

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS  
SUPERIORES DE MONTERREY  
CAMPUS MONTERREY

DIVISIÓN DE INGENIERÍA Y ARQUITECTURA  
PROGRAMA DE GRADUADOS EN INGENIERÍA



APLICACIÓN DE MÉTODOS ESTADÍSTICOS  
PARA EL RECONOCIMIENTO DE IRIS

TESIS

PRESENTADA COMO REQUISITO PARCIAL  
PARA OBTENER EL GRADO ACADÉMICO DE:

MAESTRO EN CIENCIAS CON  
ESPECIALIDAD EN AUTOMATIZACIÓN

POR:

SAÚL MONTES DE OCA ARMEAGA

MONTERREY, N.L.

DICIEMBRE DE 2005

INSTITUTO TECNOLÓGICO Y DE ESTUDIOS  
SUPERIORES DE MONTERREY  
CAMPUS MONTERREY

DIVISIÓN DE INGENIERÍA Y ARQUITECTURA  
PROGRAMA DE GRADUADOS EN INGENIERÍA

Los miembros del Comité de Tesis recomendamos que el presente proyecto de tesis presentado por el Ing. Saúl Montes de Oca Armeaga sea aceptado como requisito parcial para obtener el grado académico de:

**Maestro en Ciencias con  
Especialidad en Automatización**

Comité de Tesis:

---

Dr. Luis Eduardo Garza Castañón  
Asesor

---

Dr. Rubén Morales Menéndez  
Sinodal

---

Dr. Juan Arturo Nolasco Flores  
Sinodal

Aprobado:

---

Dr. Federico Viramontes Brown  
Dr. del Programa de Graduados en Ingeniería

Diciembre de 2005

# Dedicatoria

A MI MADRE, ANA MARÍA ARMEAGA FRÍAS, POR SU AMOR Y APOYO INCONDICIONAL A LO LARGO DE MI VIDA, POR COMPARTIR MIS SUEÑOS Y ALIVIAR MIS TRISTEZAS. HAS SIDO MI MEJOR EJEMPLO DE ENTREGA, AMOR Y HUMILDAD.

A MI PADRE, LUIS MONTES DE OCA BECERRIL, UN GRAN HOMBRE QUE HA VELADO POR MÍ, ME HA OFRECIDO SU APOYO, COBIJO, AMOR Y SABIDURÍA. HAS SIDO MI MEJOR EJEMPLO DE DEDICACIÓN, FUERZA DE VOLUNTAD, VISIÓN Y TRABAJO.

A MI HERMANA RUTH MONTES DE OCA ARMEAGA, LA NIÑA MÁS TIERNA QUE CONOZCO QUE ME HA BRINDADO TANTO CARIÑO Y APOYO EN MOMENTOS DIFÍCILES.

A MI HERMANO ESAU MONTES DE OCA ORTÍZ, QUIEN ME HA MOSTRADO LO BELLO DE LA VIDA Y LO GRANDIOSA QUE PUEDE SER CUANDO SE VIVE AL MÁXIMO.

DIOS LOS CONSERVE MUCHOS AÑOS UNIDOS Y A MI LADO.

CON MUCHO CARIÑO,  
SAÚL

# Agradecimientos

A mis padres Ana Maria Armeaga Frías y Luis Montes de Oca Becerril, por todo su amor, apoyo incondicional al emprender este sueño y sobre todo por creer en mi, mil gracias.

A mis hermanos Ruth y Esau Montes de Oca, por ser la fuente de mi fortaleza a pesar de la distancia.

A mis hermanos Josué e Israel Montes de Oca, por preocuparse de mí cuando estuve lejos de casa.

Al Dr. Luis Eduardo Garza Castañón, por la asesoría y confianza que me brindó durante el desarrollo del presente trabajo.

Al Dr. Rubén Morales Menéndez, por su valiosa asesoría, consejos y comentarios en la presente investigación.

A mis amigos Miriam Betsabé González Conzuelo, Gilberto Reynoso Meza, Sheyla Yael Aguilar Martínez y Graciela Sepúlveda Castro, por estar ahí en los momentos de flaqueza, por ser la fuente de alegría en mis tristezas, por su compañía y apoyo.

A mis amigos y compañeros de la maestría, por su paciencia y compartir conmigo sus experiencias.

A mis amigos que están lejos, que a pesar de la distancia compartieron este sueño brindándome apoyo y consejos.

Y principalmente a Dios por brindarme la oportunidad de culminar este sueño.

# Resumen

En este trabajo se propone la aplicación de métodos estadísticos para el reconocimiento automático de personas utilizando el iris como biométrica. Con esto se busca abordar el problema de variabilidad presente en la adquisición de las imágenes. Esto se refleja en defectos en imágenes tales como la iluminación deficiente o en exceso, los reflejos, el mal enfoque y las obstrucciones.

En la etapa de localización del iris y normalización se utilizó el algoritmo desarrollado por [18]. La etapa de localización se basa en la utilización de operadores integro-diferenciales y la normalización en la transformación de coordenadas rectangulares a polares. Ambas emplean la técnica de ecualización por histograma para localizar con éxito muestras con ojos muy oscuros, muy claros, con obstrucción o con defectos provenientes de la etapa de adquisición de la imagen.

En la etapa de extracción de características se hacen las aportaciones principales de la tesis, y se exponen dos líneas de investigación: la primera línea consiste en la utilización de técnicas de muestreo aleatorio y Hammersley combinadas con histogramas acumulados. En esta misma línea se incluyeron técnicas de optimización de parámetros utilizando recocido simulado y muestreo factorial. La segunda línea de investigación consiste en la modelación del iris a través de un modelo estocástico ARX. En la última, de comparación y decisión, no se tuvieron aportaciones originales ya que se utilizaron variaciones de la distancia Euclideana.

Los métodos propuestos fueron probados en modo de identificación y modo de verificación con dos bases de datos: Miles Research y UBIRIS. Los resultados obtenidos en la línea de muestreo fueron muy buenos, especialmente en el modo de identificación, donde se logró contrarrestar los efectos del ruido que se encontraba presente en las bases de datos. Para el método estocástico ARX, los resultados fueron aceptables y con muchas áreas de oportunidad para ser mejorado.

# Índice general

---

<b>1. Introducción</b>	<b>1</b>
1.1. Antecedentes . . . . .	2
1.2. Planteamiento del problema . . . . .	4
1.3. Objetivos de la investigación . . . . .	4
1.4. Justificación . . . . .	5
1.5. Contribuciones . . . . .	5
1.6. Alcances y limitaciones . . . . .	5
1.7. Trabajos relacionados . . . . .	6
1.8. Descripción de la metodología del trabajo . . . . .	7
1.9. Contenido . . . . .	7
<b>2. Introducción a los métodos de reconocimiento biométricos</b>	<b>9</b>
2.1. Requerimientos de un factor biométrico . . . . .	10
2.2. Sistemas de reconocimiento biométrico . . . . .	11
2.3. Errores de un Sistema Biométrico . . . . .	12
2.4. Factores biométricos más usados . . . . .	14
2.5. Aplicaciones de los sistemas biométricos . . . . .	19
2.6. Ventajas y desventajas de los sistemas biométricos . . . . .	20
2.6.1. Reconocimiento positivo en las aplicaciones comerciales . . . . .	20
2.6.2. Reconocimiento negativo en aplicaciones de gobierno y forenses . . . . .	21
2.6.3. Limitaciones de los sistemas biométricos unimodales . . . . .	21
2.7. Falsificación de un factor biométrico . . . . .	22
<b>3. El iris como factor biométrico</b>	<b>25</b>
3.1. Técnicas publicadas . . . . .	26
3.2. Algoritmo de Daugman . . . . .	28

3.2.1. Búsqueda del iris en la imagen . . . . .	28
3.2.2. Decodificación del Iris por Demodulación de Wavelet 2-D . . . . .	29
3.2.3. Reconocimiento de iris sin importar tamaño, posición y orientación . . . . .	32
3.3. Algoritmo de Avila . . . . .	32
3.3.1. Adquisición de la imagen y preprocesamiento . . . . .	32
3.3.2. Extracción de características . . . . .	33
3.3.3. Clasificación y verificación . . . . .	34
3.4. Algoritmo de Tisse . . . . .	34
3.4.1. Localización del iris . . . . .	34
3.4.2. Transformación de coordenadas rectangulares a polares . . . . .	35
3.4.3. Extracción de características . . . . .	35
3.4.4. Comparación . . . . .	36
3.5. Algoritmo de Li Ma . . . . .	36
3.5.1. Preprocesamiento de la imagen . . . . .	36
3.5.2. Extracción de características . . . . .	36
3.5.3. Diseño del clasificador . . . . .	38
<b>4. Descripción de los métodos propuestos</b>	<b>39</b>
4.1. Localización del iris . . . . .	40
4.1.1. Ecualización del histograma para mejorar la localización del iris . . . . .	41
4.2. Normalización . . . . .	42
4.3. Extracción de características . . . . .	43
4.3.1. Extracción de características por el método de Muestreo . . . . .	43
4.3.2. Extracción de características por el método ARX . . . . .	44
4.4. Comparación . . . . .	45
4.4.1. Comparación para el método de muestreo . . . . .	45
4.4.2. Comparación para el método ARX por medio de mínimos cuadrados . . . . .	46
4.5. Decisión . . . . .	47
4.5.1. Decisión basada en el modo identificación . . . . .	47
4.5.2. Decisión basada en el modo verificación . . . . .	48
4.6. Técnicas de optimización . . . . .	48
4.6.1. Muestreo aleatorio con reposición . . . . .	48
4.6.2. Muestreo aleatorio sin reposición . . . . .	49
4.6.3. Muestro Hammersley . . . . .	49

---

4.6.4. Comparación entre los métodos de muestreo . . . . .	50
4.6.5. Recocido Simulado . . . . .	51
4.6.6. Muestreo Factorial . . . . .	52
4.7. Resumen de los métodos propuestos . . . . .	53
4.7.1. Método ARX en modo identificación . . . . .	53
4.7.2. Método ARX en modo verificación . . . . .	53
4.7.3. Método de Muestreo en modo identificación . . . . .	54
4.7.4. Método de Muestreo en modo verificación . . . . .	55
<b>5. Resultados Experimentales y optimización</b>	<b>57</b>
5.1. Introducción . . . . .	57
5.2. Descripción de los experimentos . . . . .	58
5.2.1. Experimentos modo identificación . . . . .	58
5.2.2. Experimentos modo verificación . . . . .	58
5.3. Bases de datos . . . . .	59
5.3.1. Base de datos Miles Research . . . . .	59
5.3.2. Base de datos UBIRIS . . . . .	59
5.3.3. Evaluación de la base de datos. . . . .	60
5.3.4. Eliminación del ruido de las bases de datos . . . . .	62
5.4. Resultados de los experimentos en modo identificación . . . . .	63
5.4.1. Condiciones experimentales para los métodos propuestos . . . . .	63
5.4.2. Resultados experimentales en modo identificación . . . . .	65
5.4.3. Optimización del método de muestreo por recocido simulado . . . . .	67
5.4.4. Resultados del muestreo Hammersley optimizado por recocido simulado .	68
5.4.5. Condiciones del método de muestreo Hammersley factorial . . . . .	70
5.4.6. Resultados del método de muestreo Hammersley factorial . . . . .	70
5.5. Resultados experimentales en modo verificación . . . . .	71
5.5.1. Resumen de los resultados en modo verificación . . . . .	71
5.5.2. Resultados en la Base de datos Miles Research . . . . .	73
5.5.3. Resultados en la Base de datos UBIRIS . . . . .	74
<b>6. Conclusiones y trabajos a futuro</b>	<b>77</b>
6.1. Introducción . . . . .	77
6.2. Resumen de los resultados del trabajo de investigación . . . . .	77



6.3. Sugerencias de investigaciones futuras . . . . .	79
<b>A. Resultados de la base de datos Miles Research (Identificación)</b>	<b>85</b>
A.1. Método de muestreo aleatorio con reposición . . . . .	86
A.2. Método de muestreo aleatorio sin reposición . . . . .	87
A.3. Método de muestreo Hammerlsey . . . . .	88
A.4. Método de muestreo hammesley con parámetros optimizados . . . . .	89
A.5. Método ARX . . . . .	91
<b>B. Resultados de la base de datos UBIRIS (Identificación)</b>	<b>93</b>
B.1. Método de muestreo aleatorio con reposición . . . . .	95
B.2. Método de muestreo aleatorio sin reposición . . . . .	98
B.3. Método de muestreo Hammerlsey . . . . .	101
B.4. Método de muestreo hammesley con parámetros optimizados . . . . .	104
B.5. Método de muestreo Hammersley factorial . . . . .	108
B.6. Método ARX . . . . .	110
<b>C. Resultados de la base de datos Miles Research (Verificación)</b>	<b>111</b>
C.1. Método de muestreo aleatorio con reposición . . . . .	111
C.2. Método de muestreo aleatorio sin reposición . . . . .	113
C.3. Método de muestreo Hammerlsey . . . . .	114
C.4. Método ARX . . . . .	116
<b>D. Resultados de la base de datos UBIRIS (Verificación)</b>	<b>118</b>
D.1. Método de muestreo aleatorio con reposición . . . . .	118
D.2. Método de muestreo aleatorio sin reposición . . . . .	122
D.3. Método de muestreo Hammerlsey . . . . .	125
D.4. Método ARX . . . . .	129
<b>E. Código fuente de los programas implementados en MATLAB®</b>	<b>133</b>
E.1. Localización y normalización . . . . .	133
E.2. Extracción de características, comparación y decisión . . . . .	139
E.2.1. Método ARX . . . . .	139
E.2.2. Método de muestreo . . . . .	146
E.2.3. Optimización del muestreo Hammersley por recocido simulado . . . . .	163
E.2.4. Muestreo Hammerley Factorial . . . . .	168

# Índice de figuras

---

1.1. Estructura del iris vista de la parte frontal. . . . .	3
1.2. Esquema de la primera parte del sistema ya implementado. . . . .	6
1.3. Esquema del procedimiento de comparación de datos . . . . .	7
2.1. Diagrama de bloques de un sistema biométrico . . . . .	11
2.2. Modelo general de un sistema de identificador biométrico . . . . .	12
2.3. Proporción del error de un sistema biométrico. . . . .	13
2.4. Curva de Características Operativas del Receptor. . . . .	14
2.5. Ejemplos de factores biométricas . . . . .	15
2.6. Ejemplos de factores biométricas (continuación). . . . .	16
2.7. Ejemplos de factores biométricas (continuación). . . . .	17
2.8. Ejemplos de factores biométricas (continuación). . . . .	18
2.9. Ejemplos de aplicaciones biométricas . . . . .	20
2.10.Efectos del ruido en las imágenes de un sistema biométrico . . . . .	21
2.11 Efectos del ruido en las imágenes de un sistema biométrico . . . . .	22
2.12.Falsificación de huellas digitales . . . . .	22
3.1. Ubicación del iris en la anatomía del ojo. . . . .	25
3.2. Localización del iris por medio del algoritmo de Daugman. . . . .	29
3.3. Fase del proceso de demodulación . . . . .	30
3.4. Comparaciones por medio de la distancia de Hamming . . . . .	31
3.5. Técnica de la transformada wavelet dyadic . . . . .	34
3.6. Generación del código del iris. . . . .	36
3.7. Preprocesamiento de la imagen. . . . .	37
4.1. Representación de la búsqueda de los límites límbico y pupilar. . . . .	40

4.2. Localización del iris y la pupila. . . . .	40
4.3. Detección fallida del límite pupilar y límbico. . . . .	41
4.4. Detección del límite pupilar y límbico de los ojos de la figura 4.3. . . . .	42
4.5. Remapeo del área del análisis de coordenadas polares a rectangulares. . . . .	43
4.6. Ejemplo de la etapa de normalización. . . . .	43
4.7. Imagen que representan el área analizar de los iris. . . . .	44
4.8. Representación del modelo ARX de las cintas . . . . .	45
4.9. Ejemplo para la determinación del umbral. . . . .	47
4.10 Muestreo aleatorio simple . . . . .	50
4.11 Muestreo aleatorio sin reposición . . . . .	50
4.12 Muestreo Hammersley . . . . .	51
4.13 Algoritmo de Recocido Simulado para N iteraciones. . . . .	52
4.14 Ejemplo de los dos niveles de muestreo por el método factorial . . . . .	53
4.15 Descripción del modo identificación del método ARX. . . . .	54
4.16 Descripción del modo verificación del método ARX. . . . .	54
4.17 Descripción del modo verificación del método de muestreo. . . . .	55
4.18 Descripción del modo verificación del método de muestreo. . . . .	55
5.1. Ejemplos de localizaciones fallidas de la base de datos Miles Research. . . . .	59
5.2. Ejemplos de localizaciones fallidas de la base de datos UBIRIS. . . . .	61
5.3. Ejemplos de iris con ruido de la base de datos Miles Research. . . . .	61
5.4. Ejemplos de iris con ruido de la base de datos UBIRIS. . . . .	62
5.5. Ejemplos de iris con reflejo de la base de datos UBIRIS. . . . .	62
5.6. Optimización de los parámetros del método ARX por recocido simulado. . . . .	64
5.7. Resultados con mejores desempeños para la base de datos Miles Research . . . .	66
5.8. Resultados con mejores desempeños para la base de datos UBIRIS . . . . .	66
5.9. Pruebas para encontrar los factores de sintonización de recocido simulado . . . .	68
5.10 Resultados de recocido simulado con el método de muestreo Hammersley . . . .	69
5.11 Resultados con mejores desempeños para UBIRIS por recocido simulado . . . .	69
5.12 Resultados del muestreo Hammersley factorial la base de datos UBIRIS . . . . .	71
5.13 Efectividad de los métodos de muestreo en modo identificación. . . . .	72
5.14 Efectividad variando el umbral por el método ARX. . . . .	72
5.15 Efectividad de los métodos de muestreo en modo identificación. . . . .	73
5.16 Efectividad variando el umbral por el método ARX. . . . .	73

5.17.Distribución genuino-impostor por el muestreo Hammersley. . . . .	74
5.18.Distribución genuino-impostor por el método ARX. . . . .	74
5.19.Curva ROC donde se comparan todos los métodos de verificación. . . . .	75
5.20.Comportamiento de los parámetros de las distribuciones genuino-impostor. . . .	75
5.21.Distribución genuino-impostor por el muestreo Hammersley. . . . .	76
5.22.Curva ROC donde se comparan todos los métodos de verificación. . . . .	76
C.1. Distribución genuino-impostor por muestreo aleatorio con reposición para 100 %.	111
C.2. Distribución genuino-impostor por muestreo aleatorio con reposición para 75 %.	112
C.3. Curva ROC por el muestreo aleatorio con reposición. . . . .	112
C.4. Distribución genuino-impostor por muestreo aleatorio sin reposición para 100 %.	113
C.5. Distribución genuino-impostor por muestreo aleatorio sin reposición para 75 %.	113
C.6. Curva ROC por el muestreo aleatorio sin reposición. . . . .	114
C.7. Distribución genuino-impostor por muestreo hammersley para 100 %. . . . .	114
C.8. Distribución genuino-impostor por muestreo hammersley para 75 %. . . . .	115
C.9. Curva ROC por el muestreo hammersley. . . . .	115
C.10.Distribución genuino-impostor por el método ARX para 100 %. . . . .	116
C.11.Distribución genuino-impostor por el método ARX para 75 %. . . . .	116
C.12.Curva ROC por el método ARX. . . . .	117
D.1. Distribución genuino-impostor por muestreo aleatorio con reposición para 100 %.	118
D.2. Distribución genuino-impostor por muestreo aleatorio con reposición para 90 %.	119
D.3. Distribución genuino-impostor por muestreo aleatorio con reposición para 80 %.	119
D.4. Distribución genuino-impostor por muestreo aleatorio con reposición para 70 %.	120
D.5. Distribución genuino-impostor por muestreo aleatorio con reposición para 60 %.	120
D.6. Distribución genuino-impostor por muestreo aleatorio con reposición para 50 %.	121
D.7. Curva ROC por el muestreo aleatorio con reposición. . . . .	121
D.8. Distribución genuino-impostor por muestreo aleatorio sin reposición para 100 %.	122
D.9. Distribución genuino-impostor por muestreo aleatorio sin reposición para 90 %.	122
D.10.Distribución genuino-impostor por muestreo aleatorio sin reposición para 80 %.	123
D.11.Distribución genuino-impostor por muestreo aleatorio sin reposición para 70 %.	123
D.12.Distribución genuino-impostor por muestreo aleatorio sin reposición para 60 %.	124
D.13.Distribución genuino-impostor por muestreo aleatorio sin reposición para 50 %.	124
D.14.Curva ROC por el muestreo aleatorio sin reposición. . . . .	125

---

D.15	Distribución genuino-impostor por muestreo hammersley para 100 %.	125
D.16	Distribución genuino-impostor por muestreo hammersley para 90 %.	126
D.17	Distribución genuino-impostor por muestreo hammersley para 80 %.	126
D.18	Distribución genuino-impostor por muestreo hammersley para 70 %.	127
D.19	Distribución genuino-impostor por muestreo hammersley para 60 %.	127
D.20	Distribución genuino-impostor por muestreo hammersley para 50 %.	128
D.21	Curva ROC por el muestreo hammersley.	128
D.22	Distribución genuino-impostor por el método ARX para 100 %.	129
D.23	Distribución genuino-impostor por el método ARX para 90 %.	129
D.24	Distribución genuino-impostor por el método ARX para 80 %.	130
D.25	Distribución genuino-impostor por el método ARX para 70 %.	130
D.26	Distribución genuino-impostor por el método ARX para 60 %.	131
D.27	Distribución genuino-impostor por el método ARX para 50 %.	131
D.28	Curva ROC por el método ARX.	132

# Índice de tablas

---

2.1. Comparación de tecnologías biométricas . . . . .	19
2.2. Comparación sobre el costo de falsificar algún factor biométrico. . . . .	23
3.1. Desempeño de los algoritmos. . . . .	28
5.1. Clasificación del ruido presentado por la base de datos UBIRIS . . . . .	60
5.2. Preselección de las bases de datos . . . . .	63
5.3. Clasificación de la base de datos Miles Research . . . . .	64
5.4. Clasificación de la base de datos UBIRIS . . . . .	65
5.5. Parámetros de muestreo Hammersley obtenidos por recocido simulado . . . . .	67
5.6. Parámetros de muestreo Hammersley obtenidos por recocido simulado (cont.) . .	67
5.7. Parámetros empleados para el muestreo Hammersley factorial . . . . .	70
5.8. Parámetros empleados para el muestreo Hammersley factorial (cont.) . . . . .	70
A.1. Clasificación de la base de datos Miles Research . . . . .	85
A.2. Resultados del muestreo aleatorio con reposición para 100% y 75% . . . . .	86
A.3. Resultados del muestreo aleatorio sin reposición para 100% y 75% . . . . .	87
A.4. Resultados del muestreo hammersley para 100% y 75% . . . . .	88
A.5. Resultados del muestreo hammersley optimizado para 100% y 75% . . . . .	89
A.6. Resultados del muestreo hammersley optimizado para 100% y 75% . . . . .	90
A.7. Resultados por el método ARX para 100% y 75% . . . . .	91
B.1. Parámetros de muestreo Hammersley obtenidos por recocido simulado . . . . .	93
B.2. Parámetros empleados para el muestreo Hammersley factorial . . . . .	93
B.3. Parámetros empleados para el muestreo Hammersley factorial (cont.) . . . . .	94
B.4. Clasificación de la base de datos UBIRIS . . . . .	94
B.5. Resultados del muestreo aleatorio con reposición para 100% y 90% . . . . .	95

---

B.6. Resultados del muestreo aleatorio con reposición para 80 % y 70 % . . . . .	96
B.7. Resultados del muestreo aleatorio con reposición para 50 % . . . . .	97
B.8. Resultados del muestreo aleatorio sin reposición para 100 % y 90 % . . . . .	98
B.9. Resultados del muestreo aleatorio sin reposición para 80 % y 70 % . . . . .	99
B.10 Resultados del muestreo aleatorio sin reposición para 50 % . . . . .	100
B.11 Resultados del muestreo Hammersley para 100 % y 90 % . . . . .	101
B.12 Resultados del muestreo Hammersley para 80 % y 70 % . . . . .	102
B.13 Resultados del muestreo Hammersley para 50 % . . . . .	103
B.14 Resultados del muestreo Hammersley optimizado para 100 % y 90 % . . . . .	104
B.15 Resultados del muestreo Hammersley optimizado para 80 % y 70 % . . . . .	105
B.16 Resultados del muestreo Hammersley optimizado para 100 % y 90 % . . . . .	106
B.17 Resultados del muestreo Hammersley optimizado para 80 % y 70 % . . . . .	107
B.18 Resultados del muestreo Hammersley factorial para 100 % y 90 % . . . . .	108
B.19 Resultados del muestreo Hammersley factorial para 100 % y 90 % (cont.) . . . . .	109
B.20 Resultados por el método ARX para 100 %, 90 %, 70 %, 60 % y 50 % . . . . .	110

# Introducción

---

Conforme avanza la tecnología las necesidades de la sociedad van cambiando, una de ellas ha sido las medidas de seguridad implementadas para acceder a ciertos lugares o a cierta información. En el siglo pasado era común ver que la información considerada como confidencial ser guardada bajo llave en archiveros especiales o en cajas fuertes, el acceso a lugares era por medio de una tarjeta o la posesión de una llave. Recientemente las medidas de seguridad han avanzando, términos como número de identificación personal, firmas digitales, certificados, son conocidos como medidas de seguridad [45].

