J.H., Kim E., Park M. *Adaptive fuzzy observer based synchronization design and secure communications of chaotic systems*. *Chaos, Solitons & Fractals*. 27:930-940. 2006.
- [HOF07] Nick Hoffman. A Simplified IDEA Algorithm. *Cryptologia* 31(2):143–151, 2007. <http://www.informaworld.com/10.1080/01611190701215640>.
- [HWJ98] Lin Hong, Yifei Wan y Anil Jain. Fingerprint image enhancement: Algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):777–789, 1998.
- [HW03] Jin-Hua Hong y Cheng-Wen Wu. Cellular-array modular multiplier for fast RSA public-key cryptosystem based on modified Booth’s algorithm. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 11(3):474–484, 2003.
- [JA01] Elizabeth J. Jewell, Frank Abate. *The New Oxford American Dictionary*. Oxford University Press. USA New York. 2001.
- [JBP99] Anil Jain, Ruud bolle y Sharath Pankanti. *Biometrics Personal identification in Networked Society*. kluwer Academic Publishers. Norwell Massachusetts USA, 1999.
- [JFR03] Anil K. Jain, Patrick Joseph Flynn y Arun Abraham Ross. *Handbook of biometrics*. Springer Science + Business Media, LLC. New York, USA. 2008.
- [JP93] J. Guckenheimer, P. Holmes. *Nonlinear oscillations, dynamical systems and bifurcations of vector field*. Springer, New York, 1993.
- [JR04] Anil K. Jain y Arun Ross. Multibiometric systems. *Communications of the ACM*, 47(1):34–40, 2004.
- [JRP04] Anil K. Jain, Arun Ross y Salil Prabhakar. An introduction to biometric recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 14(1):4–20, 2004.
- [KRS00] Kennedy M.P., Rovatti R., Setti G. *Chaotic electronics in telecommunications*, CRC Press 131-149. 2000
- [LC05] López Mancilla D., Cruz Hernández C. *A note on chaos-based communication schemes*. *Revista Mexicana de Física* 51:265-269. 2005
- [LG00] Maria de las Nieves Lorenzo Gonzales. *Influencia del Ruido Gaussiano Correlacionado en la Sincronizacion de Sistemas Caóticos*. Tesis. Universidad de Santiago de Compostela. 2000.
- [LLT06] Larson E.L., Liu J.M, Tsimring L.S. *Digital communications using chaos and nonlinear dynamics*. Springer 2006.
- [LOCV01] Claudia Lerma G., Daniel Olivares C., César Contreras Faz y Marco V. José. Sistema de Análisis No Lineal de la Variabilidad de la Frecuencia Cardíaca. *Instituto Tecnológico de Chihuahua, ELECTRO 2001*. 2001. http://www.cnd.mcgill.ca/pdf_files/ELECTRO2001.pdf.

- [LPLC09] López Gutiérrez R.M., Posadas Castillo C., López Mancilla D., Cruz Hernández C. *Communicating via robust synchronization of chaotic lasers*. Chaos, Solitons & Fractals. 42:277-285. 2009.
- [MA06] Villacorta Michelena Alberto. *Enredados. El mundo de la Internet*. Capítulo hackers y criptografía, 164–165. 2006.
- [MC07] Mejía Camacho Juan Manuel. *Encriptamiento de Información Mediante Caos*. Tesis. Universidad Autónoma de Baja California. 2007.
- [MCPMC06] Daesung Moon, Yongwha Chung, Sung Bum Pan, Kiyoun Moon y Kyo II Chung. An Efficient Selective Encryption of Fingerprint Images for Embedded Processors. *ETRI Journal*, 28(4):444–452, 2006.
- [MEN98] Mensour B., Longtin A. *Synchronization of delay-differential equations with application to private communication*. Phys Lett A, 244:59-70. 1998.
- [MIC04] Microsoft. *Diccionario de Informática e Internet*. McGraw-Hill/Interamericana de España, 2004.
- [MJ04] Davide Maltoni y Anil K. Jain. *Biometric authentication*. Springer-Verlag Berlin Heidelberg. 2004.
- [MMJP03] Davide Maltoni, Dario Maio, Anil K. Jain y Salil Prabhakar. *Handbook of Fingerprint Recognition*. Springer-Verlag New York, Inc. 2003.
- [MOV00] Alfred J. Menezes, Paul C. van Oorschot y Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, USA. 2000.
- [MPRS09] Florian Mendel, Thomas Peyrin, Christian Rechberger y Martin Schl  ffer. Improved Cryptanalysis of the Reduced Gr  stl Compression Function, ECHO Permutation and AES Block Cipher. Michael Jacobson, Vincent Rijmen y Reihaneh Safavi-Naini. *Selected Areas in Cryptography*, volumen 5867 de *Lecture Notes in Computer Science*, 16–35. Springer Berlin/Heidelberg, 2009. http://dx.doi.org/10.1007/978-3-642-05445-7_2.
- [MSR04] Morris Hirsch, Stphen Smale, Robert L. Devaney. *Differential equations, Dynamical Systems An introduction to chaos*. Academic Press, 2004.
- [ORM10] Lic. Jorge Ormachea. Seguridad en los sistemas biometricos. *Website*, Martes 01-Junio-2010. http://rosalesuriona.com/spip.php?article114&debut_articles=260.
- [PL01] Palaniyandi P., Lakshmanan M. Secure digital transmission by multistep parameter modulation and alternative driving of transmitter variables, *Int J Bifurct Chaos*, 11:2031-2036. 2001.
- [PLC08] Posadas Castillo C., López Gutiérrez R.M., Cruz Hernández C. *Synchronization of chaotic solid state Nd:YAG lasers: application to secure communication*. Commun Nonlinear Sci Numer Simul 13:1655-1667. 2009.
- [SA99] Jorge Sánchez Arriazu. Descripción del Algoritmo DES. *Website*. Diciembre 1999. <http://www.tierradelazaro.com/public/libros/des.pdf>