Por desgracia al igual que han aumentado las medidas de seguridad también han aparecido individuos con tendencias delictivas los cuales se han encargado de sobrepasar todas estas herramientas que resguardan información confidencial o el acceso a lugares restringidos. Una respuesta que ha tomado fuerza en seguridad es la tecnología biométrica, que hasta el momento a arrojado resultados sorprendentes, basándose en reconocimiento de huellas digitales, de la voz, del iris, entre otros [21].

La principal característica de la autenticación biométrica es el uso de algún rasgo distintivo que permita ser comparado con otro, tomado normalmente en ese instante para analizar si son iguales o no; ese rasgo es conocido como factor biométrico. El reto es conseguir que las máquinas puedan llevar a cabo estas operaciones de forma rápida y libre de errores. Algunos de estos dispositivos reconocedores de rasgos biométricos han pasado de las películas de ciencia ficción a catálogos de muchos fabricantes.

Una de las ventajas principales de estos sistemas de seguridad es que son mucho más cómodos y seguros que los sistemas tradicionales, haciendo el sistema de seguridad más confiable en la actualidad [45].

El funcionamiento general de un sistema biométrico se inicia capturando una muestra del biométrico del individuo por medio de un sensor adecuado para la aplicación, con dicha muestra se realiza alguna función matemática para poder representarla como una plantilla<sup>1</sup>, la cual contendrá las principales características del biométrico. La información permite la comparación del biométrico contra otras plantillas previamente almacenadas para definir la identidad del individuo [44].

Una de las ventajas del reconocimiento del iris según [44] es que no requiere de contacto físico con el sensor, lo cual lo hace un método no invasivo y con una mayor probabilidad de ser aceptado por los usuarios. El reconocimiento por iris es una de las formas más seguras de autenticación que existe y representa una auténtica barrera para la intrusión de personas. Dos propiedades destacan del iris en su papel de identificación: es un órgano protegido debido a que se encuentra detrás de la córnea y la úvea [7]; clínicamente es imposible modificarlo sin correr riesgos de perder la visión [44].

---

<sup>1</sup>Del término en inglés *template*, es la representación de un factor biométrico que contiene la información de las principales características distintivas.



Una de las aplicaciones más famosas es la del aeropuerto británico Heathrow en London uno de los más transitados del mundo, lugar donde se introdujo un sistema de reconocimiento de iris como parte del control de inmigración [3].

El científico que inventó la fórmula matemática para leer la información del iris es el doctor John Daugman, profesor de la Universidad de Cambridge. Uno de los logros con el que se dio a conocer fue la identificación de la joven afgana que fue portada de la revista de National Geographic en 1984 y que fue descubierta 18 años después. El algoritmo inventado por Daugman permitió establecer que los ojos de esa envejecida mujer eran iguales a los de la portada en 1984 [27].

Actualmente se cuentan con un gran número de algoritmos, sin embargo el del doctor Daugman mantiene su liderato como el método más efectivo en el reconocimiento de iris ofreciendo excelentes resultados. Su algoritmo fue patentado desde 1991 y ha sido incorporado por diversas empresas encargadas de sistemas de seguridad y autenticación de personas [15].

El funcionamiento general de un sistema de reconocimiento de iris consiste en un equipo de adquisición de imágenes que cuenta con una cámara de video, un sistema de iluminación imperceptible para el ojo humano y una pantalla que permita desplegar la posición del operador. Una vez que se tiene la imagen se procede a localizar los límites del iris y la pupila para delimitar la zona de análisis. Con el área obtenida en la etapa de localización se extraen los rasgos y las características principales que permiten representar al iris como una plantilla. Como último paso se tiene la comparación, con la cuál se puede concluir si dos pares de iris son idénticos o no, o bien si el iris pertenece a una base de datos recopilada previamente [42].

En este trabajo se pretende diseñar e implementar un algoritmo de reconocimiento de iris basado en métodos estadísticos desarrollando e implementando un algoritmo en la etapa de extracción y comparación de plantillas que representen al iris de una persona. La decisión de que sean métodos estadísticos es por las ventajas que ofrece ante la variabilidad presentada en el proceso de reconocimiento, es imposible tomar la imagen del iris sin que se presente este factor.

Las líneas de investigación tuvieron dos vertientes, una de ellas es la caracterización del iris como un modelo estocástico en una de sus representaciones más sencillas y conocidas, como lo es el modelo Autoregresivo con una entrada exógena (ARX<sup>2</sup>). La segunda línea es el uso de muestreo estadístico para la representación de los rasgos del iris, está fue el área más explorada por las ventajas que puede ofrecer ante la variabilidad presentada al capturar las imágenes con ruido.

También fue empleada la herramienta de recocido simulado para la optimización de los modelos; en el ARX se implementó para encontrar el número de parámetros óptimos que representan mejor al iris y en el método de muestro para encontrar los parámetros que brinden una maximización de su desempeño o efectividad.

## 1.1. Antecedentes

En la actualidad los sistemas de reconocimiento más restringidos emplean indicadores biométricos para establecer patrones que ayuden a la identificación y verificación de personas para el control de acceso. La biometría se basa en la medición de características anatómicas o comportamientos distintivos que provienen de algún rasgo biológico complejo y único; los ejemplos más familiares incluyen las huellas digitales, reconocimientos de voz, geometría de la mano, entre otros. A pesar de que tienen rasgos únicos, debido a sus características universales muestran un cierto grado de variabilidad aleatoria y algunos son considerados como altamente invasivos [10]. Típicamente, el usuario requiere de tener contacto físico con el dispositivo de sensado o tomar parte de alguna acción especial como recitar una específica

<sup>2</sup>Por sus siglas en inglés: AutoRegresive with eXogenous input

secuencia fonética [25]. Existen pocos procedimientos no invasivos, entre los cuáles se pueden mencionar el reconocimiento de características faciales o el reconocimiento de iris.

[44] menciona que la estructura del iris es única en cada individuo y que existen dos fuentes de información para reforzar esta afirmación. La primera evidencia es un estudio clínico donde se observaron un gran número de ojos, oftalmólogos y anatomistas notaron que entre el ojo izquierdo y derecho de la persona se observaron muchas diferencias en la estructura. La segunda evidencia es en el campo de la biología, donde se encontró que la estructura general del iris está genéticamente determinada y las particularidades minuciosas dependen críticamente de las circunstancias del desarrollo en la etapa fetal, por lo que es altamente improbable replicar las circunstancias naturales que lo formaron.

Las técnicas enfocadas al reconocimiento de iris se basan en los detalles que presenta la textura del iris, que por sus características anatómicas complejas cuenta con grietas, criptas, estrías, huecos, pecas, collaretes, entre otras, conformando una textura altamente detallada y única para cada individuo [2]. En la figura 1.1 se pueden observar algunas de ellas.

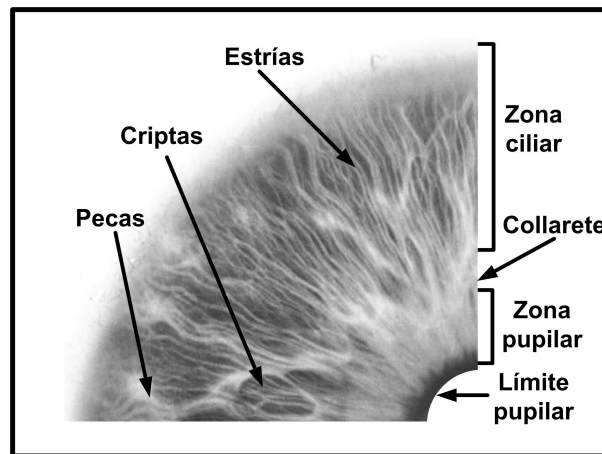


FIGURA 1.1: Estructura del iris vista de la parte frontal.

Estas características hacen que el tejido del iris un órgano distintivo, que puede ser utilizado como elemento de reconocimiento al ser capturada su imagen bajo ciertas condiciones de iluminación.

### Principales etapas del reconocimiento de iris

De acuerdo con [42] las etapas básicas del proceso de identificación del iris humano se dividen en:

1. **Adquisición de la imagen:** Es capturada una serie de imágenes de alta calidad usando una cámara de video con una iluminación especial que es imperceptible para el ojo humano.
2. **Localización del iris:** Son delimitados los bordes de la pupila y el limbo en la imagen adquirida del iris.
3. **Normalización:** La imagen del iris puede ser capturada en diferentes tamaños, incluso la imagen de una misma persona al ser capturada por segunda vez puede tener variaciones debido a la iluminación u otros factores, por lo que es estandarizada para que pueda ser comparada con muestras tomadas anteriormente. El objetivo aquí es entregar una imagen con tamaño y formatos únicos.
4. **Extracción de características:** El iris provee abundante información debido a su textura. Es extraída la información o rasgos del iris para formar un plantilla con el fin de

representarlo, antes de extraer las características la mayoría de los algoritmos aplican filtros o técnicas de preprocesamiento de imágenes para resaltar las características de la plantilla.

5. **Comparación y decisión:** La información de la plantilla es clasificada y comparada mediante diversas técnicas, en esta etapa se pueden reconocer los modos de operación de un sistema biométrico, el modo de verificación y el modo de identificación dependiendo de la aplicación del sistema de reconocimiento de iris.

En la actualidad se cuentan con diversos métodos de reconocimiento basados en diversas técnicas de procesamiento de imágenes y señales aplicando algoritmos como redes neuronales, lógica difusa, transformada wavelet, entre otras. Sin embargo el pionero en el reconocimiento de Iris es el Dr. John Daugman de la Universidad Inglesa de Cambridge, su algoritmo se encuentra implementado en la mayoría de los sistemas actuales de reconocimiento de iris ofreciendo muy buenos resultados.

## 1.2. Planteamiento del problema

De acuerdo a [42] los algoritmos descritos en su trabajo no fueron probadas con pruebas que tuvieran muchas variaciones de luz y tamaño, por lo regular son implementados con la base de datos CASIA [34], la cual se encuentra disponible para el dominio público, incluye 756 imágenes de iris de 108 individuos. De acuerdo a [36] esta base de datos se encuentra libre de ruido por lo que beneficia el desempeño reportado por cada algoritmo.

El enfoque de este trabajo de investigación se encuentra dirigida a la implementación de métodos estadísticos en la etapa de la extracción de las características del iris, para introducir un método eficiente que reduzca el efecto del ruido por la variabilidad y con resultados similares o mejores a los algoritmos desarrollados actualmente.

## 1.3. Objetivos de la investigación

El objetivo general es diseñar un algoritmo que sea robusto al momento de realizar la identificación de personas. Esta robustez implica reducir los efectos de una deficiente iluminación, mal enfoque de la imagen u obstrucciones de párpados y pestañas.

Se busca profundizar en aspectos científico-técnicos referentes a la identificación de iris y aprovechar las aplicaciones de métodos estadísticos. Las principales áreas que se abordarán en esta tesis serán:

- Selección, diseño e implementación de un algoritmo que permita extraer las principales características del iris por medio de métodos estadísticos.
- Selección, diseño e implementación de un algoritmo que permita validar las características extraídas del iris con una base de datos que contendrá datos de otros iris.

## 1.4. Justificación

Debido a que el algoritmo de Daugman está patentado, mucha información que permite la implementación se encuentra limitada, incluso existen diferencias al momento de reportar resultados del algoritmo Daugman cuando es implementado para poder compararse con algún método en particular.

Se busca crear tecnología propia en este rubro que permita desarrollos científicos relacionados con el tema como por ejemplo la criptografía del iris.

Esta tesis pretende aportar el diseño de un nuevo sistema que permita la extracción y comparación de las características del iris. A pesar de los avances logrados en las últimas dos décadas aún pueden realizarse mejoras a los algoritmos actuales como los porcentajes de error, el tiempo de identificación, entre otros [15].

Los algoritmos desarrollados por [11], [44], [4], [46], [28], [29], [41] y [32] no han explorado la representación del iris por métodos estadísticos, siendo un tema con muchas áreas de oportunidad. El empleo de métodos estadísticos permite recopilar la información empleando como principal característica el azar y la probabilidad para contrarrestar la variabilidad de los procesos [30]. La variabilidad en este trabajo de investigación se presenta al tomar las imágenes del iris en diferentes condiciones obteniendo cada vez imágenes diferentes.

## 1.5. Contribuciones

El emplear técnicas que no han sido tan exploradas abren puerta a una gran gama de contribuciones, sin embargo entre las principales se tienen:

- Introducción de un método de muestreo estadístico para representar los rasgos o características principales del iris.
- Representación del iris a través de modelos estocásticos como lo es el ARX.
- Introducción del método de optimización de recocido simulado para encontrar los mejores parámetros de muestreo y del modelo ARX.

## 1.6. Alcances y limitaciones

El trabajo de investigación se enfocará en las etapas de extracción y comparación, para la etapa de adquisición de la imagen del iris no se realizará ninguna aportación; se trabajará con bases de datos que presentan ruido y son de dominio público.

El algoritmo será desarrollado en MATLAB® por lo tanto los tiempo de ejecución de cada etapa no podrán ser sujetos a compararse con métodos desarrollados por otros lenguajes. Una de las principales características de MATLAB® es que es un lenguaje de alto nivel basado en vectores, arreglos y matrices; por lo tanto cuando se implementen los algoritmos que involucren los ciclos *for* la velocidad de procesamiento se verá reducida en gran medida.

## 1.7. Trabajos relacionados

Se cuenta con un sistema que ya se encarga de realizar la identificación de iris y éste se muestra en la figura 1.2 y 1.3. Este sistema fue implementado por [18], está basado principalmente en el algoritmo de Daugman por lo que muchas de las etapas del reconocimiento de iris son muy similares a las mencionadas en los antecedentes. Entre las principales funciones con las que cuenta el algoritmo implementado, se pueden mencionar las siguientes:

1. **Redimensionamiento (Pleerjpg):** Una vez capturada la imagen, puede presentarse en muy diversos tamaños; por lo regular cuenta con un alto grado de resolución por lo que es necesario redimensionarla para trabajar con menos pixeles y estandarizar el tamaño de las imágenes.
2. **Localización de iris (Piris1ci):** Muy similar a la etapa de localización, en la cuál implementa la ubicación de los límites limbo y pupilar del iris utilizando el algoritmo de una versión discreta de la ecuación(3.2).

Extrae el iris y convierte de coordenadas polares a coordenadas rectangulares empleando las ecuaciones (3.17) y (3.18); la salida de esta función es una matriz de pixeles (imagen) de  $n$  renglones y  $m$  columnas.

3. **Filtros de Gabor (fft2d8):** En esta última función de la figura 1.2 se implementan los filtros de Gabor, empleando la ecuación (3.4); la salida de esta función es un conjunto de  $K$  matrices de números complejos de tamaño  $m \times n$ . Donde  $K$  depende del número de frecuencias y el ángulo de orientación de los filtros de Gabor.

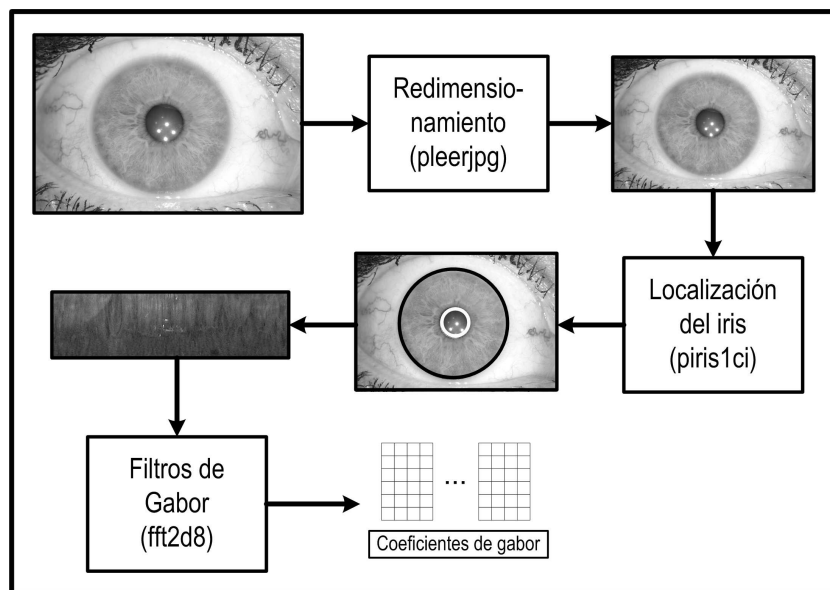


FIGURA 1.2: Esquema de la primera parte del sistema ya implementado basado en el algoritmo de Daugman.

En el figura 1.3 se muestra el procedimiento que se lleva a cabo en la comparación del iris con una base de datos y la principal función de esta etapa se describe a continuación:

1. **Comparación y decisión (Cmpdirdb):** Divide las matrices de los coeficientes de Gabor de cada iris en cuadrículas de tamaño variable, para extraer de cada cuadrícula parámetros estadísticos (actualmente la media) y hacer demodulación de fase (ver figura 3.3), de esta manera se realiza la codificación a bits.

También implementa la comparación del iris muestra (iris sample) contra una base de datos y decide en base a un criterio (porcentaje de bits iguales) si acepta o rechaza la existencia del iris muestra en la base de datos.

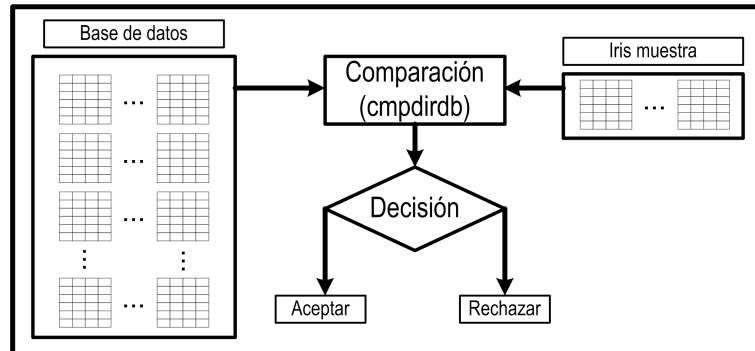


FIGURA 1.3: Esquema del procedimiento de comparación de datos.

## 1.8. Descripción de la metodología del trabajo

Se realizó un estudio de los métodos y algoritmos actuales de reconocimiento de iris para conocer el estado del arte y como ha ido evolucionando para analizar los resultados obtenidos hasta el momento.

Antes de definirse los métodos estadísticos, se estudió el sistema que ya se encuentra implementado y se identificaron las fases de cada etapa, para determinar cuales eran los puntos críticos del reconocimiento de iris. Se comprendió el algoritmo empleado probando sus resultados para poder trabajar sobre él y aprovechar cada área de oportunidad.

La selección de los métodos estadísticos a implementar fue uno de los principales puntos de la tesis; para obtener los resultados esperados fue necesario conocer las ventajas y desventajas para determinar cual era el más factible de implementar y que permitiera obtener mejores resultados.

Una vez seleccionados los métodos adecuados se implementó el algoritmo para realizar pruebas de validación con una base de datos simulando un pequeño sistema de acceso. Por último fueron comparados todos los resultados para probar las bondades de cada método.

## 1.9. Contenido

En el capítulo dos se introduce al tema de reconocimiento biométrico, donde muchos de los conceptos ahí expuestos son base para entender los resultados y conclusiones expuestos por este trabajo de investigación.

El tercer capítulo está dedicado al estado del arte sobre el reconocimiento automático de personas basado en el análisis del iris. Se exponen a grandes rasgos los principales algoritmos desarrollados hasta el momento y se profundiza más en cuatro de ellos, los cuales fueron comparados por [42].

El capítulo cuatro son presentados los conceptos básicos para entender las soluciones presentadas para el problema expuesto en la sección 1.2. Son explicados el método de muestreo y el ARX, métodos estadísticos usados para el análisis del iris. También son implementadas herramientas de optimización como recocido simulado y muestreo factorial.

El quinto capítulo se presenta el resumen de los resultados en los modos de operación de un sistema biométrico: verificación e identificación. Son presentados los resultados como fue evolucionando el tema de acuerdo a los conceptos expuestos en el capítulo cuatro.

El último capítulo presenta las conclusiones a las que se llegaron y las propuestas de posible desarrollo del tema expuesto en este trabajo.

En los apéndices son presentados todos los resultados obtenidos en cada corrida efectuada, cada experimento tiene el método con el que fue hecho y su nomenclatura. También se agrega una breve descripción del código fuente de los programas implementados en MATLAB®.

# Introducción a los métodos de reconocimiento biométricos

---

Desde la antigüedad el hombre ha tratado de controlar el acceso a determinados lugares o a cierta información, como sobres lacrados con el sello real, el conocimiento de alguna frase, la utilización de algún uniforme o la posesión de una llave. En la actualidad se han sustituido estos objetos por contraseñas, números de identificación personal, certificados y firmas digitales, entre otras.

Sin embargo estos objetos o datos pueden ser robados, falsificados, filtrados o deducidos; es fácil conocer la contraseña de una persona o adivinar un número de identificación personal. Para permitir autenticar a una persona, acceder a un lugar, efectuar una transacción bancaria o realizar una compra, se deben buscar métodos que no dependan de una llave determinada, sino que la propia persona sea la llave que le permita autenticarse. Es aquí donde entra una de las aplicaciones de la biometría [35].

El término de biometría viene del vocablo griego *bios* que significa vida y de *metrikos* cuyo significado es medida. Biometría es un conjunto de métodos automatizados de identificación y verificación de la identidad de una persona, basados en alguna característica fisiológica con rasgos unívocos del individuo, la cual es analizada y medida para crear una plantilla que puede ser almacenado en una base de datos y recuperado para su comprobación posterior [39].

A pesar de lo desconocido que suene el término, los humanos usan intuitivamente las características del cuerpo, como la cara, su andar o la voz, para reconocerse unos a otros. En los últimos años se han desarrollado una gran variedad de aplicaciones que requieren la verificación o identificación confiable de un individuo, por lo que han surgido interesantes tecnologías de reconocimiento basado en las características del cuerpo humano. Los tradicionales sistemas de verificación e identificación usados para el acceso a lugares restringidos como tarjetas o contraseñas pueden ser violados fácilmente y son poco confiables, mientras que un factor biométrico no puede ser prestado, robado u olvidado y falsificarlo es prácticamente imposible [21].

Una de las primeras aplicaciones de la biometría como medio de identificación de individuos fue realizada Alphonse Bertillon, jefe de la División de Identificación Criminal en el Departamento de Policía en París a mediados del siglo XIX. Bertillon desarrolló un sistema en el cual una persona era candidata a tres posibles categorías a partir de la medida del diámetro de su cabeza. Posteriormente, subdividió esas categorías utilizando la medida del dedo medio, y sucesivamente así con 11 diferentes medidas del cuerpo. A partir de sus mediciones, Bertillon calculó que la probabilidad de que dos personas tuvieran exactamente las mismas once medidas corporales era de una en cuatro millones. Y la posibilidad de que un criminal pudiera hacer algo por modificar su diámetro craneal era nula, gracias a su método se identificaron 800 individuos en tres años.



En 1903, un detenido llamado Will West fue examinado por los guardias, y sus medidas altura, brazos, cabeza, orejas y nariz fueron registradas bajo el sistema Bertillon. Aunque Will West enfrentaba su primera detención, el guardia recordaba haber tomado sus medidas con anterioridad. En los archivos de la prisión se encontró una medición igual perteneciente a un hombre llamado William West, quien había sido encarcelado dos años antes en Leavenworth y que seguía cumpliendo su condena en esa prisión: Will West y William West eran como una ironía de la coincidencia, dos hombres distintos, pero sus medidas bajo el sistema de Bertillon eran virtualmente idénticas.

Pocos años después, los oficiales de la prisión encontraron una característica distintiva entre los West: sus huellas digitales eran inconfundiblemente diferentes. Fue entonces que los departamentos de policía y las prisiones europeas y estadounidenses cambiaron al sistema de identificación de huellas digitales desarrollado por Sir Richard Edward Henry de Scotland Yard. Con esta medida, no sólo se obtuvo mayor certeza en la identificación, sino que se facilitó el proceso; era más fácil pasar por tinta el dedo del sospechoso que presionar la cabeza para tomar su medida craneal exacta [23].

## 2.1. Requerimientos de un factor biométrico

Cualquier humano cuenta con características fisiológicas y/o conductas que pueden ser usadas como una característica biométrica, sin embargo deben satisfacer los siguientes requerimientos:

- *Universalidad*: cada persona debe contar con la característica biométrica.
- *Distintividad*: dos personas cualquiera deben ser lo suficientemente diferentes en término de la característica evaluada.
- *Durabilidad*: la característica debe ser lo suficientemente invariable (con respecto a un criterio de validez) por encima de un periodo de tiempo.
- *Coleccionabilidad*: la característica puede medirse cuantitativamente.

Sin embargo, en la práctica un sistema biométrico tiene problemas que deben ser considerados, entre los más importantes se encuentran:

- *Desempeño*: los recursos requeridos para lograr la exactitud y velocidad deseadas, así como los factores operacionales y ambientales que lo afecten.
- *Aceptabilidad*: que indica hasta que punto las personas están dispuestas a aceptar el uso de un particular identificador biométrico en su vida diaria.
- *Engaño*: que tan fácilmente puede engañarse por métodos fraudulentos.

Un sistema biométrico debe encontrarse con especificaciones de exactitud y velocidad de reconocimiento, recursos requeridos, peligros que presentan para los usuarios, el nivel de aceptación de la población y deben ser lo suficientemente robustos a los métodos fraudulentos o ataques al sistema [26].

## 2.2. Sistemas de reconocimiento biométrico

Un sistema de reconocimiento biométrico es esencialmente un sistema de reconocimiento de patrones que opera adquiriendo datos biométricos de forma individual, extrayendo las características de los datos adquiridos y comparando estas contra una plantilla que pertenece a una base de datos. Dependiendo del contexto de la aplicación, un sistema biométrico puede operar en modo de identificación o verificación:

- *En el modo de verificación*, el sistema valida la identidad de una persona comparando el dato biométrico capturado contra su patrón biométrico que se encuentra en la base de datos. En un sistema, un individuo que decide ser reconocido pide una identidad, usualmente por medio de un número de identificación personal, un usuario o una tarjeta inteligente. La verificación de una persona es usada típicamente para el reconocimiento positivo, donde el objetivo consiste en que múltiples personas usen un mismo modo de identidad.
- *En el modo de identificación*, el sistema de reconocimiento de un individuo por medio de la búsqueda de un patrón de todos los usuarios en una base de datos para relacionarlo. Por consiguiente, el sistema maneja una comparación para establecer la identidad individual (Ver figura 2.1). La identificación es un componente crítico de las aplicaciones de un reconocimiento negativo, donde el sistema establece a que persona le niega el acceso. El propósito del reconocimiento negativo previene que una persona use varias identidades. La identificación puede ser usada en el reconocimiento positivo pidiendo una identidad con los tradicionales métodos de reconocimiento personal como contraseñas, números de identificación personal, llaves o fichas; el reconocimiento negativo sólo puede establecerse por métodos biométricos [26].

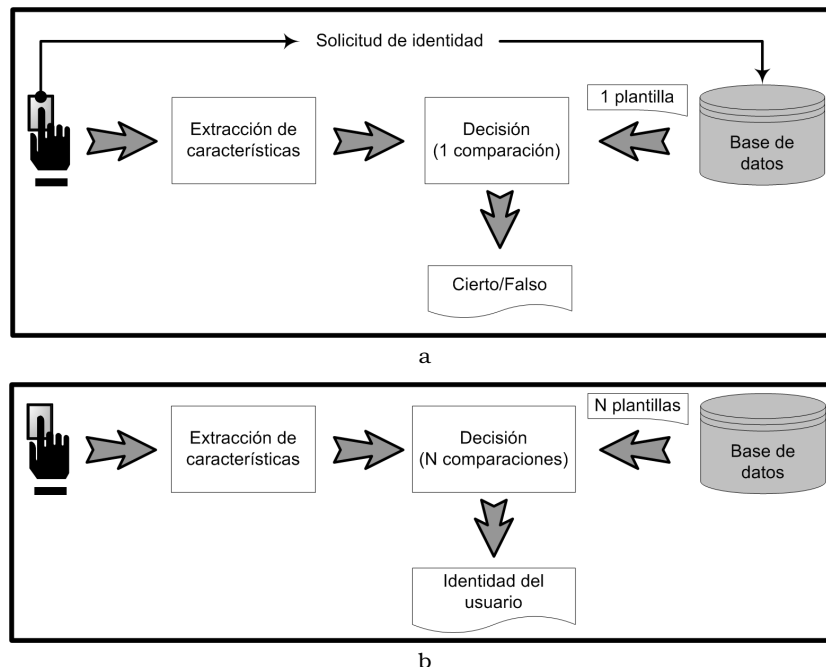


FIGURA 2.1: Diagrama de bloques de verificación e identificación. a) En modo verificación el usuario pide una identidad y el sistema autentifica si su identidad es igual a la identidad solicitada. b) En modo de identificación el usuario es comparado con el registro de toda la base de datos para determinar su identidad [25].

El diseño general de un sistema biométrico se basa en cuatro módulos:

- *Coleccionador de datos*: el rasgo o dato biométrico de un individuo es capturado con ayuda de un sensor; se deben cuidar aspectos como el mantenimiento y las condiciones de operación del sensor para su correcto funcionamiento y el éxito del reconocimiento. Para pasar a la siguiente etapa es necesario comprimir la información del sensor para poder transmitirla o para almacenarla en una base de datos.
- *Extracción de características*: el identificador biométrico es procesado para extraer los rasgos característicos, por lo regular se intentan aplicar previamente técnicas de preprocesamiento para poder resaltar ciertos rasgos que permitan normalizar y estandarizar las plantillas.
- *Comparación*: las plantillas son comparadas dependiendo del modo de operación (verificación e identificación) contra una base de datos tomada con anterioridad.
- *Decisión*: cuando la identidad del usuario es establecida, o es aceptada o rechazada según sea el modo de operación del sistema biométrico [43].

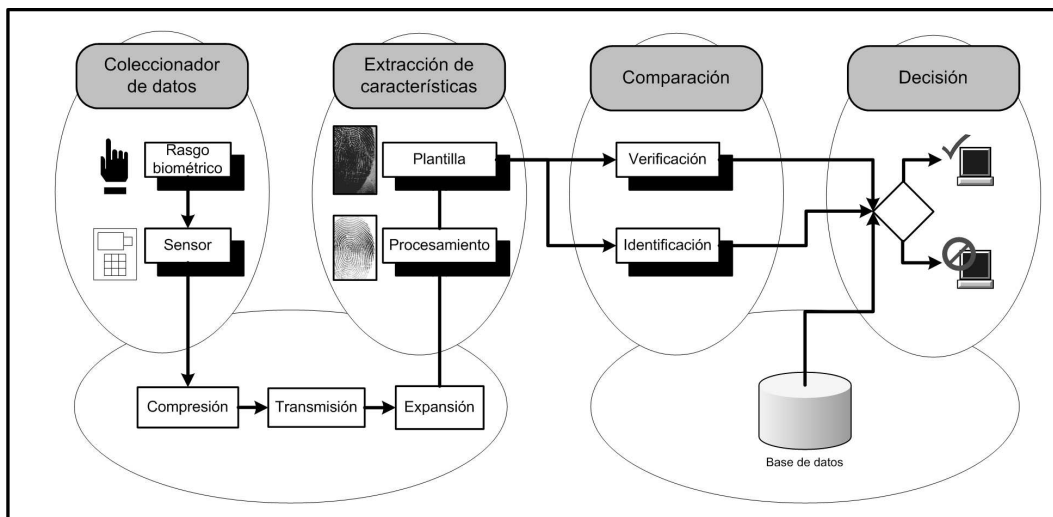


FIGURA 2.2: Modelo general de un sistema de identificador biométrico.

## 2.3. Errores de un Sistema Biométrico

Dos plantillas de un mismo factor biométrico de una persona (por ejemplo, dos impresiones de la huella digital del dedo pulgar de la mano derecha) no son exactamente iguales debido a las condiciones imperfectas de la imagen (polvo o dedos sucios), cambios fisiológicos del usuario (cortaduras o daños al dedo), condiciones ambientales (temperatura y humedad) y la interacción del usuario con el sensor (lugar del dedo al tomar la imagen). Por lo tanto, la respuesta de un sistema biométrico debe cuantificar las similitudes entre la entrada  $X_M$  y la plantilla  $X_P$ . El sistema de decisión es regulado por el umbral  $t$ . Pares de muestras biométricas que generan resultados mayores o iguales al umbral  $t$  se infiere que provienen de la misma persona, y si generan resultados menores al umbral  $t$  se infiere que las muestras provienen de diferentes personas. Las distribuciones de los resultados generados de los pares de muestras de diferentes personas es conocida como la distribución del impostor, mientras que los resultados generados de los pares de muestras de la misma persona es conocida como la distribución genuina (Ver figura 2.3) [25].

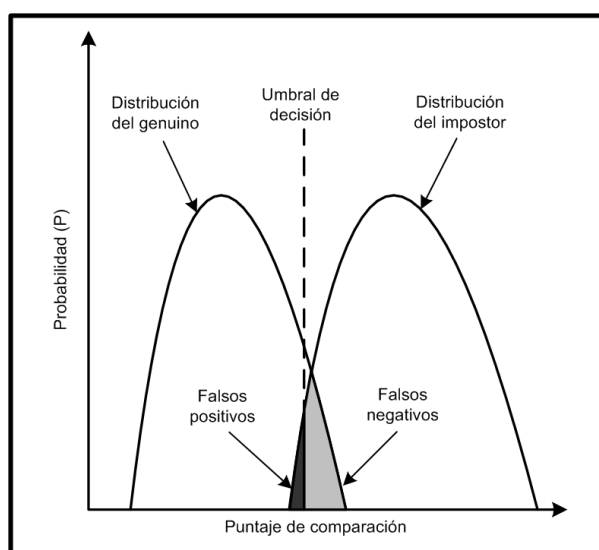


FIGURA 2.3: Proporción del error de un sistema biométrico. Las curvas muestran las distribuciones del impostor y el genuino dada por el umbral de decisión  $t$ .

Un sistema de verificación biométrica puede tener dos tipos de errores:

1. Error en las mediciones biométricas donde dos diferentes personas son relacionadas como la misma persona, mejor conocida como Proporción de Falsos Positivos (PFP).
2. Error en dos mediciones biométricas de la misma persona que provienen de dos diferentes personas, mejor conocida como Proporción de Falsos Negativos (PFN).

Cualquier sistema biométrico hace un intercambio entre PFP y PFN, las cuales son funciones del umbral  $t$ . Si  $t$  decrece provoca que el sistema sea más tolerante a variaciones en la entrada y ruido, entonces PFP aumenta. Si  $t$  incrementa, hace que el sistema sea más seguro, entonces PFN aumenta. Si se representan todos los puntos de operación del umbral  $t$  se puede describir la curva de Características Operativas del Receptor mejor conocidas como curva ROC (Receiver Operating Characteristic) que se muestra en la figura 2.4.

Matemáticamente los errores de un sistema de verificación pueden ser calculados. Si el patrón biométrico del usuario  $P$  es representado por  $X_P$  y la entrada adquirida para el reconocimiento es representada por  $X_M$ , la hipótesis nula y alternativa pueden plantearse como:

- $H_0 : X_M \neq X_P$
- $H_1 : X_M = X_P$

Las decisiones asociadas son las siguientes:

- $D_0$ : La persona no es quien el sistema dice que es.
- $D_1$ : La persona es quien el sistema dice que es.

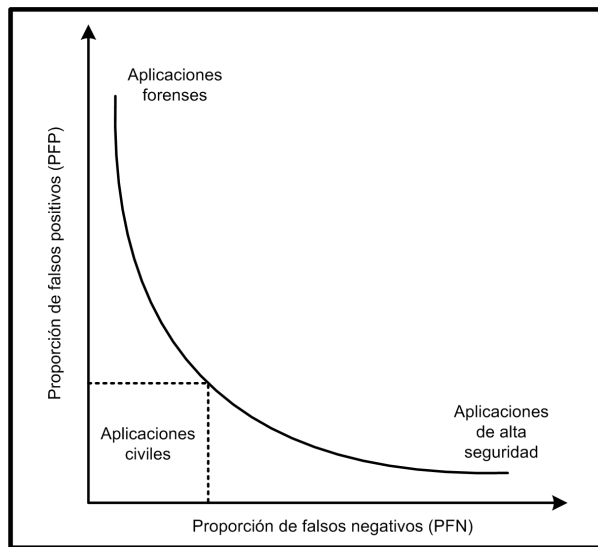


FIGURA 2.4: Curva de Características Operativas del Receptor donde se ejemplifica los diferentes tipos de aplicaciones biométricas en función de PFP (Proporción de Falsos Positivos) y PFN (Proporción de Falsos Negativos).

Si se siguen las reglas de decisión y la manera en que está formulada la hipótesis se pueden tener dos tipos de errores:

- Tipo I: Falso positivo ( $D_1$  es decidido cuando  $H_0$  es verdadera).
- Tipo II: Falso negativo ( $D_0$  es decidido cuando  $H_1$  es verdadera)

PFP es la probabilidad del error tipo I y PFN es la probabilidad del error tipo II:

$$PFP = P(D_1|H_0) \quad (2.1)$$

$$PFN = P(D_0|H_1) \quad (2.2)$$

La expresión  $(1-PFN)$  es conocida como la potencia de la prueba de hipótesis y los valores tanto de PFP y PFN dependerán de las aplicaciones del sistema biométrico como se muestra en la figura 2.4 [26].

## 2.4. Factores biométricos más usados

Existen un gran número de características biométricas y son usadas en diversas aplicaciones. Cada biométrico tiene sus ventajas y desventajas, la elección de cada uno de ellos depende de la aplicación. En pocas palabras no existe un factor biométrico óptimo. La relación entre un factor biométrico y una aplicación es determinado por el modo operacional de la aplicación y las propiedades de éste.

### Ácido Desoxirribonucleico (ADN)

El ADN es de una dimensión (1-D) y presenta un código único para cada individuo, excepto por el hecho de que los gemelos idénticos tienen patrones de ADN semejantes. Sin embargo, es muy usado en el contexto de aplicaciones forenses para el reconocimiento de personas. Tres problemas limitan la utilidad de este identificador biométrico para otras aplicaciones:

1. Contaminación y sensibilidad: es fácil robar una pieza de ADN para cualquier propósito.
2. Problemas de reconocimiento en tiempo real: la tecnología para reconocer el ADN requiere de incómodos procesos químicos y de algún experto en la materia.
3. Problemas de privacidad: la información puede ser usada con fines de lucro.

### Oído

Se ha sugerido que la forma del oído y la estructura del tejido cartilaginoso del pabellón auditivo son distintivos. El reconocimiento del oído aprovecha la distancia de los puntos prominentes del pabellón auditivo para identificar la localización del oído. Los rasgos de un oído no son muy distintivos para establecer la identidad de un individuo.

### Rostro

El reconocimiento de rostro es un método no invasivo y las imágenes faciales son probablemente de los factores biométricos más usados para hacer reconocimiento de individuos. El reconocimiento facial se basa principalmente en la localización y la forma de los ojos, cejas, nariz, labios y barbilla, así como sus relaciones espaciales. Los sistemas basados en esta tecnología presentan dificultades en el reconocimiento debido a los cambios drásticos al momento de tomar las imágenes, estados de ánimo y las diferentes condiciones de iluminación.

### Termogramas infrarrojos faciales, de la mano y las venas de la mano

Los rasgos de esta característica biométrica pueden ser capturados por una cámara infrarroja, es un sistema no invasivo, pero la adquisición de la imagen es tomada bajo ambientes no controlados que presentan emisiones de calor (como cuartos calientes o tuberías de líquidos calientes) en la vecindad del cuerpo. Los sensores infrarrojos son excesivamente caros lo que impide que se difunda el uso de termogramas.



FIGURA 2.5: Ejemplos de factores biométricos presentados por [26]: a)ADN. b)Oído. c)Rostro. d)Termograma facial

### Huellas digitales

Este ha sido el método más usado para la identificación personal, las características de las crestas y valles de la huella digital son determinadas a los siete meses del desarrollo del feto. Una de las ventajas es que las huellas digitales son diferentes entre gemelos idénticos e incluso entre cada dedo de la misma persona. Actualmente el escáner de huellas digitales tiene un costo muy accesible convirtiéndolo en uno de los sistemas biométricos más económicos. Uno de los problemas que presenta el reconocimiento de huellas digitales es que requieren de un gran número de recursos computacionales especialmente cuando esta en el modo de operación y en una pequeña fracción de la población puede ser poco conveniente debido a factores genéticos, envejecimiento u ocupación del individuo.