- [SC05] Michael G. Solomon y Mike Chapple. *Information Security Illuminated*. Jones and Bartlett Learning, 2005.
- [SCB09] CIB Sistemas Controles Biometricos. ¿Que es el viridi 300?. *Website*, Viernes 25-Sep-2009. <http://www.cib-ca.com/principal.htm>.
- [SCH10] Schneier.com. The Blowfish Encryption Algorithm. *Website*, 2010. <http://www.schneier.com/blowfish.html>
- [SCH93] Bruce Schneier. Description of a New Variable-Length Key, 64-bit Block Cipher (Blowfish). In *Fast Software Encryption, Cambridge Security Workshop Proceedings*, Diciembre 1993.
- [SEC09] SecureRus. Viridi Biometric Fingerprint Recognition System, Viridi 300N/S. *Website*, Lunes 28-Sep-09. <http://securerus.com/Pdflibrary/Accesscontrolsystems/SecurerusViridiBiometric.pdf>.
- [SH94] S.H. Strogatz. *Nonlinear dynamics and chaos: With application to physics, biology, chemistry and engineering*, Addison-Wesley, Reading, 1994.
- [SM+01] Amparo Fúster Sabater, Dolores de la Guía Martínez, Luis Hernández Encinas, Fausto Montoya Vitini y Jaime Muñoz Masqué. *Técnicas Criptográficas de Protección de Datos*. Alfaomega Ra-Ma. 2 edición. 2001.
- [SR00] Raúl Sánchez Refillo. *Mecanismos de Autenticación Biométrica Mediante Tarjeta Inteligente*. Tesis. Universidad Politécnica de Madrid, Escuela Técnica Superior de Ingenieros de Telecomunicación. 2000.
- [ST09a] Tecnologia Aplicada al Control Sistemas Tecnologicos. Cerraduras Biométricas STLK1000. *Website*, Martes 29-Sep-09. <http://www.sistecbio.com.ar/productos/stlk1000.asp>.
- [ST09b] Tecnologia Aplicada al Control Sistemas Tecnologicos. FingerPrint STCA02. *Website*, Martes 29-Sep-09. <http://www.sistecbio.com.ar/productos/stca02.asp>
- [ST09c] Tecnologia Aplicada al Control Sistemas Tecnologicos. Lector Biométricas STUU5000. *Website*, Martes 29-Sep-09. <http://www.sistecbio.com.ar/productos/stuu5000.asp>.
- [SW10] Xiaoling Sun y Meiqin Wang. Differential Cryptanalysis of Blow-CAST-Fish. Khaled Elleithy, Tarek Sobh, Magued Iskander, Vikram Kapila, Mohammad A. Karim y Ausif Mahmood. *Technological Developments in Networking, Education and Automation*, páginas 523–528. Springer Netherlands, 2010. http://dx.doi.org/10.1007/978-90-481-9151-2_91
- [TAN03] Andrew S. Tanenbaum. *Redes de computadoras*. Capítulo Seguridad en Redes. Pearson Prentice Hall, Pearson Educación, México, 2003.
- [TGC06] Jesús J. Ortega Triguero, Miguel Ángel López Guerrero y Eugenio C. García del Castillo Crespo. *Introducción a la criptografía: historia y actualidad*. Capítulo Los primeros pasos, Cifrado por sustitución. 19–35. Colección Monografías, Universidad de Castilla-La Mancha, España, 2006.
- [TL01] Tang S., Liu J.M. *Message encoding-decoding at 2.5 Gbits/s through synchronization of chaotic pulsing semiconductor lasers*. Opt Lett, 26:1843-1845. 2001.

- [UPJP04] Umut Uludag, Sharath Pankanti, Anil K. Jain y Salil Prabhakar. Biometric Cryptosystems: Issues and Challenges. *Proceedings of the IEEE*. 92(6):948–960. 2004.
- [VS95] V.S. Anishchenko. *Dynamical chaos-models and experiments: Appearance routes and structure of chaos in simple dynamical systems*. World Scientific, Singapore, 1995.
- [VTDL03] E. Verbitskiy, P. Tuyls, D. Denteneer y J.P. Linnartz. Reliable Biometric Authentication with Privacy Protection, *24th Benelux Symposium on Information Theory*, 125–132. 2003.
- [WM] Boukhari Wassaila y Benyettou Mohamed. Personal Palmprints Based-Identification Using Liquid State Machine. *Laboratoire de Modélisation et d’Optimisation des Systèmes Industriels-LAMOSI, University of Science and Technologies-Mohamed Boudiaf- Oran - Algeria*. Website. <http://eref.uqu.edu.sa/files/eref2/folder6/f88.pdf>. Consultado el 20 marzo del 2010.
- [WON02] Wong K.W. A fast chaotic cryptographic scheme with dynamic look-up table, *Phys Lett A*, 298:238-242. 2002.
- [YWC97] Yang T., Wu C.W., Chua L.O. Cryptography based on chaotic systems, *IEEE Trans. Circ. Syst. I*. 44:469-472. 1997.