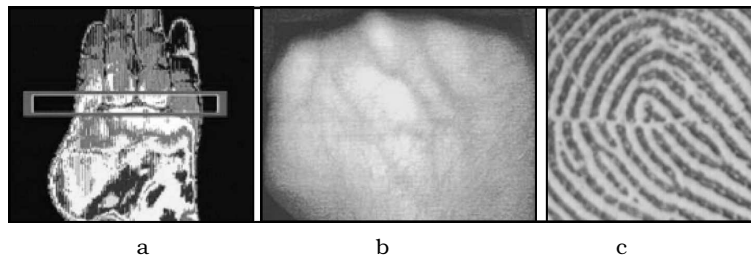


FIGURA 2.6: Ejemplos de algunos factores biométricos presentados por [26]: a) Termograma de la mano. b) Venas de la mano. b) Huellas digitales

### Forma de caminar

La forma de caminar no es un método muy distintivo, pero es suficientemente discriminatorio la verificación en aplicaciones de baja seguridad. La forma de caminar es un método biométrico conductual y no permanece invariante, especialmente en un largo periodo de tiempo, debido a las fluctuaciones del peso corporal, lesiones o ebriedad. Los sistemas basados en la forma de caminar usan una secuencia de video para observar la longitud de la zancada y medir los diferentes movimientos de cada articulación, esto implica muchos factores y un gran número de cálculos computacionales.

### Geometría de los dedos y la mano

El reconocimiento en la geometría de la mano está basado en un gran número de medidas como la forma, el tamaño de la palma, la longitud y anchura de los dedos. La verificación basada en la geometría de la mano se encuentra instalada en cientos de lugares en el mundo, la técnica es muy simple, relativamente fácil de usar y barata. Algunos de los problemas que presenta son:

1. Factores ambientales como el clima seco o húmedo afectan negativamente la exactitud del sistema.
2. La geometría de la mano no es muy distintiva para la identificación de un individuo en una gran población.
3. La información de la mano varía mucho en el desarrollo de los niños.

Algunos de los sistemas de verificación están basados en las medidas de unos cuantos dedos (típicamente el índice y medio) en lugar de toda la mano, lo cual requiere de un dispositivo más pequeño que para la verificación de toda la mano, pero aún así es mucho más grande que

algunos dispositivos usados por otros métodos biométricos como el de huellas digitales, facial o de la voz.

### Iris

El iris es la región angular entre el límite de la pupila y la esclera (parte blanca del ojo). La textura visual del iris es formada durante el desarrollo fetal y se estabiliza en los dos primeros años de vida. La compleja estructura del iris posee información distintiva útil para el reconocimiento de una persona. La exactitud y velocidad de los sistemas basados en el reconocimiento de iris son prometedores al grado de que los sistemas de identificación a gran escala se basan en la información del iris. Cada iris es distintivo al igual que las huellas digitales, incluso entre gemelos idénticos y entre cada iris de una misma persona. Es extremadamente difícil alterar la estructura del iris quirúrgicamente y es bastante fácil detectar iris artificiales como los lentes de contacto, sin embargo requiere de la participación del usuario y un alto costo por el sistema.

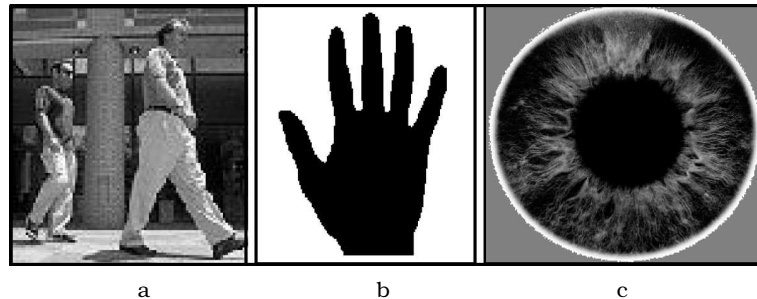


FIGURA 2.7: Ejemplos de algunos factores biométricos presentados por [26]: a)Manera de caminar. b)Iris. c)Huella de la palma

### Manera de teclear

Se supone que cada persona tiene su propia manera de teclear, este método biométrico conductual no es considerado único en cada individuo pero ofrece información suficientemente discriminatoria que permite la verificación de identidad. La manera de teclear de una persona puede ser monitoreada sin molestarla haciendo que teclee alguna información.

### Olor

Es conocido que cada objeto emite un olor, el cual tiene una composición química característica y puede ser usada para distinguir varios objetos. Un componente del olor emitido por el cuerpo humano es distintivo de cada individuo, sin embargo no es claro por la variedad que puede presentar debido al olor de los desodorantes o a las variantes de la composición química del medio ambiente.

### Huella de la palma

Las palmas de la mano humana contiene patrones de crestas y valles muy parecidas a las de huella digital. El área de la palma es mucho más grande que a la de un dedo y como resultado se esperan más rasgos distintivos que en las huellas digitales. Debido a que el reconocimiento de palmas requiere de un dispositivo más grande que para el de huellas digitales el costo es más elevado, sin embargo la palma cuenta con información adicional como las



lineas principales y las arrugas que pueden permitir el escaneo por dispositivos de más baja resolución y más económicos.

### Escaneo retinal

La estructura de la retina se supone que es una característica de cada individuo y cada ojo. Se afirma que es uno de los métodos biométricos más seguros debido a que no es fácil de cambiar o replicar. La adquisición de la imagen requiere de la cooperación de la persona y se requiere el contacto directo lo cual afectan la aceptación de este método. La estructura de la retina puede revelar algunas condiciones médicas como hipertensión, lo cual sería otro factor para apoyar la aceptación del escáneo retinal.

### Firma

La manera de firmar de una persona es conocida como una característica individual. Aunque la firma requiere del contacto con el instrumento para escribir y un esfuerzo por parte del usuario, ha sido aceptado en el gobierno, en transacciones comerciales y legales como un método de verificación. Las firmas son un método biométrico conductual inseguro debido a que cambian en un periodo de tiempo influido por condiciones físicas y emocionales, algunas personas varían sustancialmente y un falsificador profesional puede reproducirla sin mayores dificultades.

### Voz

La voz es una combinación de factores biométricos conductuales y fisiológicos. Los rasgos de la voz de un individuo están basados en la forma y el tamaño de las cuerdas vocales, boca, cavidades nasales y labios, que son usados para la síntesis del sonido. Estas características fisiológicas permanecen invariantes en cada individuo, pero la parte conductual provoca cambios en la forma de hablar de la persona debidos a la edad, condiciones médicas, estado emocional, etc. Una desventaja del reconocimiento de voz es que los rasgos del habla son sensibles a factores de ruido en el medio [26].

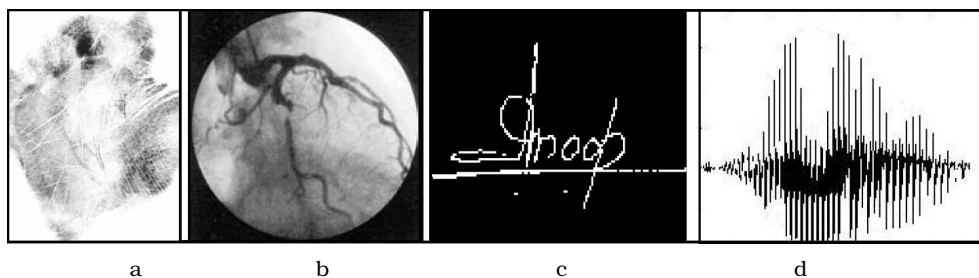


FIGURA 2.8: Ejemplos de algunos factores biométricos presentados por [26]: a)Huella de la palma. b)Retina. c)Firma. d)Voz.

Una comparación resumida de los factores biométricos se presenta en la tabla 2.1 basada en siete factores. La aplicación de alguna técnica de reconocimiento biométrico dependerá de los requerimientos de la aplicación.

TABLA 2.1: Comparación de varias tecnologías biométricas presentada en [26] basada en la percepción de los autores. A, M y B denotan alto, mediano y bajo respectivamente.

Identificador biométrico	Universalidad	Distintividad	Durabilidad	Coleccionabilidad	Desempeño	Aceptabilidad	Engaño
ADN	A	A	A	B	A	B	B
Oído	M	M	A	M	M	A	M
Rostro	A	B	M	A	B	A	A
Termogramas faciales	A	A	B	A	M	A	B
Huellas digitales	M	A	A	M	A	M	M
Forma de caminar	M	B	B	A	B	A	M
Geometría de la mano	M	M	M	A	M	M	M
Geometría de las venas	M	M	M	M	M	M	B
Iris	A	A	A	M	A	B	B
Manera de teclear	B	B	B	M	B	M	M
Huella de la palma	M	A	A	M	A	M	M
Tretina	A	A	M	B	A	B	B
Firma	B	B	B	A	B	A	A
Voz	M	B	B	M	B	A	A

## 2.5. Aplicaciones de los sistemas biométricos

Las aplicaciones biométricas pueden ser divididas en tres grupos principales:

- **Aplicaciones civiles o comerciales** como en la entrada a redes computacionales, seguridad de datos electrónicos, comercio electrónico, acceso a internet, control acceso físico, teléfonos celulares, archivos médicos, enseñanza a distancia, etc.
- **Aplicaciones de alta seguridad** como tarjetas de identificación, licencias de manejo, seguro social, control de pasaportes, control de acceso, etc.
- **Aplicaciones forenses** como la identificación de cadáveres, investigación criminalística, identificación terrorista, determinación de paternidad, búsqueda de niños, etc.

Hasta hace unos años, muchas de las aplicaciones biométricas sólo eran parte de la ciencia ficción, sin embargo en la actualidad los sistemas biométricos han empezado a crecer en el desarrollo de sus aplicaciones con civiles (Ver figura 2.9) [26]. Un ejemplo es el aeropuerto Schiphol de Amsterdam, el cual se encuentra utilizando con éxito un sistema de seguridad biométrico basado en la identificación de iris para acelerar los procedimientos de control de visa y pasaporte. El proyecto es llamado Privium, los pasajeros introducen primero sus datos personales, se les toma una fotografía del iris y compara la información de la imagen con la de una tarjeta de acceso que contiene el registro del pasajero [5].



FIGURA 2.9: Ejemplos de aplicaciones biométricas presentados por [26]: a) Sistema de verificación de huellas digitales desarrollado por Digital Persona, es usado para el acceso a computadoras y redes. b) Puntos de venta basados en verificación de huellas digitales desarrollado por Indivos, se emplea para el pago en tiendas, restaurantes y cafeterías. c) Cerraduras basadas en verificación de huellas digitales desarrollado por BioThentica Corporation, usado en accesos restringidos. d) Sistema acelerado del Servicio de Inmigración y Naturalización (INSPASS), instalado en los aeropuertos de Estados Unidos basado en la verificación de la geometría de la mano y desarrollado por Recognition System. e) El sistema FacePass de Viisage es usado para la verificación en puntos de venta. f) Aeropuerto Ben Gurion en Tel Aviv emplean la geometría de la mano para sistemas de seguridad e inmigración.

## 2.6. Ventajas y desventajas de los sistemas biométricos

Para analizar las ventajas y desventajas de los sistemas biométricos se examinarán en dos grupos, el reconocimiento positivo de las aplicaciones comerciales que pueden operar en modo de identificación o verificación y el reconocimiento negativo en aplicaciones de gobierno y forenses que requieren de identificación.

### 2.6.1. Reconocimiento positivo en las aplicaciones comerciales

Las tecnologías tradicionales logran el reconocimiento positivo por métodos basados en el conocimiento de números de identificación personal o contraseñas, o basados en la posesión de algún objeto como una llave o credencial.

La mayoría de las personas emplea contraseñas basadas en palabras o dígitos que son fáciles de recordar, como nombres, cumpleaños, películas favoritas, el nombre de su mascota; sin embargo este tipo de contraseñas son fáciles de descifrar, mientras que las llaves o credenciales pueden ser duplicadas, perdidas o robadas. Los factores biométricos no pueden ser perdidos u olvidados, requieren que la persona se encuentre presente al momento en que el sistema realice el reconocimiento y son muy convenientes para quienes se les dificulta recordar contraseñas muy largas y/o complicadas.

La oportunidad de un ataque a un sistema biométrico dependerá de la exactitud del mismo,

para poder engañarlo se requerirá de un gran número de muestras del biométrico lo cual es mucho más difícil que generar números de identificación personal o contraseñas.

Sin embargo para las aplicaciones comerciales se requiere realizar un análisis costo-beneficio al instalar un sistema basado en un factor biométrico, debido a que el costo de instalación y mantenimiento es mucho más elevado que el de los métodos convencionales. Algunos de los métodos biométricos son considerados como invasivos y la mayor parte de la población los rechaza o les incómoda su uso.

### 2.6.2. Reconocimiento negativo en aplicaciones de gobierno y forenses

Las aplicaciones de reconocimiento negativo son empleadas para controlar y prevenir el acceso de terroristas a bordo de aeroplanos o trenes, para la identificación de cadáveres y criminales. La implementación de estos sistemas, al igual que en el reconocimiento positivo, requieren de un análisis costo-beneficio, sin embargo al final de la identificación un experto en el área es quien toma la decisión final.

El uso de algún identificador biométrico en este tipo de aplicaciones no infringe con las libertades civiles a menos de que los datos personales sean introducidos en una base de datos criminalística, por lo que se necesita una apropiada legislación para proteger el abuso en los sistemas biométricos.

### 2.6.3. Limitaciones de los sistemas biométricos unimodales

Un sistema biométrico unimodal es aquel que emplea un biométrico para la identificación o verificación de un individuo. El éxito de la instalación de un sistema biométrico unimodal no implica que todos los problemas se hayan resuelto, debido a que aún cuenta con limitantes como:

- *Ruido en el dato sensado:* El ruido es el resultado del mantenimiento inapropiado del sensor, de la incorrecta interacción con el sensor o por las condiciones ambientales desfavorables (Ver figura 2.10).



FIGURA 2.10: Efectos del ruido en las imágenes de un sistema biométrico, la figura de la izquierda muestra una huella digital obtenida durante el registro y la figura de la derecha muestra el resultado de la imagen de la verificación después de tres meses. El desarrollo de cicatrices o cortadas también pueden resultar en un erróneo reconocimiento de las huellas digitales [26].

- *Variaciones de un rasgo biométrico:* El dato biométrico adquirido de forma individual durante la autenticación es muy diferente al dato usado para generar la plantilla al momento del registro, afectando la etapa de comparación y decisión. Estas variaciones se deben comúnmente a la incorrecta interacción con el sensor, cuando las características han sido modificadas o ha sido reemplazado.



FIGURA 2.11: Un ejemplo de una mala interacción con el sensor, en la figura obtenida de [26] se muestran cuatro impresiones de la huella digital de un individuo que no tenía mucha tinta en el dedo para plasmar las grietas principales de su huella digital.

- *No universalidad:* Cuando la característica biométrica no puede ser adquirida con exactitud debido a que no tiene los rasgos definidos para ser sometido al reconocimiento.
- *Ataques de engaño:* Cuando un impostor intenta engañar al sistema aprovechándose de la vulnerabilidad del identificador biométrico, como el reconocimiento por medio de la firma o voz [26].

## 2.7. Falsificación de un factor biométrico

La falsificación de un sistema biométrico implica un costo que varía dependiendo del método se desee falsificar, uno de los más fáciles de replicar son las huellas digitales con ayuda de algún material maleable (Ver figura 2.12) por lo que se introduce el concepto de sistemas biométricos multimodales, es decir, se emplean más de dos identificadores biométricos para la identificación de un individuo.

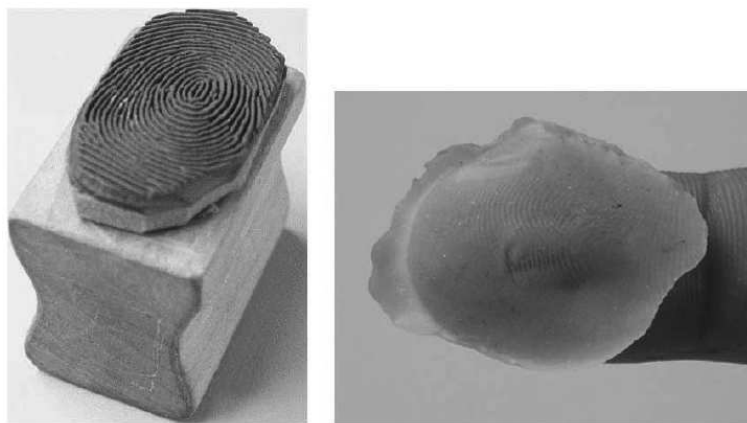


FIGURA 2.12: Falsificación de huellas digitales con consentimiento de los usuarios. La impresión de la huella puede ser duplicada en materiales muy maleables [26].

En la tabla 2.2 se presenta un condensado sobre los costos por la falsificación de algunos factores biométricos. Para poder realizar la falsificación requiere de la información que le proporciona el factor biométrico, hay algunos que son fáciles de conseguir y otros muy complicados si no se tiene cooperación por parte del usuario.

TABLA 2.2: Comparación sobre el costo de falsificar algún factor biométrico presentado en [15].

<b>Factor biométrico</b>	<b>Sensor o método</b>	<b>Costo de la falsificación</b>
<b>Huella digital</b>	Óptico, termal, capacitivo	Muy baja
<b>Rostro</b>	Óptico, infrarojo	Baja
<b>Iris</b>	Óptico, infrarojo	Media o alta
<b>ADN</b>	Análisis químico	Alta
<b>Retina</b>	Infrarojo	Alta a muy alta

Para lograr la falsificación del iris se requiere de tener una imagen de alta calidad con iluminación en diferentes direcciones gracias a que el iris es una estructura de tres dimensiones. Por consiguiente la adquisición de una copia se lograría con un sensor óptico como una cámara-CCD (Coupling Charge Device) simulando las condiciones ópticas apropiadas, como el acercamiento o la distancia [15].



## El iris como factor biométrico

---

La palabra iris proviene del término *ἶρις* que significa arcoiris. Debido a todas las características mostradas en la figura 1.1 la estructura del iris es única, contiene muchas capas que hacen de este órgano una estructura de tres dimensiones. [44] menciona que la estructura del iris es estable con la edad del individuo, es decir que no se ve modificado conforme aumenta la edad como ocurre con las huellas digitales o la voz. El iris está compuesto por un tejido elástico que hace del iris una estructura dinámica, el diámetro de la pupila oscila dependiendo de la luz que reciba.

Una ventaja que presenta el iris sobre otros biométricos es que se encuentra protegido por otras dos capas oculares, como se puede ver en la figura 3.1, se encuentran la córnea y el humor acuoso; el iris constituye la porción más anterior de la capa vascular del ojo [33].

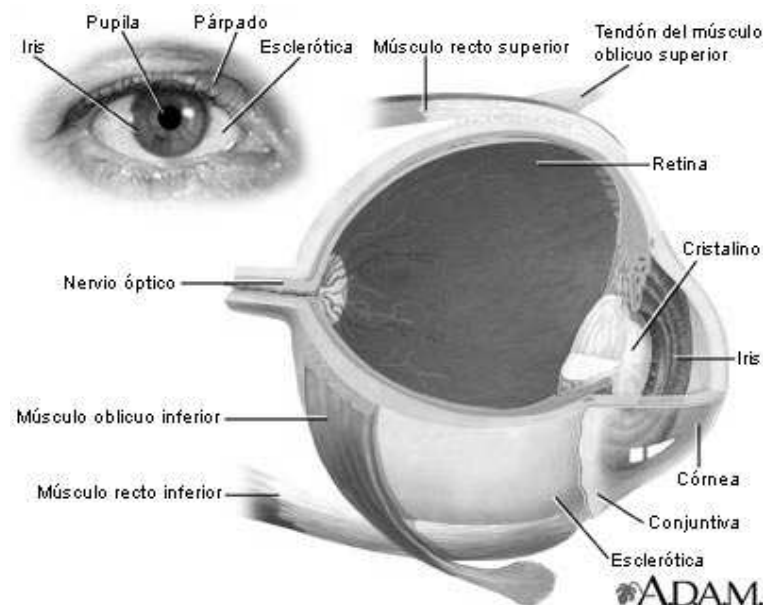


FIGURA 3.1: Imagen obtenida de [33] que permite identificar la ubicación del iris en el ojo humano.

La morfología del iris es completada durante los primeros ocho meses de gestación, durante el primer año de vida las células cromáticas aún provocan cambios de coloración del iris, pero existe evidencia médica que indica la estabilidad de su estructura a lo largo de la vida [8].



El reconocimiento de iris desde el punto de vista de su potencial como aplicación en un sistema biométrico se basa en:

- Estabilidad frente a cambios originados por accidentes, debido a la protección que le confiere la córnea.
- Fácil mecanismo de detección de sujeto vivo. Pequeñas variaciones en la iluminación producen alteraciones en la apertura de la pupila. Incluso, con iluminación fija, el iris no se encuentra estático, sino que presenta pequeñas y continuas variaciones en su apertura.
- La imagen se pueden capturar de forma no invasiva, al ser visible desde el exterior por la transparencia de la córnea.
- El intento de falsificar el iris de una persona conllevaría operaciones quirúrgicas que podrían dañar muy seriamente la visión.

Se puede concluir que esta técnica presenta una unicidad extremadamente alta, comodidad para el usuario, dificultades de falsificación sin la cooperación del sujeto, garantizando su viabilidad para ser utilizada en identificación en entornos de alta seguridad.

### 3.1. Técnicas publicadas

La primera idea para utilizar el iris como patrón de reconocimiento de personas fue propuesto en 1936 por el oftalmólogo Frank Burch. Sin embargo, no fue hasta la década de los 80, cuando empezó a ser conocida en forma de diversas películas de ficción; fue hasta 1987, cuando Leonard Flom y Aran Safir, oftalmólogos americanos, patentaron el concepto de Burch. Su interés en desarrollar el sistema, les empujó a contactar con John G. Daugman, para que éste desarrollase los algoritmos necesarios para realizar el reconocimiento biométrico a través del patrón del iris. Estos algoritmos, patentados por Daugman en 1994 [9], y publicados en parte en [8] y [11], son la base de todos los sistemas de reconocimiento por iris existentes [40].

[42] y [15] mencionan que los algoritmos de reconocimiento de iris más reconocidos son:

#### 1. Daugman

Emplea un operador integro-diferencial para detectar el centro y el radio del iris y la pupila para realizar una conversión de coordenadas polares a rectangulares que es el área que requiere para el análisis. Para extraer las características se emplea una modificación de los filtros de Gabor. Por último en la etapa de comparación es calculado un operador booleano (OR-exclusivo) para obtener la distancia de Hamming, donde la distancia cero equivale a comparar dos iris idénticos [11].

#### 2. Wildes, et al

Cuenta con un prototipo patentado que captura imágenes de alta calidad y no distorsionadas, teniendo como principal característica una iluminación que es cómoda para el usuario. En la etapa de localización del iris y la pupila emplea la transformada de Hough sobre una imagen binarizada. La representación se realiza mediante filtros isotrópicos derivados de una aplicación de los filtros Laplacianos de la Gaussiana de la imagen. Por último al comparación es llevada a cabo por una correlación normalizada sobre bloques de 8x8 en bandas de frecuencias espaciales de la pirámide laplaciana. Este algoritmo tiene la desventaja de que sólo se ha empleado en modo verificación [44].

### 3. Boles y Boashash

En su etapa de localización y normalización usan algoritmos de detección de orillas, con la transformada wavelet de cero intersecciones son calculados varios niveles de resolución sobre círculos concéntricos en el iris. Tienen como resultado una señal de una dimensión que representa las características del iris, esta señal es modelada para ser almacenada en una base de datos; si llega un iris al sistema se obtendrá el modelo y será comparado con las representaciones almacenadas realizar su comparación [4].

### 4. Zhu, et al.

Para la localización del iris y la pupila emplean dos círculos no concéntricos maximizando los cambios de un perímetro normalizado, sumando la escala de grises a lo largo del círculo. La extracción de características la hacen por medio de filtros de Gabor de multiresolución combinados con la transformada wavelet 2D. Por último la clasificación emplea la distancia de Hamming ponderada [46].

### 5. Li Ma, et al.

La etapa de localización la efectúa en dos fases, en la primera dentro de una zona rectangular para localizar el iris y la segunda fase implementando la transformada de Hough para localizar el iris y la pupila. Emplea filtros circulares simétricos para la extracción de características capturando la información en una banda de frecuencia específica. En la comparación utiliza un clasificador basado en el método de la línea característica más cercana [28].

### 6. Avila, et al.

Trabaja con imágenes de alta resolución y en escala de grises realizando una exploración de la imagen incrementando el radio y el ángulo. Como principal característica es la extracción de características por medio de la transformada wavelet dyadic discreta. La comparación de las señales se lleva a cabo por tres métodos, la distancia Euclidiana, la distancia de Hamming y la distancia directamente relacionada con las intersecciones nulas [29].

### 7. Tisse, et al.

Para la localización del iris y la pupila emplea una combinación de la transformada de Hough y el operador integro-diferencial. La extracción de características emplea la transformada de Hilbert 2D generando un código similar al de Daugman que le permite emplear la distancia de Hamming en la fase de comparación [41].

De acuerdo a [42] se realizó la implementación de cuatro algoritmos de reconocimiento de iris en MATLAB 6.5®. Dichos algoritmos fueron probados con la base de datos de imágenes de iris de CASIA [34], la cual se encuentra disponible para el dominio público. La base de datos incluye 756 imágenes de iris de 108 individuos. Por cada ojo, siete imágenes fueron capturadas en dos sesiones, en la primera se tomaron tres muestras y cuatro en la segunda. Para realizar las comparaciones de los algoritmos las primeras tres muestras fueron tomadas para el entrenamiento y las últimas cuatro para las pruebas. El desempeño de los resultados se basaron en los errores de Proporción de Falsos Positivos (FPF) y en la Proporción de Falsos Negativos (PFN), el resultado de las implementaciones se encuentra en la tabla 3.1 donde se muestra por mucho el superior desempeño del algoritmo de Daugman.

TABLA 3.1: Desempeño de los algoritmos.

Algoritmo	Técnica	PFP/PFN (%)	Exactitud (%)
Avila	Transformada wavelet dyadic	0.03/2.08	97.89
Li Ma	Filtros simétricos circulares	0.02/1.98	98.00
Tisse	Análisis frecuencial	1.84/8.79	89.37
Daugman	Filtros de gabor	0.01/0.09	99.90

## 3.2. Algoritmo de Daugman

### 3.2.1. Búsqueda del iris en la imagen

#### Adquisición de la imagen del iris

La imagen debe capturar todos los patrones del iris, donde el sistema de resolución debe tener como mínimo 70 píxeles en el radio del iris. Las cámaras comúnmente usadas para capturar la imagen del ojo son monocromáticas CCD (Coupling Charge Device) debido a que emplean una iluminación NIR (near-infrared) con una banda de 700-900 nm, la cual es invisible para el ojo humano.

Durante el proceso una serie de imágenes de suficiente calidad son adquiridas y sólo una de ellas para al siguiente proceso por medio de una prueba para ver si soporta el reconocimiento. Los factores que dificultan el enfoque de la imagen son las restricciones de iluminación, el movimiento del individuo, la distancia y el tamaño.

Para evaluar la calidad de imagen se apoya en los bordes de alto contraste entre la frontera del iris y la esclera. Emplea como principal herramienta para evaluar el enfoque de la imagen la transformada de Fourier 2-D. Una imagen puede ser representada en el plano real  $I(x, y)$  como:

$$F(\mu, \nu) = \frac{1}{(2\pi)^2} \int \int I(x, y) e^{-i(\mu x + \nu y)} dx dy \quad (3.1)$$

El desenfoque de una imagen es representado normalmente por una convolución de una imagen bien enfocada y la función de la ecuación (3.1), el resultado es modelado como una Gaussiana cuyo espacio constante representa el grado de desenfoque [11].

#### Extracción del iris

El sistema de Daugman emplea tres parámetros para obtener los límites límbico y pupilar. Aprovechando que el iris y la pupila presentan un contorno circular, éstos pueden ser parametrizados por la localización de su centro  $(x_c, y_c)$  y su radio  $r$ .

El primer paso es localizar el límite pupilar con los cambios abruptos del brillo en la imagen del iris, el cambio será detectado por la exploración de una serie de círculos; el contorno circular se ajusta mediante la maximización del siguiente operador integro-diferencial:

$$(r, x_0, y_0) = \underset{max}{\left| \frac{\partial}{\partial r} G(r) * \oint_{r, x_c, y_c} \frac{I(x, y)}{2\pi r} ds \right|} \quad (3.2)$$

donde  $G(r) = (1/\sqrt{2\pi}\sigma)e^{-(r-r_0)^2/2\sigma^2}$  es un radio Gaussiano con centro en  $r_0$  y desviación estándar  $\sigma$  que uniformiza la imagen a la escala espacial seleccionada, el símbolo  $*$  denota convolución,  $ds$  un arco circular del elemento,  $I(x, y)$  representa la imagen contenida en las coordenadas  $(x, y)$  y la división por  $2\pi r$  sirve para la normalización de la integral [44].

Para localizar el límite límbico se realizan algunas modificaciones de la ecuación (3.2), para su localización sólo se exploran dos arcos de  $45^\circ$  grados cada uno, estos dos arcos van a ubicarse sobre la horizontal del iris uno a  $0^\circ$  y el otro a  $180^\circ$ . Matemáticamente el operador implementado para buscar el radio de la pupila es mediante la maximización de:

$$\max_{r \in [1, 5r_0, 10r_0]} \frac{\partial}{\partial r} \int_{\rho=r-\delta}^{r+\delta} \frac{2}{\pi\delta r} \int_{\theta=\phi-\pi/8}^{\phi+\pi/8} I(\rho, \theta) \rho d\rho d\theta \quad (3.3)$$

donde  $r_0$  es el radio de la pupila,  $\delta$  es el incremento fino del radio (típicamente de  $0,1 r_0$ ),  $I(\rho, \theta)$  es la imagen expresada en coordenadas polares  $\rho$  y  $\theta$  y  $\phi$  es igual a  $0$  o  $\pi$  que corresponde a la ubicación de los arcos en la circunferencia del iris [9].

El operador de la ecuación (3.2) y (3.3) sirve para encontrar los límites de la pupila y el iris como lo muestra la figura 3.2.

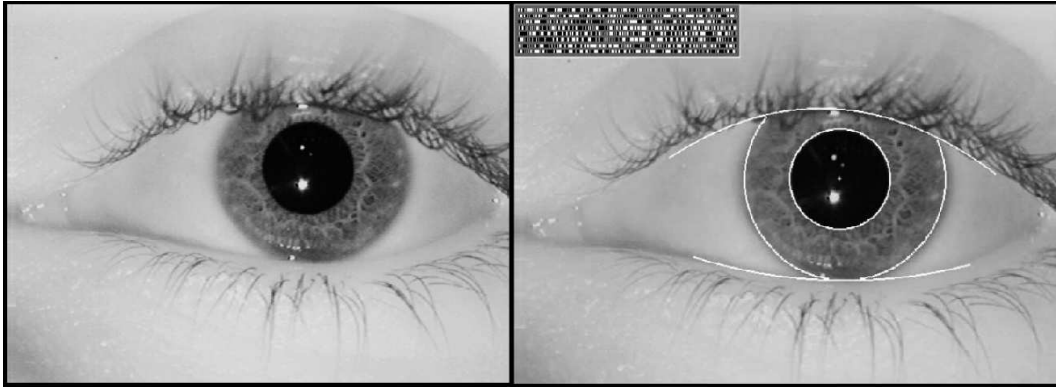


FIGURA 3.2: Ejemplo de una imagen capturada a una distancia de 30 cm. Las líneas muestran los resultados de la localización de los límites de la pupila y el limbo [8].

### 3.2.2. Decodificación del Iris por Demodulación de Wavelet 2-D

Cada imagen es demodulada para extraer las características usando filtros de Gabor 2D, el proceso de codificación es mostrado en la figura 3.3. Para determinar el cuadrante en el plano complejo de cada fasor resultante del área del iris proyectada sobre valores complejos del filtro de Gabor 2D se utiliza la siguiente ecuación:

$$h_{\{Re, Im\}} = \text{sgn}_{\{Re, Im\}} \iint_{\rho} \iint_{\phi} I(\rho, \phi) e^{-i\omega(\theta_0-\phi)} e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_0-\phi)^2/\beta^2} \rho d\rho d\phi \quad (3.4)$$

donde  $h_{\{Re, Im\}}$  puede ser considerado como un bit de valores complejos donde la parte real e imaginaria son 1 ó 0 (sgn) dependiendo del signo de la integral 2D;  $I(\rho, \phi)$  es la imagen del iris después de haber sido corregida por la diferencia de tamaño y translación como será explicado más adelante;  $\alpha$  y  $\beta$  son la multiescala de los parámetros de la transformada de Wavelet 2D,  $\omega$  es la frecuencia de la transformada de Wavelet y  $(r_0, \theta_0)$  representa las coordenadas polares de cada región del iris de cada coordenada del fasor. Las regiones del iris son proyectadas por la ecuación (3.4) sobre el cuadrante de los filtros de Gabor 2D generando valores de coeficientes complejos, donde la parte real e imaginaria son las coordenadas de un fasor en el plano complejo. El ángulo de cada fasor es cuantificado en uno de los cuatro cuadrantes, traducidos

en dos bits de información. El proceso es repetido frente al iris con diversos tamaños de wavelets, frecuencias y orientaciones para extraer 2048 bits.

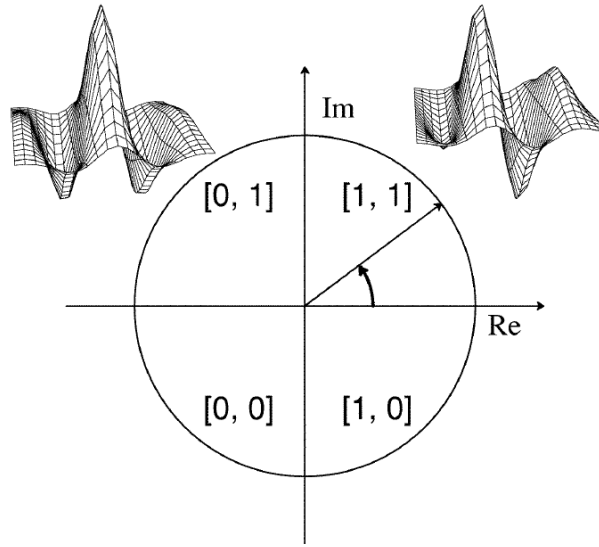


FIGURA 3.3: La fase del proceso de demodulación es usada para codificar las características del iris [11].

La importancia del reconocimiento del iris es la prueba estadística de independencia, que envuelve muchos grados de libertad que permiten garantizar que al comparar dos ojos, la prueba falle. La prueba estadística de independencia es implementada por un simple operador booleano XOR  $\otimes$  aplicado a los 2048 bits del iris de la prueba y el de una base de datos, enmascarado con el operador AND  $\cap$ , que corresponde al vector máscara que previene que las características que no son propias del iris no afecten la comparación. El operador XOR detecta discrepancias entre los pares correspondientes de bits, mientras que el operador AND garantiza que la estimación entre el par de bits no esté siendo corrompida por pestañas, párpados, reflejos o algún otro ruido. Las reglas (||) son el resultado del bit del vector (utilizado para calcular una fracción de la Distancia de Hamming HD) y son empleadas como una medida de desigualdad entre los dos iris comparados; denotados como  $\{codeA, codeB\}$  y los vectores máscara son denotados por  $\{maskA, maskB\}$ :

$$HD = \frac{|(codeA \otimes codeB) \cap maskA \cap maskB|}{|maskA \cap maskB|} \quad (3.5)$$

donde el denominador cuenta el número de agentes que no pertenecen al iris como pestañas o parpados, por lo que la resultante de HD es una medida fraccional donde cero representa una perfecta igualdad entre los pares de iris comparados.

Debido a que las diferencias entre el iris son medidas por 1 ó 0, la proporción de éxito entre los bits de dos iris diferentes es de HD=0.5, por lo tanto la mejor herramienta para estimar la probabilidad de las pruebas es una distribución binomial, donde la fracción está dada por:

$$f(x) = \frac{N!}{m!(N-m)!} p^m (1-p)^{(N-m)} \quad (3.6)$$

donde N es el número de grados de libertad dados por  $N = p(1-p)/\sigma^2$ , donde  $p = 0,5$ ,  $\sigma$  es la desviación estándar de la base de datos de donde provenga la imagen del iris y  $x = m/N$  es la fracción del número de éxitos, en este caso la Distancia de Hamming [11].

La figura 3.4 es conocida como al curva impostor-genuino, la curva del genuino que se obtiene la frecuencia de la distancia de Hamming entre pares de iris iguales  $P_{Ge}(H)$  y la del impostor con la frecuencia de la distancia de Hamming entre pares de iris diferentes  $P_{Im}(H)$ .

Con la información de las curvas pueden obtenerse cuatro probabilidades, que un genuino sea aceptado (GA), que un genuino sea rechazado (GR), que un impostor sea aceptado (IA) o que un impostor sea rechazado (IR). Si las reglas de decisión son:

- Aceptado si la Distancia de Hamming < Umbral
- Rechazado si la Distancia de Hamming > Umbral

donde las probabilidades son definidas por la densidades de la curvas de genuino e impostor en función de:

$$P(GA) = \int_0^U P_{Ge}(H)dH \quad (3.7)$$

$$P(GR) = \int_U^1 P_{Ge}(H)dH \quad (3.8)$$

$$P(IA) = \int_0^U P_{Im}(H)dH \quad (3.9)$$

$$P(IR) = \int_U^1 P_{Im}(H)dH \quad (3.10)$$

Estas cuatro probabilidades tiene significado para las cuatro áreas mostradas en la figura 3.4. Las probabilidades unidas en pares suman una unidad y los otros pares están gobernadas por las siguientes desigualdades [9]:

$$P(GA) + P(GR) = 1 \quad (3.11)$$

$$P(IA) + P(IR) = 1 \quad (3.12)$$

$$P(GA) > P(GR) \quad (3.13)$$

$$P(IA) > P(IR) \quad (3.14)$$

$$(3.15)$$

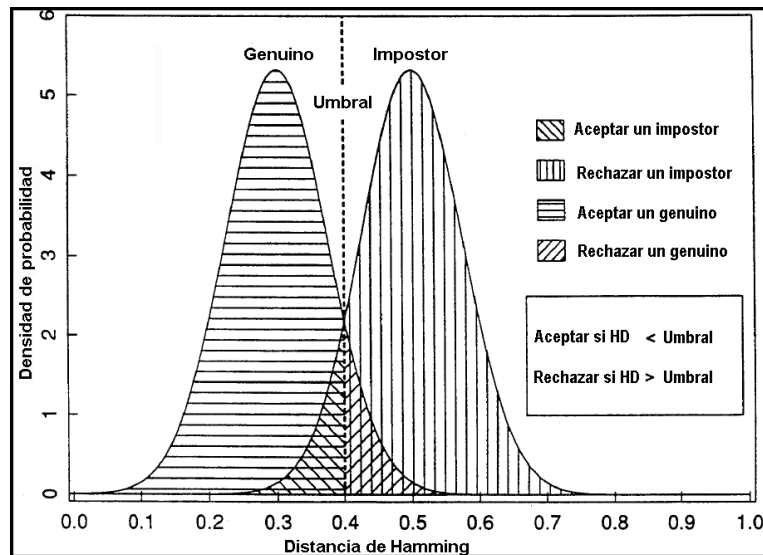


FIGURA 3.4: Comparaciones por medio de la distancia de Hamming [9].

### 3.2.3. Reconocimiento de iris sin importar tamaño, posición y orientación

Es necesario solucionar problemas por cambios de tamaño, posición y orientación debido a múltiples factores como la distancia a la que fue tomada la imagen, la iluminación, la rotación del ojo, la posición de la cámara, entre otros.

Para la rotación de la imagen se emplea un sistema coordenado pseudo-polar, que puede ser descrito por dos dimensiones por la variable polar  $r$  en un intervalo unitario  $[0,1]$  y por el ángulo  $\theta$  en un intervalo de  $[0,2\pi]$ . La imagen del iris es remapeada de  $I(x,y)$  manejando coordenadas cartesianas  $(x,y)$  a un sistema de coordenadas polar no concéntrico  $(r,\theta)$  y pueden ser representadas como:

$$I(x(r,\theta), y(r,\theta)) - I(r,\theta) \quad (3.16)$$

donde  $x(r,\theta)$  y  $y(r,\theta)$  se definen como la combinación lineal del límite de la pupila  $(x_p(\theta), y_p(\theta))$  y el límite del limbo  $(x_s(\theta), y_s(\theta))$ , ambas son detectadas para encontrar el máximo del operador (3.2) como:

$$x(r,\theta) = (1-r)x_p(\theta) + rx_s(\theta) \quad (3.17)$$

$$y(r,\theta) = (1-r)y_p(\theta) + ry_s(\theta) \quad (3.18)$$

Desde que se emplea un rango en las coordenadas polares para localizar el límite de la pupila y el iris son corregidos los problemas debido a la diferencia de cambios de tamaño de la pupila.

La localización del iris y el sistema de coordenadas descrito eliminan las variaciones de posición 2D y tamaño del iris. Para eliminar las diferencias por la orientación del iris se emplea la siguiente ecuación:

$$f_n(x) = nf_0(x)[1 - F_0(x)]^{n-1} \quad (3.19)$$

donde  $f_n(x)$  está asociada a las  $n$  diferentes orientaciones relativas para encontrar la mejor  $n$  que cumpla con los criterios de la prueba,  $f_0(x)$  es la binomial definida por la ecuación 3.6 y  $F_0(x)$  es la acumulada de  $f_0(x)$  de 0 a  $x$  [11].

## 3.3. Algoritmo de Avila

La idea principal de esta técnica es representar las características del iris por aproximaciones con diferentes niveles de resolución de grises basados en la representación de la transformada Wavelet dyadic con intersecciones nulas.

### 3.3.1. Adquisición de la imagen y preprocesamiento

La imagen es capturada usando una cámara de alta resolución con luz infrarroja. El primer paso es convertir la imagen en escala de grises y es creado su histograma. Entonces la imagen es delimitada, primero se localiza el centro del iris y el borde entre el iris y la esclera. La detección se lleva a cabo por medio de la maximización de  $D$  en la ecuación que se muestra en 3.20, donde  $(x_0, y_0)$  es el centro del iris,  $\Delta_r$  y  $\Delta_\theta$  son los incrementos del radio y del ángulo, y  $I(x,y)$  son los niveles de grises de la imagen.

$$D = \sum_m \sum_{k=1}^5 (I_{n,m} - I_{n-k,m}) \quad (3.20)$$

donde

$$I_{i,j} = I(x_0 + i\Delta_r \cos(j\Delta_\theta), y_0 + i\Delta_r \sin(j\Delta_\theta)) \quad (3.21)$$

Una vez detectados los límites del iris, a cada imagen se le suprime lo que se encuentre fuera de éstos. Se realiza el mismo procedimiento para encontrar los límites de la pupila y se suprime todo lo que se encuentra en fuera de esa área.

Las imágenes son escaladas para tener el mismo diámetro de la imagen original. Se escoge el centro de la pupila como punto de referencia para la extracción de características, los niveles de grises en el contorno del círculo virtual son registrado como un conjunto de datos de 256 bits, de esta manera son referenciadas como un indicador del iris.

### 3.3.2. Extracción de características

Este paso lo lleva a cabo mediante la técnica de la transformada de wavelet dyadic discreta. El término “wavelet” es usado para cualquier función de  $L^2(\mathfrak{R})$  y la transformada wavelet de una función  $f$  puede ser vista como la respuesta de una familia de filtros paso-bandas. Un problema fundamental es entender que la representación por intersecciones nulas define una representación completa y estable de la señal original, por lo que el procedimiento puede emplearse para reconocimiento de patrones.

Si  $f \in L^2(\mathfrak{R})$  y si  $\{W_{2^j} f(x)\}_{j \in \mathbb{Z}}$  en la transformada wavelet dyadic, la posición de intersecciones nulas puede representarse como una función constante:

$$Z_{2^j} f = \frac{\int_{z_{n-1}}^{z_n} W_{2^j} f dx}{z_n - z_{n-1}} \quad (3.22)$$

En las implementaciones prácticas, la señal de entrada, en este caso el indicador del iris, es medida con una resolución finita que impone una escala fina para computar la transformada de wavelet dyadic. Para cuestiones de normalización se supone que la escala más fina es igual a uno y la escala más grande es de  $2^j$ , por lo que se puede obtener la transformada wavelet dyadic discreta del indicador de un iris (IS) con la siguiente expresión:

$$\{S_{2^j}(IS), (W_{2^j}(IS))_{1 \leq j \leq J}\} \quad (3.23)$$

donde  $S_{2^j}(IS)$  es la señal ordinaria y  $(W_{2^j}(IS))_{1 \leq j \leq J}$  puede ser interpretada como los detalles discretos aprovechables cuando se suaviza el indicador del iris a la escala de uno, pero que desaparecen cuando el indicador del iris se suaviza a la escala más grande de  $2^j$ .

Las intersecciones nulas de  $(W_{2^j}(IS))_{1 \leq j \leq J}$  son estimadas con los cambios de signo de sus muestras. La posición de cada intersección nula es estimada mediante una interpolación lineal entre dos muestras de diferente signo. La figura 3.5 es la gráfica que muestra los resultados en diferentes niveles resoluciones.



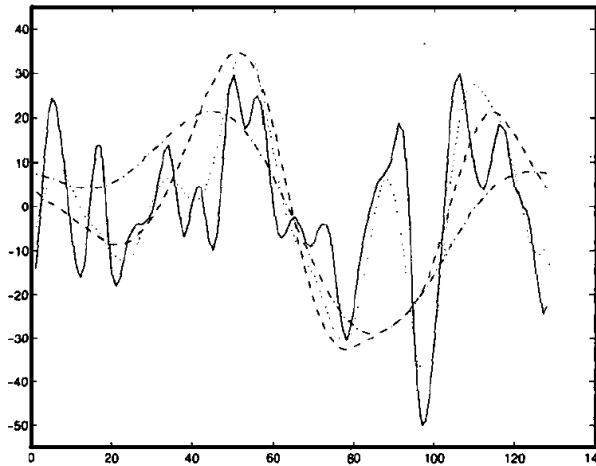


FIGURA 3.5: Niveles de resoluciones donde  $3 \leq j \leq 6$ , con la técnica de la transformada de wavelet dyadic [29].

### 3.3.3. Clasificación y verificación

Todas las características obtenidas son empleadas en el proceso de comparación, de acuerdo con [42] en esta etapa se usaron tres algoritmos de comparación:

- *Distancia Euclidiana*: Considerada como una de las técnicas más comunes y se basa principalmente en la siguiente ecuación:

$$d_E(y, p) = \sqrt{\sum_{i=1}^L (y_i - p_i)^2} \quad (3.24)$$

donde  $L$  es la dimensión del vector característico,  $y_i$  son las componentes del vector característico de la muestra y  $p_i$  son las componentes del vector característico del patrón.

- *Distancia de Hamming*: Emplea la ecuación (3.5) explicada en el algoritmo de Daugman.
- *Distancia relacionada directamente con las intersecciones nulas*: se emplean los valores de los conjuntos desiguales de la muestra y el patrón, cuya resolución se encuentra en el intervalo  $[K, M]$  donde el promedio de desigualdades es calculado como [29]:

$$d_Z(y, p) = \sum_{j=k}^M \frac{d_j(y, p)}{K - M - 1} \quad (3.25)$$

## 3.4. Algoritmo de Tisse

Este algoritmo implementa en el sistema de reconocimiento el gradiente de descomposición de la transformada de Hough en combinación con el operador integro-diferencial y emplea la transformada de Hilbert 2D para la extracción de información de la textura del iris.

### 3.4.1. Localización del iris

El algoritmo de Daugman emplea un operador Integro-diferencial (ecuación 3.2) para detectar el centro de la pupila y el diámetro del iris, sin embargo el operador es sensible al reflejo de la luz en la pupila.

Para contrarrestar este efecto [41] sugiere la combinación del operador integro-diferencial con la transformada de Hough. Consiste primero en el uso de la técnica para el cálculo de orillas para aproximar la posición global del ojo (centro de la pupila) y después aplicar el operador integro-diferencial para tener una localización más precisa de los límites de la pupila. Actualmente la estrategia hace uso de la descomposición en gradientes de la transformada de Hough, que es una variante de la transformada de Hough aplicada en la detección de formas circulares. De la ecuación del círculo  $(x - x_0)^2 - (y - y_0)^2 = r^2$ , donde  $r$  es el radio, se pueden expresar las coordenadas del centro  $(x_0, y_0)$  en función de dos gradientes de primer orden ( $G_x$  para el eje  $x$ ,  $G_y$  para el eje  $y$ ) de la siguiente manera:

$$x_0 = x \pm \frac{r}{\sqrt{1 + \frac{G_y^2}{G_x^2}}} \quad (3.26)$$

$$y_0 = x \pm \frac{r}{\sqrt{1 + \frac{G_x^2}{G_y^2}}} \quad (3.27)$$

### 3.4.2. Transformación de coordenadas rectangulares a polares

Esta fase es igual a la aplicada en el algoritmo de Daugman de acuerdo a las ecuaciones 3.17 y 3.18 que se explicaron anteriormente.

### 3.4.3. Extracción de características

En esta parte del método se introducen dos nuevos conceptos, la fase instantánea y la frecuencia emergente. La fase instantánea es obtenida por la construcción de una señal analítica, que resulta de la combinación de la señal original y su transformada de Hilbert, la cual puede ser formulada como:

$$z_x(t) = x(t) + j.H\{x(t)\} \quad (3.28)$$

donde  $H(x(t))$  es la transformada de Hilbert de  $x(t)$  y la señal  $z_x(t)$  es llamada la señal de fase. La transformada de Hilbert  $H_x$  de la señal original  $x(t)$  puede ser expresada en el dominio de Fourier como:

$$TF[H_x](f) = (-j.\text{signe}(f)).X(f) \quad (3.29)$$

La extensión de la transformada de Hilbert a multidimensional es conocida como la imagen analítica.

Los algoritmos de demodulación de FM son aplicados directamente a las componentes de la función por medio del uso de la imagen analítica  $z(\rho, \theta)$  de la siguiente manera:

$$|\Delta\varphi Di(\rho, \theta)| = \arccos \left[ \frac{z(\rho, \theta + 1) + z(\rho, \theta - 1)}{2z(\rho, \theta)} \right] \quad (3.30)$$

Cualquier componente imaginaria diferente de cero debe ser descartada antes de ser evaluada en la función  $\arccos$ . En el caso de la demodulación de la frecuencia sólo enfocada sobre la información de la horizontal  $\theta$  en la representación rectangular de la textura del iris. La frecuencia dominante del  $\Delta\varphi Di(\rho, \theta)$  es conocida como el vector de la frecuencia emergente. Además se puede obtener la fase instantánea  $\varphi_i(\rho, \theta)$  efectuando la siguiente operación algebraica:

$$\varphi_i(\rho, \theta) = \arctan \frac{\Im(z_i(\rho, \theta))}{\Re(z_i(\rho, \theta))} \quad (3.31)$$

El código del iris se genera de manera similar al sistema de Daugman donde la parte real e imaginaria son obtenidas de las ecuaciones (21 y 22) y la imagen es seccionada en nueve zonas. Para reducir los efectos de los reflejos y los párpados se ajusta el código del iris truncando pequeñas zonas como se muestra en la figura 3.6.

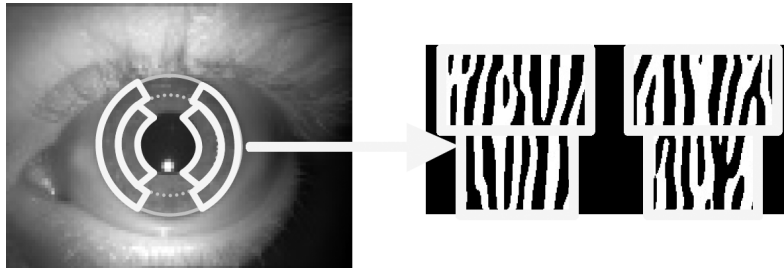


FIGURA 3.6: Generación del código del iris por el método propuesto por [41].

#### 3.4.4. Comparación

Para el caso de la comparación se emplea la distancia de Hamming de manera muy similar al algoritmo de Daugman de acuerdo a la ecuación (3.5) [41].

### 3.5. Algoritmo de Li Ma

La principal característica del algoritmo de Li Ma es el uso de filtros circulares simétricos para la extracción de características.

#### 3.5.1. Preprocesamiento de la imagen

La localización del iris se efectúa en dos pasos, el primero es aproximando la región en una proyección horizontal y vertical; el segundo paso se realiza obteniendo los dos círculos que limitan al iris mediante la transformada de Hough sobre la región que es determinada en el primer paso.

Las imágenes de los iris son capturados en diferentes tamaños y pueden presentar variaciones en la iluminación y algunos otros factores. Para contrarrestar estos problemas se realiza un mapeo a un bloque rectangular ajustando la textura a un determinado tamaño. También es empleado un filtro Gaussiano pasa-bajos para remover los ruidos de altas frecuencias en la imagen obteniendo la imagen, de la figura 3.7.

#### 3.5.2. Extracción de características

En el dominio de la frecuencia se puede extraer mucha información de una imagen usando filtros, como los filtros de Gabor o los filtros circulares simétricos. La diferencia entre estos filtros es la modulación de las funciones senoidales. Un filtro circular simétrico se define de la siguiente manera:

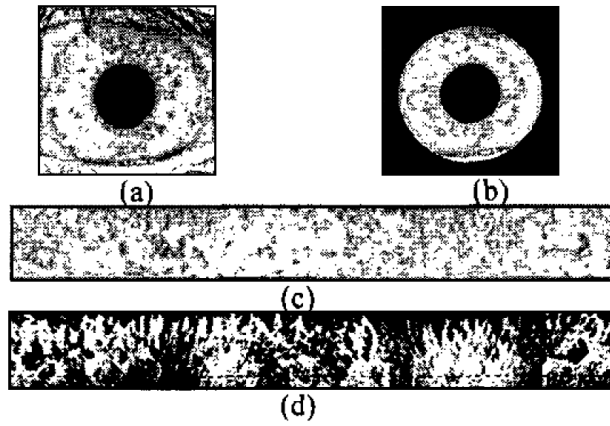


FIGURA 3.7: Preprocesamiento de la imagen por [28]. a)Imagen original. b)Detección de la región del iris. c)Mapeo de la imagen. d)Imagen después de ser eliminado el ruido (usada para extraer las características).

$$G(x, y, f) = \frac{1}{2\pi\delta_x\delta_y} \exp \left[ -\frac{1}{2} \left( \frac{x^2}{\delta_x^2} + \frac{y^2}{\delta_y^2} \right) \right] M_i(x, y, f); \quad i = 1, 2. \quad (3.32)$$

$$M(x, y, f) = \cos \left[ 2\pi i f (\sqrt{x^2 + y^2}) \right] \quad (3.33)$$

donde  $M(x, y, f)$  es la función de demodulación,  $f$  es la frecuencia de la función senoidal,  $\delta_x$  y  $\delta_y$  son las constantes del espacio Gaussiano a lo largo del eje  $x$  y  $y$  respectivamente.

La selección de los parámetros de las ecuaciones (3.32 y 3.33) es muy similar a la de los filtros de Gabor. Los filtros simétricos circulares pueden capturar la información de una imagen en una banda de frecuencia específica, sin embargo no pueden proveer de información con respecto a la orientación debido a que son simétricos.

Por último se extrae el vector característico mediante la división de la información en bloques de  $8 \times 8$  de la imagen filtrada, obteniendo un total de 384 vectores característicos. Cada característica es el promedio de la desviación absoluta y cada bloque se define de la siguiente manera:

$$V = \frac{1}{N} \left( \sum_N |f(x, y) - m| \right) \quad (3.34)$$

donde  $N$  es el número de píxeles en el bloque de la imagen,  $m$  es la media y  $f(x, y)$  es el valor del punto  $(x, y)$ .

El promedio de la desviación absoluta de cada bloque filtrado constituyen los componentes del vector característico, los cuales serán de una dimensión (1D).

### 3.5.3. Diseño del clasificador

Para la comparación de iris se emplea una clasificación mediante el método de la línea característica más cercana. El método define la distancia de un punto  $p_x$  y la línea característica  $\overline{p_i p_j}$  de la siguiente manera:

$$D(p_x, \overline{p_i p_j}) = \|p_x - p_i\| \quad \beta < T_1 \quad (3.35)$$

$$D(p_x, \overline{p_i p_j}) = \|p_x - p_x\| \quad T_1 \leq \beta \leq T_2 \quad (3.36)$$

$$D(p_x, \overline{p_i p_j}) = \|p_x - p_i\| \quad \beta > T_2 \quad (3.37)$$

donde  $p_x$  es la proyección de  $p_x$  sobre la línea característica,  $\overline{p_i p_j}$  es llamada el parámetro de posición,  $T_1$  y  $T_2$  son dos umbrales. Estos umbrales, de acuerdo al método de la línea característica más cercana, les corresponden los valores de  $T_1 = -\infty$  y  $T_2 = +\infty$ , sin embargo experimentalmente se eligieron los valores de  $T_1 = -0.5$  y  $T_2 = 1.5$  debido a que produce tan buenos resultados como en el método original [28].

# Descripción de los métodos propuestos

---

El sistema desarrollado cuenta con varias etapas similares a los algoritmos que ya se encuentran implementados sin tomar en cuenta la adquisición de la imagen del iris. En la fase de localización se emplea como base el operador integro-diferencial implementado por Daugman, se propone realizar una ecualización de histogramas para mejorar el contraste de las imagen logrando incrementar la diferencia entre la esclera y el iris.

En la etapa de normalización se realizó la implementación del algoritmo de Daugman para realizar el mapeo del área de análisis del iris de coordenadas polares a rectangulares; para eliminar algunos problemas de iluminación y reflejos se optó por aplicar ahora una ecualización de histogramas local realzando las características del iris.

Las aportaciones fuertes de la tesis se centran en la extracción de características, como primer método se explica el de muestreo que toma subimágenes del área que representa al iris; de cada subimagen se obtendrá su histograma acumulado y el conjunto de estos formará la plantilla que represente al iris. La comparación se realizará estableciendo un criterio de aceptación y rechazo similar a la propuesta por Daugman empleando la diferencia del histograma de cada subimagen.

El otro método que se presenta es la representación del iris a través de un modelo estocástico ARX (Autoregressive with exogenous input), la fase de comparación será realizada por la herramienta de mínimos cuadrados lineales.

Después de la explicación de los métodos se proponen técnicas de optimización enfocados principalmente al método de muestreo por la naturaleza estadística del mismo, se proponen tres tipos de muestreo que serán comparados para probar el que ofrece mejores resultados.

La técnica de recocido simulado será empleada para mejorar los resultados de cada método; para el método ARX serán mejorados los parámetros del modelo que representa al iris, en cambio para el método de muestreo servirá para encontrar la combinación que brinde el más alto desempeño.

En las últimas secciones se resume el funcionamiento general de cada método en sus dos modos de operación, identificación y verificación.

## 4.1. Localización del iris

La localización de los límites límbico y pupilar es llevado a cabo por un programa que incorpora el operador integro-diferencial mostrado en la ecuación 3.2.

Para localizar el límite límbico, la implementación del operador emplea un detector de orillas circulares, en el cual se hace un barrido de toda la imagen con el fin de buscar una trayectoria circular donde exista un máximo cambio en el valor de los píxeles variando el radio  $r$  y la posición de su centro  $(x, y)$ . En la primera iteración se hace la búsqueda de todo el contorno circular, para la segunda iteración tan sólo se realiza la búsqueda parcial del contorno circular, es decir de  $315^\circ$  a  $45^\circ$  y de  $135^\circ$  a  $225^\circ$  con una búsqueda más fina, reduciendo el diferencial del radio y el desplazamiento de las coordenadas del centro (Ver figura 4.1).

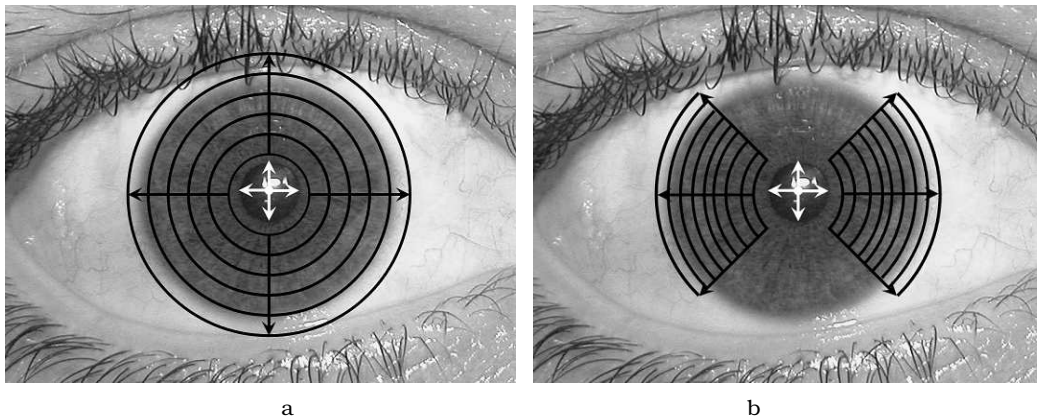


FIGURA 4.1: Las flechas negras muestran las variaciones de radio  $r$  y las flechas blancas las variaciones del centro  $(x, y)$  a) Representación de la búsqueda de los límites límbico y pupilar en la primera iteración. b) Representación de la búsqueda de los límites límbico y pupilar en las siguientes iteraciones.

Para localizar la pupila se repite el procedimiento realizado en el límite límbico, sólo que aquí se busca un mínimo cambio en el valor de los píxeles variando el radio  $r$  y la posición de su centro  $(x, y)$ ; sin embargo la variación se realiza barriendo la zona límbica ubicada en el paso anterior. Un ejemplo de la localización de los límites se presenta en la figura 4.2, estos límites se pueden denotar como límite límbico o exterior y la frontera pupilar o interior.

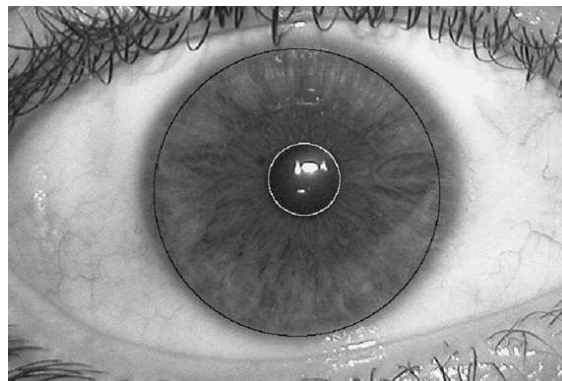


FIGURA 4.2: Localización del iris y la pupila.

Dado que las imágenes contenían ruido, muchas de ellas no son correctamente localizadas, por obstrucciones de pestañas o párpados, iluminación deficiente, reflejos e incluso algunos iris son muy claros para que se presentaran una diferencia notable entre el iris y la esclera.

Algunos ejemplos de localizaciones erróneas se muestran en la figura 4.3.

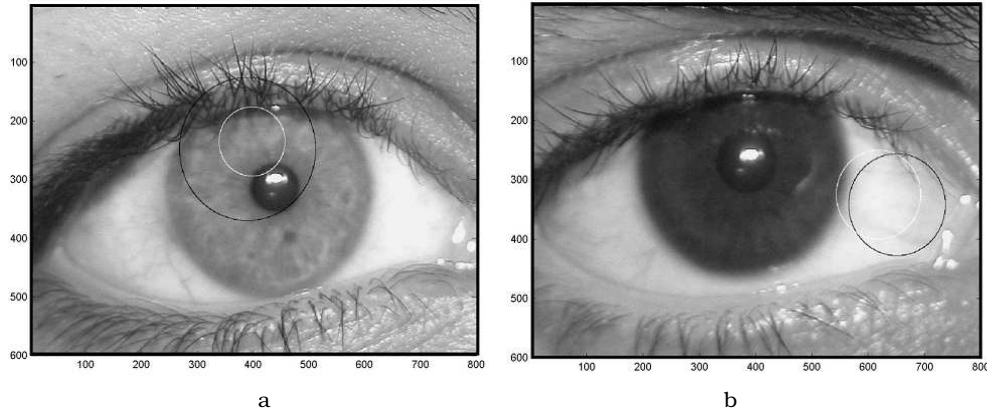


FIGURA 4.3: Detección fallida del límite pupilar y límbico. a) El iris es muy claro y como se observa se pierde localizando una zona cerca de las pestañas. b) El ojo está parcialmente abierto por lo que se tienen dificultades para localizar correctamente los límites del iris.

#### 4.1.1. Ecuación del histograma para mejorar la localización del iris

Con el objetivo de corregir las deficiencias del programa que detecta los límites del iris se propone modificar la imagen para contrarrestar los problemas de iluminación que presenta las bases de datos. El preprocesamiento seleccionado es el de ecualización del histograma, el cual es una transformación de contraste (diferencia entre los píxeles claros y los oscuros) de una imagen y se emplea el histograma para averiguar si la imagen es clara u oscura, por lo que se tendrá que tomar en cuenta todos los píxeles que conforman la imagen.

Si se quiere tomar la información global de toda la imagen, la manera más sencilla es analizar y modificar su histograma, con ello se pretende que éste se ajuste lo más posible a una forma predeterminada. La forma más usual es la conocida como ecualización por histograma, en que se tiene como fin que éste sea horizontal, es decir, que para todos los valores de gris se tenga el mismo número de píxeles.

La ecualización del histograma se realiza trabajando sobre el histograma acumulado:

$$H(i) = \sum_{k=0}^{L-1} h(k) \quad (4.1)$$

donde L es el nivel de grises de la imagen, que en este caso es de 256.

Si el histograma fuese totalmente plano, el histograma acumulado para cada nivel de gris sería:

$$G(i') = (i' + 1) \frac{NM}{256} \quad (4.2)$$

donde N y M son las dimensiones de la imagen y 256 el número de niveles de gris de la imagen.

Idealmente se desea que  $G(i') = H(i)$ , luego entonces se pueden igualar la ecuación (4.1) y (4.2) para obtener:

$$H(i) = (i' + 1) \frac{NM}{256} \quad (4.3)$$



Se despeja el valor del nivel de grises ideal de la ecuación (4.3) obteniendo la siguiente expresión:

$$i' = \frac{256}{NM}H(i) - 1 \quad (4.4)$$

Sin embargo, los niveles de grises son números enteros, por lo que se realiza un cambio en los niveles de grises siguiendo la ecuación:

$$i' = \left[ \frac{256}{NM}H(i) - 1 \right] \quad (4.5)$$

donde  $i'$  es el nuevo valor del nivel de gris,  $i$  es el antiguo valor del nivel de gris y  $[ \ ]$  denota que sólo se tomará la parte entera del cálculo [16],[20].

Después de aplicar el preprocesamiento en las imágenes se mostró mejora en la localización de los límites, en la figura 4.4 se muestran las imágenes de la figura 4.3 y los resultados de esta etapa.

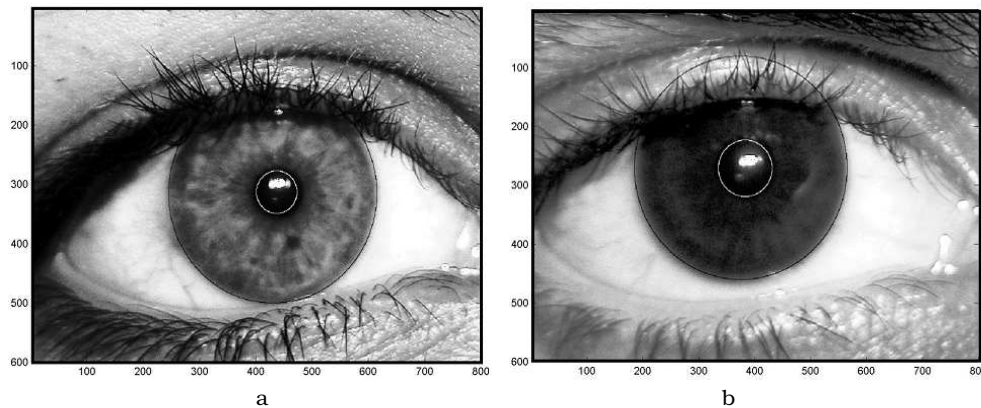


FIGURA 4.4: Detección del límite pupilar y límico de los ojos de la figura 4.3. a) El iris claro cambio de contraste y se volvió más oscuro logrando diferenciar la esclera y el iris. b) La esclera se hizo más nítida lo que permite realizar una localización más efectiva.

## 4.2. Normalización

Como se sabe, la pupila humana tiene la capacidad de dilatarse y contraerse dependiendo de las condiciones de iluminación, por lo que es necesario estandarizarlas. El área que se encuentra entre el límite pupilar y límite límico es mapeada de coordenadas polares a rectangulares (ver figura 4.5) empleando las ecuaciones (3.17) y (3.18).

Un ejemplo de la etapa de normalización se muestra claramente en la figura 4.6, las dos imágenes pertenecen a la misma persona y ambas tiene diferente iluminación lo que provoca que la pupila se dilate, sin embargo la imagen obtenida de cada muestra es muy similar y permite que puedan ser evaluadas como la misma a pesar de la dilatación presentada por la pupila.

Del proceso de normalización se obtiene una imagen que puede ser representada por una matriz de pixeles de 720 columnas y 200 renglones que es proporcional al radio unitario y los  $360^\circ$  que comprenden la circunferencia, dicha matriz será nombrada como imagen normalizada del iris.

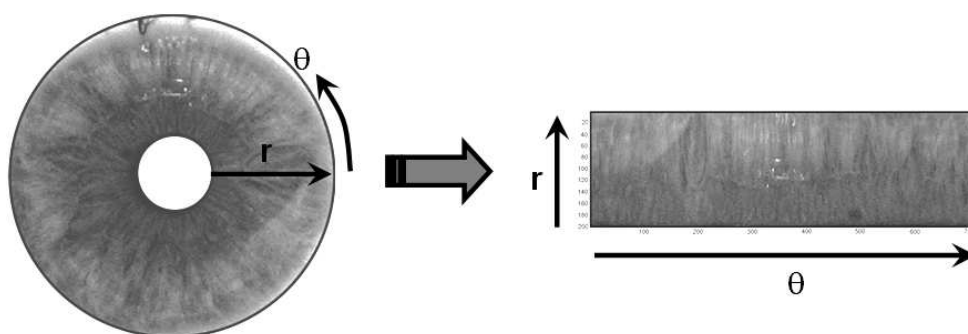


FIGURA 4.5: Remapeo del área del análisis de coordenadas polares a rectangulares.

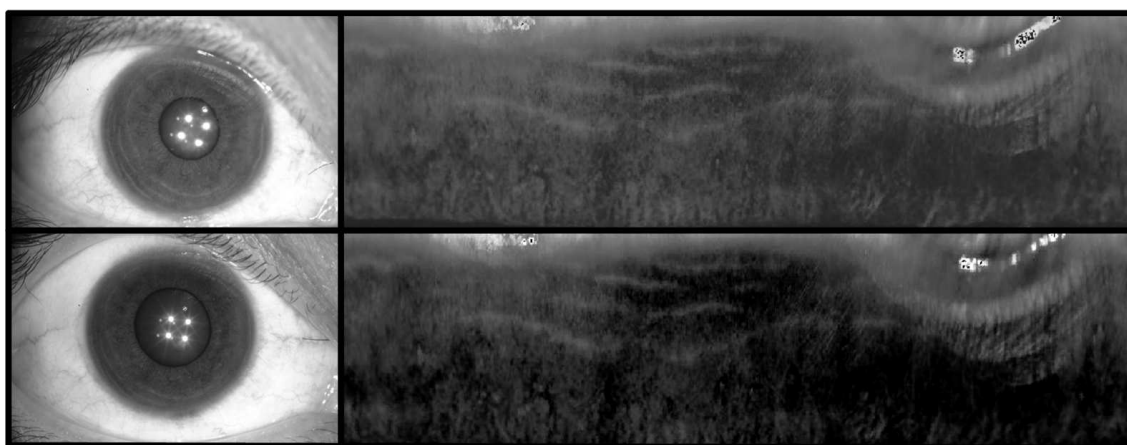


FIGURA 4.6: Ejemplo de dilatación del iris de una misma persona en la etapa de normalización.

### 4.3. Extracción de características

La figura 4.6 aún muestra diferencias entre pares de ojos que son iguales debido a las condiciones de mala iluminación que presenta, teniendo como resultados dos imágenes con tonalidades de grises diferentes.

Se realiza una nueva modificación de histograma con el fin de alterar el contraste de la imagen. Esta es considerada como una modificación local debido a que idealmente ya no interfiere ningún elemento ajeno a la zona a inspeccionar como párpados, pestañas, color de piel, reflejos, esclera, etc. Un ejemplo del preprocesamiento de la imagen normalizada del iris se muestra en la figura 4.7, que es el resultado de haber ecualizado las imágenes de la figura 4.6, obteniendo características más definidas y mucho más similares entre sí.

#### 4.3.1. Extracción de características por el método de Muestreo

Como su nombre lo sugiere la imagen normalizada del iris será muestreada, cada muestra será una zona rectangular que se le denominará subimagen. Se emplea el histograma acumulado para representar las características del iris, este cálculo se realiza sobre las subimágenes muestreadas de la imagen normalizada.

El objetivo es tomar las principales características del iris para tener una buena representación sin tener información innecesaria, por lo que se proponen tres tipos de muestreo explicados en la sección de 4.6.

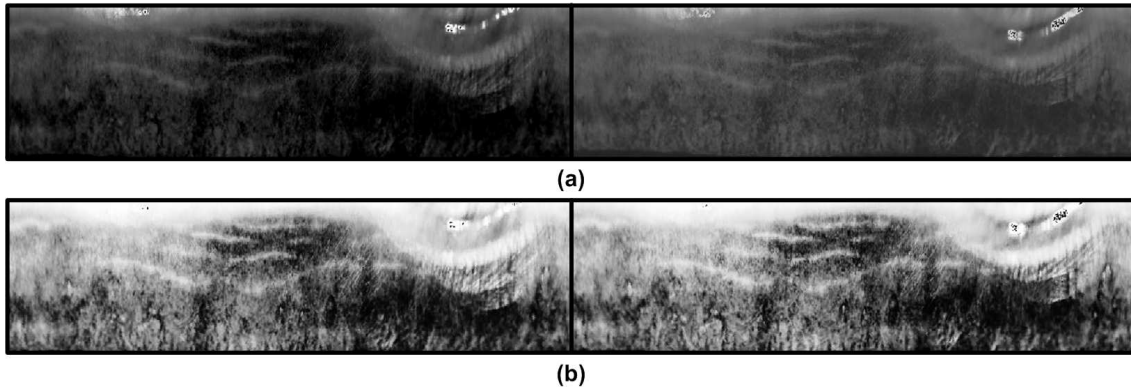


FIGURA 4.7: Imágenes normalizadas que representan al iris de la figura 4.6. a) Imagen de la matriz normalizada obtenida sin ecualización de histogramas. b) Imagen de la matriz normalizada obtenida con ecualización de histogramas.

Para cada subimagen se obtiene el histograma acumulado mediante la ecuación (4.1), el patrón del iris se formará por el conjunto de subimágenes, esta información formará la base de datos (Ver figura 4.15 y 4.16). El número óptimo de subimágenes y del tamaño de los mismos será determinado por experimentos variando estos parámetros.

### 4.3.2. Extracción de características por el método ARX (Autoregressive with exogenous input)

Se propone ajustar la imagen o matriz de pixeles normalizada mostrada en la figura 4.7 (b) a un modelo estocástico ARX. Según [17] y [24] un modelo el cual representa a un sistema dinámico en tiempo discreto cuya salida  $Y_k$  depende de la entrada  $U_k$  y de los valores anteriores de ambas variables, quedando definida por la siguiente ecuación:

$$y_k = a_1 y_{k-1} + a_2 y_{k-2} + \dots + a_{na} y_{k-na} + b_1 u_{k-1-nd} + \dots + b_{nb} u_{k-nb-nd} \quad (4.6)$$

es decir:

$$\frac{Y(z)}{U(z)} = \frac{b_1 z^{-1} + b_2 z^{-2} + \dots + b_{nb} z^{-nb}}{1 - a_1 z^{-1} - a_2 z^{-2} - \dots - a_{na} z^{-na}} z^{-nd} \quad (4.7)$$

Para representar al iris como un modelo ARX es necesario definir una salida y una entrada. El iris esta representado por una matriz normalizada de 200x720 pixeles y cada pixel contiene el nivel de la escala de gris (0-255) y representará la salida del modelo ARX; para definir la entrada se utilizará la ubicación del pixel en la matriz normalizada, sólo se utilizará la ubicación en los renglones para su modelación.

Si se empleará la posición de todos los pixeles la representación del modelo se vería afectado por el ruido, por alguna posible rotación o traslación de la imagen. Se propone realizar una segmentación de la matriz normalizada en zonas rectangulares de tamaño variable (Ver la figura 4.8) por lo tanto la salida quedará definida por el media empleando la siguiente ecuación:

$$y_{kl} = \frac{1}{grid^2} \sum_{i=1}^{grid} \sum_{j=1}^{grid} p_{i'j'} \quad (4.8)$$

donde  $y_{kl}$  es la media de los píxeles que comprenden la zona cuadrangular,  $k$  es el número de columna y  $l$  el número de renglón de la segmentación resultante;  $i'$  y  $j'$  son el número de columna y de renglón de la imagen del iris donde  $i' = k * (grid - 1) + i$  y  $j' = l * (grid - 1) + j$ . El tamaño del lado en píxeles de la zona cuadrangular será denotado por  $grid$  y es uno de los valores que serán variados al realizar los experimentos.

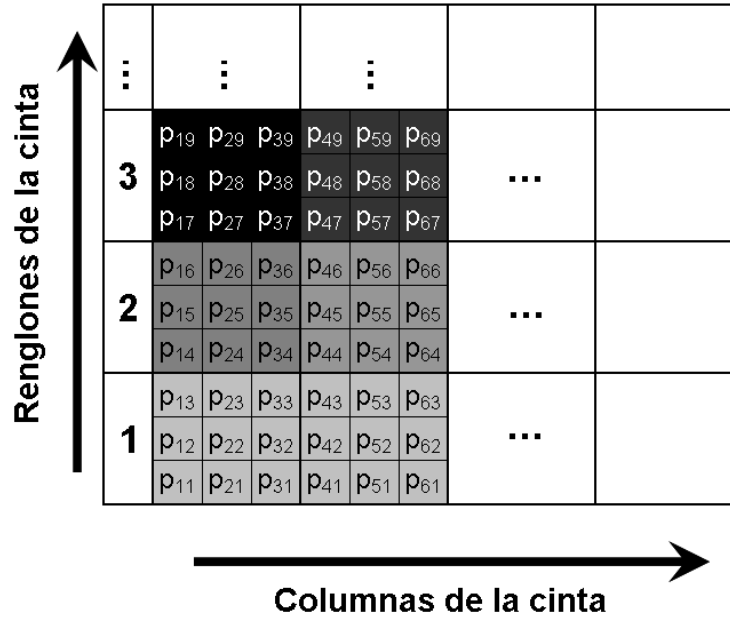


FIGURA 4.8: Procedimiento para la definición de la entrada y salida que formarán el modelo ARX, el promedio de los píxeles que se encuentran en la zona cuadrangular formará el vector de salidas y su posición formará el vector de entradas.

La última suposición es que no habrá tiempo muerto en el modelo, por lo dando  $nd = 0$  y sólo se variarán el número de na's y nb's de la ecuación (4.6). El objetivo es que cada iris plantilla tenga sus propios parámetros y sean estos los que representen las características del iris.

## 4.4. Comparación

Debido a la naturaleza de los resultados de cada método será comparado de diferente manera, el de muestreo utiliza la diferencia obtenida por la comparación de cada histograma acumulado de las subimágenes muestreadas, mientras que el ARX emplea la mínima diferencia del error cuadrático medio.

### 4.4.1. Comparación para el método de muestreo

La manera en que se realizará la comparación para saber si un iris pertenece o no a la base de datos es por medio de la suma de la distancia Euclidiana entre cada una de las subimágenes. Ésta es una de las formas más comunes para medir diferencias o semejanzas entre observaciones u objetos; idealmente si los iris son iguales la distancia Euclidiana debe resultar de cero. La distancia Euclidiana empleada para comparar cada iris está dada por la siguiente ecuación:

$$DE_{ij} = \sqrt{\sum_{k=1}^n (P_k^{IT} - P_k^{IS})^2} \quad (4.9)$$

donde  $DE_{i,j}$  es la distancia Euclidiana entre el iris muestra  $i$  y el iris de la base  $j$ ,  $n$  es el número de subimágenes muestreadas y  $P_k^{IT}$  y  $P_k^{IS}$  es el histograma acumulado de cada subimagen del iris que pertenece a la base de datos y a la muestra respectivamente.

#### 4.4.2. Comparación para el método ARX por medio de mínimos cuadrados

Entre los métodos de identificación de modelos paramétricos discretos se encuentra el de mínimos cuadrados, el cual tiene la característica de minimizar un criterio de error entre el modelo real y el modelo estimado. Se trata de obtener los parámetros  $a_i$  y  $b_i$  que minimizan su energía, definiendo los siguientes vectores:

$$\Psi^T = [y_{k-1} \ y_{k-2} \ \cdots \ y_{k-na} \ u_{k-1-nd} \ \cdots \ u_{k-nb-nd}] \quad (4.10)$$

$$\theta_N = [a_1 \ a_2 \ \cdots \ a_{na} \ b_1 \ \cdots \ b_{nb}] \quad (4.11)$$

por lo tanto el sistema general de la ecuación (4.6) puede describirse como:

$$y_k = \Psi_k^T \theta \quad (4.12)$$

De acuerdo a [17] y [24] método de mínimos cuadrados busca encontrar a partir de un muestreo de los datos reales de un proceso, los valores del vector  $\theta$  que ajusten de la mejor forma a la respuesta  $y_k$  del sistema a partir de una ecuación de predicción:

$$\hat{y}_k = \Psi_k^T \theta + e_k \quad (4.13)$$

donde  $k = nm, nm + a, \dots, N$  y  $nm = \max(na, nb + nd)$ .

Como se puede deducir de la ecuación (4.13) el error puede obtenerse de la siguiente ecuación:

$$e_N = Y_N - \Psi_N \theta_N \quad (4.14)$$

El objetivo de mínimos cuadrados es minimizar el error cuadrático medio:

$$J = \sum_{k=nm}^N e_k^2 = e_N^T e_N \quad (4.15)$$

La base de datos estará formada por un modelo ARX que representará cada uno de los iris. Si llega un iris al sistema, el método de comparación será por medio de la ecuación (4.15), siendo obtenido el error de la ecuación (4.14) donde  $Y_N$  y  $\theta_N$  pertenecen al iris de la base de datos ( $I^T$ ) y la matriz  $\Psi_N$  la formarán los datos del iris muestra ( $I^S$ ), por lo tanto:

$$e_N = Y_N^{IT} - \Psi_N^{IS} \theta_N^{IT} \quad (4.16)$$

donde  $e_N$ ,  $Y_N^{IT}$ ,  $\Psi_N^{IS}$  y  $\theta_N^{IT}$  se definen como:

$$e_N = \begin{bmatrix} e_{nm} \\ e_{nm+1} \\ \vdots \\ e_N \end{bmatrix}, \quad Y_N^{IT} = \begin{bmatrix} y_{nm}^{IT} \\ y_{nm+1}^{IT} \\ \vdots \\ y_N^{IT} \end{bmatrix},$$

$$\Psi_N^{I^S} = \begin{bmatrix} y_{nm-1}^{I^S} & \cdots & y_{nm-na}^{I^S} & u_{nm-1-nd}^{I^S} & \cdots & u_{nm-nb-nd}^{I^S} \\ y_{nm}^{I^S} & \cdots & y_{nm-na+1}^{I^S} & u_{nm-nd}^{I^S} & \cdots & u_{nm-nb-nd+1}^{I^S} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ y_{N-1}^{I^S} & \cdots & y_{N-na}^{I^S} & u_{N-1-nd}^{I^S} & \cdots & u_{N-nb-nd}^{I^S} \end{bmatrix} \quad y \quad \theta_N^{I^T} = \begin{bmatrix} a_1^{I^T} \\ a_2^{I^T} \\ \vdots \\ a_{na}^{I^T} \\ b_1^{I^T} \\ \vdots \\ b_{nb}^{I^T} \end{bmatrix}$$

## 4.5. Decisión

Un sistema de reconocimiento biométrico puede operar en dos modos dependiendo del contexto de la aplicación como fue explicado en 2.2 y los métodos serán probados en cada uno de los modos de operación.

### 4.5.1. Decisión basada en el modo identificación

La principal característica de este modo de operación radica en que una vez capturada la imagen del iris es comparada contra una base de datos codificada previamente; el resultado será comprobar si pertenece o no a la base de datos, en caso de pertenecer también deberá establecer la identidad de la persona.

Para determinar la identidad de una persona es necesario establecer un umbral  $t$ , cuyo valor dependerá del tipo de método; el umbral se calcula con un conjunto de muestras donde el 50% de ellas pertenecen a la base de datos y 50% no pertenecen, como se puede observar en la figura 4.9.

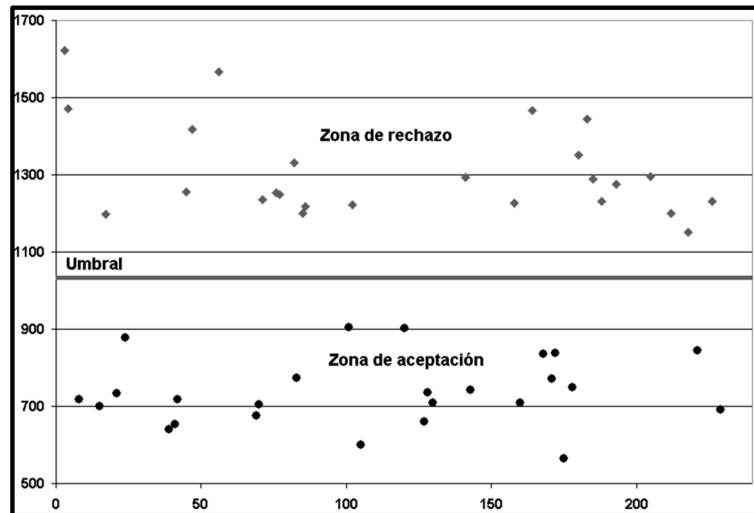


FIGURA 4.9: La figura muestra el ejemplo para obtener el umbral con 25 iris que pertenecen a la base de datos y 25 que no pertenecen.

### 4.5.2. Decisión basada en el modo verificación

En el modo de operación de verificación sólo se determina si dos pares de iris son o no son iguales. Con los resultados obtenidos es posible determinar las curvas de distribución del genuino  $P_G(DE)$  y impostor  $P_I(DE)$  ambas en función de la distancia Euclidiana dependiendo del método, como se observa en la figura 2.3. A partir de estas curvas se calcularán las siguientes proporciones determinando un umbral  $t$ :

- **Proporción de Verdaderos Positivos:** proporción de pares de muestras que fueron aceptadas como iguales siendo que si lo eran. Su probabilidad es calculada como:

$$PVP = \int_0^t P_G(DE) dDE \quad (4.17)$$

- **Proporción de Verdaderos Negativos:** proporción de pares de muestras que fueron rechazadas como iguales siendo que no lo eran. Su probabilidad es calculada como:

$$PVN = \int_t^1 P_I(DE) dDE \quad (4.18)$$

- **Proporción de Falsos Positivos:** proporción de pares de muestras que fueron aceptadas como iguales siendo que no lo eran. Su probabilidad es calculada como:

$$PFP = \int_t^1 P_G(DE) dDE \quad (4.19)$$

- **Proporción de Falsos Negativos:** proporción de pares de muestras que fueron rechazadas como iguales siendo que si lo eran. Su probabilidad es calculada como:

$$PFN = \int_0^t P_I(DE) dDE \quad (4.20)$$

Para representar todos los puntos de operación del umbral  $t$  se elabora la curva de Características Operativas del Receptor mejor conocida como curva ROC explicada en la sección 2.3, que determina el grado de confiabilidad de decisión del sistema de reconocimiento biométrico.

## 4.6. Técnicas de optimización

### 4.6.1. Muestreo aleatorio con reposición

Para que las inferencias sean válidas, las muestras deben ser representativas y cuando este proceso garantiza que cada uno de los elementos de la población tiene la misma oportunidad de ser incluidos en dicha muestra, denominamos al proceso de selección: muestreo aleatorio con reposición o simple.

Para lograr que la muestra salga insesgada, es deseable seleccionar variables aleatorias que representarán una observación de la población. Las variables a seleccionar serán las coordenadas de las subimágenes. Por definición las variables aleatorias  $(X_1, X_2, \dots, X_n)$  son independientes con cada  $X_i$  y con la misma distribución de probabilidad.[30]

Como su nombre lo indica cada elemento muestreado que es extraído por este método es regresado al total de la población teniendo cada elemento la misma probabilidad de ser elegido.

### 4.6.2. Muestreo aleatorio sin reposición

Para realizar un muestreo más efectivo se implementará el muestreo aleatorio sin reposición, en el cual se dan las siguientes circunstancias:

- La probabilidad de que  $X_i$  sea elegido en primer lugar es  $\frac{1}{N}$ .
- La probabilidad de que sea elegido en el segundo intento es de  $\frac{1}{N-1}$ .
- En el  $i + 1$  intento en el cual  $X_i$  no haya sido seleccionado de  $N - i$  elementos, la probabilidad de que ocurra en ese momento es de  $\frac{1}{N-i}$ .
- Tiene una distribución uniforme en el intervalo  $[0, 1] U \rightarrow U(0, 1)$  [12].

Una pequeña modificación se realizará a este muestreo, la idea principal de implementar este método es que un pixel pertenezca tan sólo a una subimagen, es decir idealmente las subimágenes no deben empalmarse repitiendo información. Cada que es establecida la posición de una subimagen las siguientes no podrán muestrear ningún pixel que pertenezca a ésta, en caso de no cumplir la condición es rechazada la subimagen y es generada otra.

### 4.6.3. Muestro Hammersley

El muestreo Hammersley es una parte de los métodos cuasi-Monte Carlo o de la familia de muestreo de baja discrepancia. El prefijo cuasi se refiere a que el muestreo emplea un algoritmo determinístico para generar muestras en un espacio  $n$ -dimensional; por lo tanto los generadores de números aleatorios Monte Carlo, son llamados pseudo-aleatorios ofreciendo un grado de uniformidad más alta, llevando el cálculo a un mejor rendimiento en bajas dimensionalidades [38].

Métodos cuasi-Monte Carlo como la secuencia de Hammersley muestran un bajo límite de error en problemas multidimensionales como la integración, según [22] el límite de error para un método cuasi Monte Carlo es de  $O(N^{-1/2})$  y para la integración clásica es de  $O(N^{-2/n})$ . Sin embargo, la secuencia Hammersley tiene un límite de error bajo con  $O(N^{-1}(\log_{10}N)^{n-1})$  donde  $N$  es el número de muestras y  $n$  son las dimensiones del espacio designado. Usualmente conforme  $n$  aumenta Hammersley muestra mejores resultados notando que el límite de error de pseudo-Monte Carlo es un límite probabilístico. De acuerdo a [13] y [19], el algoritmo que genera un conjunto de puntos  $N$  el cual hace uso de la notación de base  $R$  de un entero, es decir que un entero  $p$  se puede representar en notación de base  $R$  como:

$$\begin{aligned} p &= p_m p_{m-1} \cdots p_2 p_1 p_0 \\ p &= p_0 + p_1 R + p_2 R^2 + \cdots + p_m R^m \end{aligned} \quad (4.21)$$

donde  $m = \lceil \log_R p \rceil = \left\lceil \frac{\ln p}{\ln R} \right\rceil$ , y los corchetes  $\lceil \cdot \rceil$  denotan la porción entera del número dentro de los corchetes.

La función inversa de la notación de base  $R$  se construye en un intervalo único de  $[0, 1]$  invirtiendo el orden de los dígitos de  $p$  obtenidos de la ecuación 4.22 alrededor del punto decimal teniendo como resultado:

$$\begin{aligned} \phi_R(p) &= 0.p_0 p_1 p_2 \cdots p_m \\ \phi_R(p) &= p_0 R^{-1} + p_1 R^{-2} + \cdots + p_m R^{-m-1} \end{aligned} \quad (4.22)$$



Para obtener la secuencia de puntos Hammersley de  $k$ -dimensiones está dada la siguiente expresión:

$$x_n(p) = \left( \frac{p}{N}, \varphi_{R_1}(p), \varphi_{R_2}(p), \dots, \varphi_{R_{n-1}}(p) \right) \quad (4.23)$$

donde  $p = 0, 1, 2, \dots, N - 1$ ; y los valores de  $R_1, R_2, \dots, R_{n-1}$  son los primeros  $n - 1$  números primos (2, 3, 5, 7, 11, 13, 17, ...). Esto es aprovechado para generar el conjunto de puntos  $N$  en un espacio de  $k$ -dimensiones  $[0, 1]^k$ .

#### 4.6.4. Comparación entre los métodos de muestreo

Una de las desventajas del muestreo aleatorio simple es que no presenta uniformidad, es decir se pueden presentar aglomeraciones en puntos o zonas en particular y zonas donde el muestreo es nulo, como se observa en la figura 4.10.

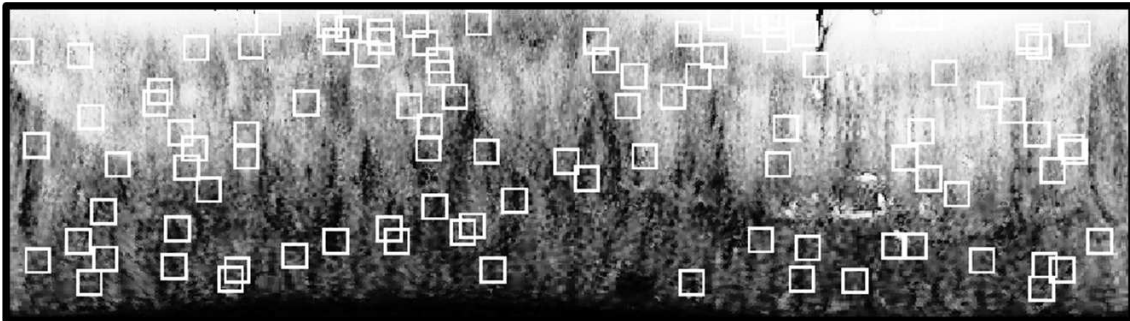


FIGURA 4.10: Muestreo de una imagen normalizada de 720x200 píxeles con el método aleatorio simple, se pueden observar que hay zonas que no son muestreadas y zonas donde se acumulan muchos datos.

El muestreo aleatorio sin reposición se implementó evitando que las zonas muestreadas se repitan, esto con el fin de evitar la recopilación innecesaria de información, el resultado de la implementación se muestra en la figura 4.11, sin embargo aún se encuentran zonas sin muestrear a pesar de que no se haya sido repetida la información.

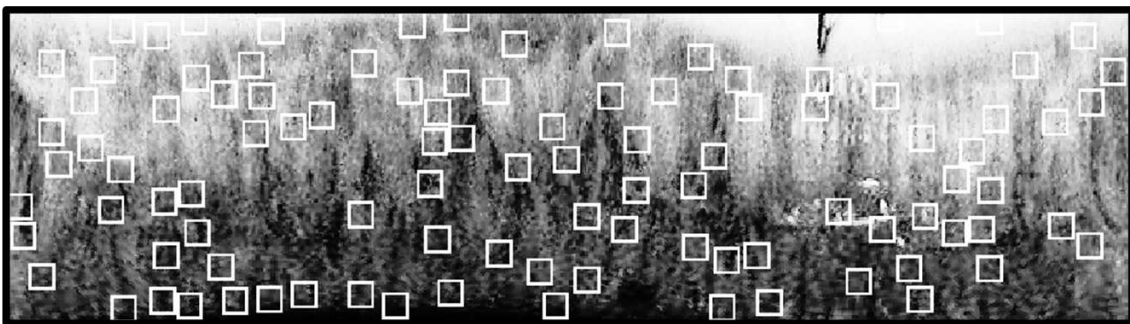


FIGURA 4.11: Muestreo de una imagen normalizada de 720x200 píxeles con el método aleatorio sin reposición, se pueden observar que hay zonas que no son muestreadas.

El resultado de la implementación del algoritmo para muestrear por el método de Hammersley es presentado en la figura 4.12, de esta manera se obtiene un muestreo más uniforme sobre la imagen y teniendo como resultados una caracterización del iris más eficiente.

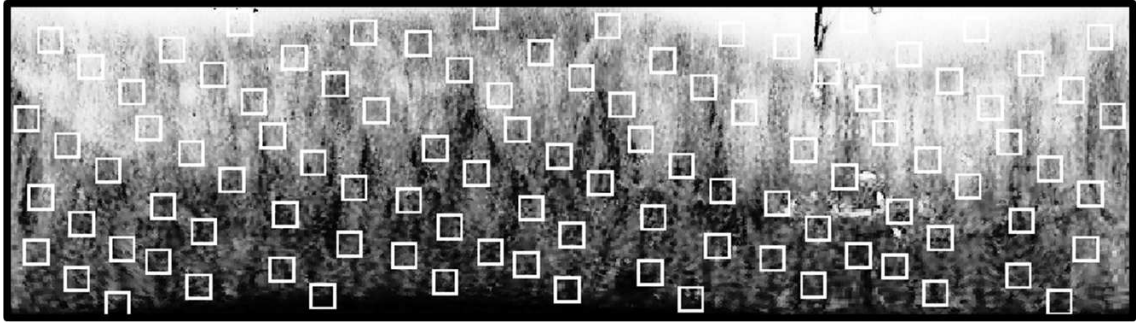


FIGURA 4.12: Muestreo de una imagen normalizada de 720x200 píxeles con el método de Hammersley, se puede observar un muestreo más distribuido.

#### 4.6.5. Recocido Simulado

El concepto de Recocido simulado fue introducido por Kirkpatrick en 1983, esta metaheurística ha mostrado ser una herramienta muy exitosa para resolver problemas de optimización combinatoria. Su simplicidad y buenos resultados en numerosos problemas han convertido a esta herramienta muy popular con cientos de aplicaciones en los más variados cambios.

A pesar de ser una herramienta muy simple de usar, es difícil lograr un buen funcionamiento por ser una estrategia heurística que necesita varias decisiones para que quede totalmente diseñado, estas decisiones tienen gran repercusión en la calidad de las soluciones generadas [14].

Recocido Simulado es generado por el método de Monte Carlo Cadenas de Markov, esta técnica involucra una cadena de Markov no homogénea cuya distribución invariante en  $i$  iteraciones obteniendo:

$$p_i(x) \propto p^{1/T_i}(x) \quad (4.24)$$

donde  $T_i$  es un decrecimiento de la temperatura con  $\lim_{i \rightarrow \infty} T_i = 0$ . Bajo regularidades débiles asumidas en  $p(x)$ ,  $p^\infty(x)$  es una densidad de probabilidad que se concentra en el conjunto del máximo global de  $p(x)$  [1].

La figura 4.13 se muestra el pseudocódigo del algoritmo de recocido simulado. La probabilidad de aceptación es representada por:

$$A(x^{(i)}, x^*) = \min \left\{ 1, \frac{p^{1/T_i}(x^*)q(x^{(i)}|x^*)}{p^{1/T_i}(x^{(i)})q(x^*|x^{(i)})} \right\} \quad (4.25)$$

donde la distribución propuesta  $q(x^*|x^{(i)})$  involucra una muestra candidata de valor  $x^*$  dada por el valor actual  $x^{(i)}$ .

[1] menciona que para obtener un resultado eficiente es importante seleccionar la distribución a maximizar y una apropiada función de decrecimiento  $\alpha$ .

Esta herramienta será empleada para darle solución al número na's y nb's necesarias para representar al iris como un modelo ARX; mientras que para los métodos de muestreo será aplicado sobre aquel que haya dado los mejores resultados, se seleccionará el número de subimágenes y el tamaño de las mismas que permita maximizar su desempeño.

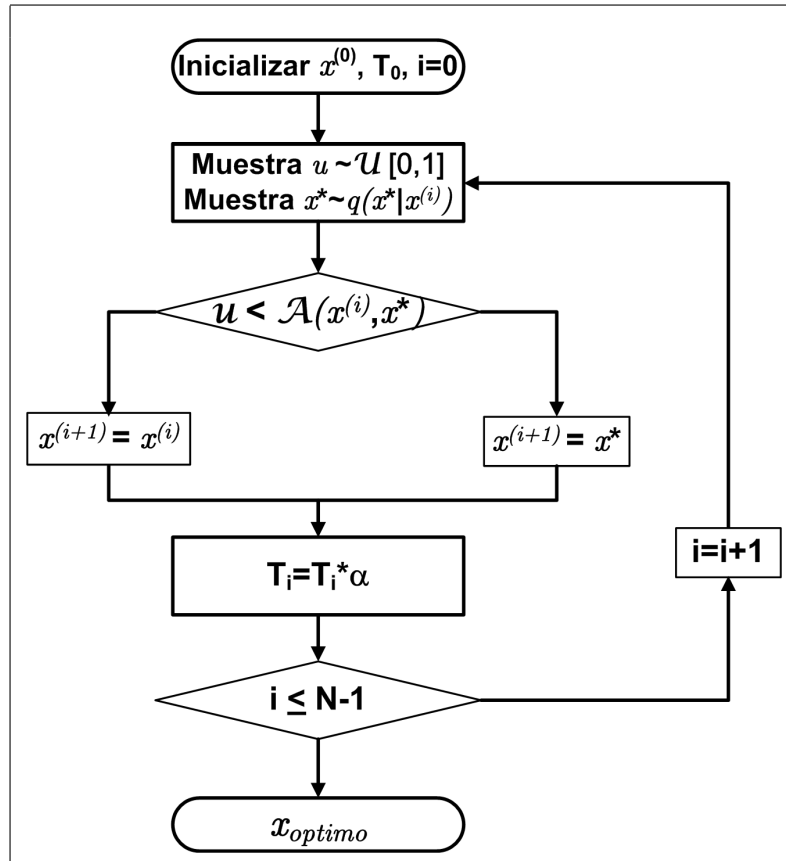


FIGURA 4.13: Algoritmo de Recocido Simulado para N iteraciones.

#### 4.6.6. Muestreo Factorial

El muestreo factorial es una herramienta propuesta por [31]. De acuerdo con [6] el método propuesto es muy efectivo cuando el modelo contiene un gran número de muestras.

El diseño experimental propuesto por [6] está compuesto por experimentos donde se evalúa el cambio de un factor en un tiempo determinado. Cada cambio en ese factor puede asumirse como un número discreto de valores, llamado niveles. La elección del rango de variación varía dependiendo del problema a resolver.

La sensibilidad medida se basa en un concepto llamado efecto elemental que se atribuye al efecto que causa una determinada entrada y se define como:

$$EE_i(x) = \frac{[y(x_1, \dots, x_{i-1}, x_i + \Delta, x_{i+1}, \dots, x_k) - y(x)]}{\Delta} \quad (4.26)$$

donde  $x = (x_1, x_2, \dots, x_k)$  son valores seleccionados de  $\Omega$  que es la región de interés, se asume que es un hipercubo unitario de k-dimensiones.  $\Delta$  es un múltiplo predeterminado de  $1/(p-1)$  y p es el valor del p-nivel.

Las entradas serán definidas como el número de subimágenes, el ancho y el alto de las mismas; los niveles a los que será aplicado el muestreo serán dos. En el primer nivel de muestreo se desecharán los iris de la base de datos que tienen poca probabilidad de ser iguales al iris muestra, la característica de este muestreo será su alta velocidad de ejecución sin perder la efectividad. El segundo nivel de muestreo sólo se trabajará con las imágenes que pasaron el primer nivel; el número de imágenes a analizar se ve reducido, por lo tanto este será un muestreo más fino para obtener una alta efectividad.

Para seleccionar los valores del primer y segundo muestreo se emplearán los valores obtenidos por recocido simulado, los que hayan presentado los menores tiempos serán empleados para el primer muestreo y los que hayan mostrado mejores resultados serán empleados para el segundo muestreo.

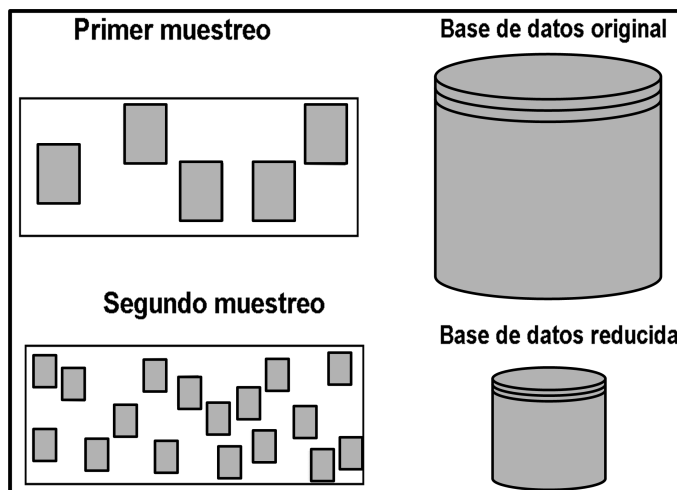


FIGURA 4.14: Ejemplo de los dos niveles de muestreo por el método factorial

## 4.7. Resumen de los métodos propuestos

A continuación se presenta de manera muy puntual la descripción de cada método que fue explicado paso a paso a lo largo de esta sección para los dos métodos propuestos: ARX y muestreo y los dos modos de operación: identificación y verificación.

### 4.7.1. Método ARX en modo identificación

El iris muestra llegará al sistema, de éste será obtenido su modelo ARX descrito por la ecuación 4.6, de ese modelo se tomarán los coeficientes de  $a's$  y  $b's$  para representarlo. Ese vector característico es el que permitirá compararlo contra una base de datos previamente construida.

Para aceptar o rechazar el iris muestra, la comparación será mediante el valor de la ecuación (4.15); primero se determinará con qué plantilla de la base de datos se parece más, posteriormente con respecto a un umbral se decidirá si el iris pertenece o no a la base de datos e identificará quién llegó al sistema. Un esquemático del funcionamiento en modo identificación por el método ARX se muestra en la figura 4.15.

### 4.7.2. Método ARX en modo verificación

Se obtendrá del iris muestra su modelo ARX descrito por la ecuación 4.6, serán tomados los coeficientes de  $a's$  y  $b's$  para representar al iris. El vector resultante es comparado contra un iris plantilla mediante el valor de la ecuación (4.15); el criterio de decisión será por medio de un umbral, la resolución implica dos posibles soluciones, si son o no son iguales los iris comparados. Un esquema del funcionamiento en modo verificación por el método ARX se muestra en la figura 4.16.

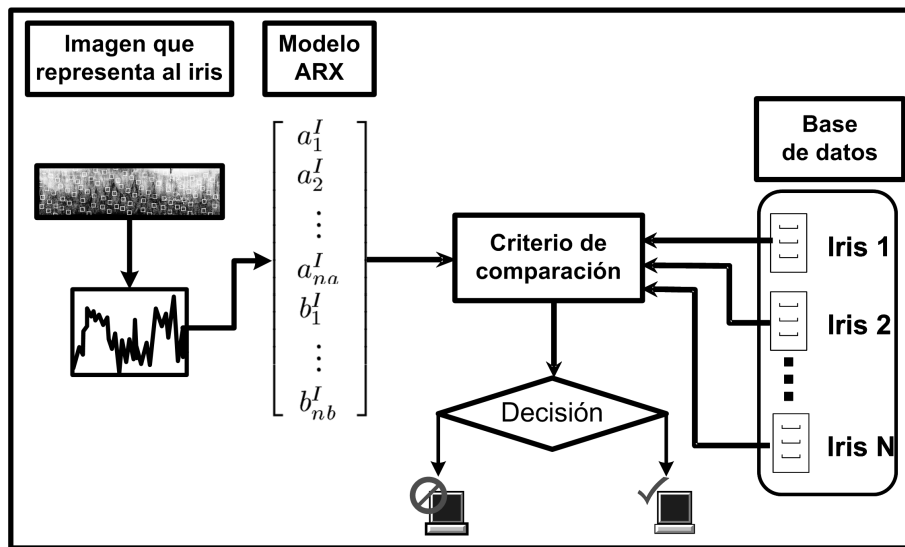


FIGURA 4.15: Descripción del modo identificación del método ARX.

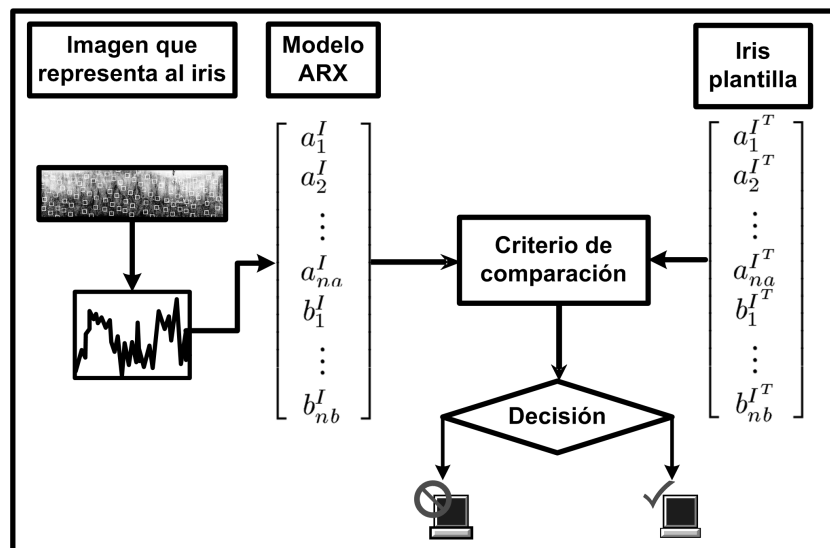


FIGURA 4.16: Descripción del modo verificación del método ARX.

### 4.7.3. Método de Muestreo en modo identificación

El iris muestra llegará al sistema, la imagen obtenida del proceso de normalización será muestreada de acuerdo a los métodos vistos en la sección 4.6. Del muestreo se obtienen subimágenes de la imagen normalizada de  $m \times n$  tamaño de los cuales se obtendrá de cada uno su histograma acumulado; la representación del iris estará dada por el conjunto de histogramas acumulados.

La base de datos contendrá iris que también se encuentran representados por el conjunto de histogramas acumulados, cada histograma será comparado bin a bin para obtener la diferencia entre ellos de acuerdo a la ecuación 4.9. Como primer paso se identificará a cuál iris de la base de datos se parece más y por último, con base a un umbral, se decidirá si éste pertenece o no a la base datos. Una explicación más gráfica es mostrada en la figura 4.17).

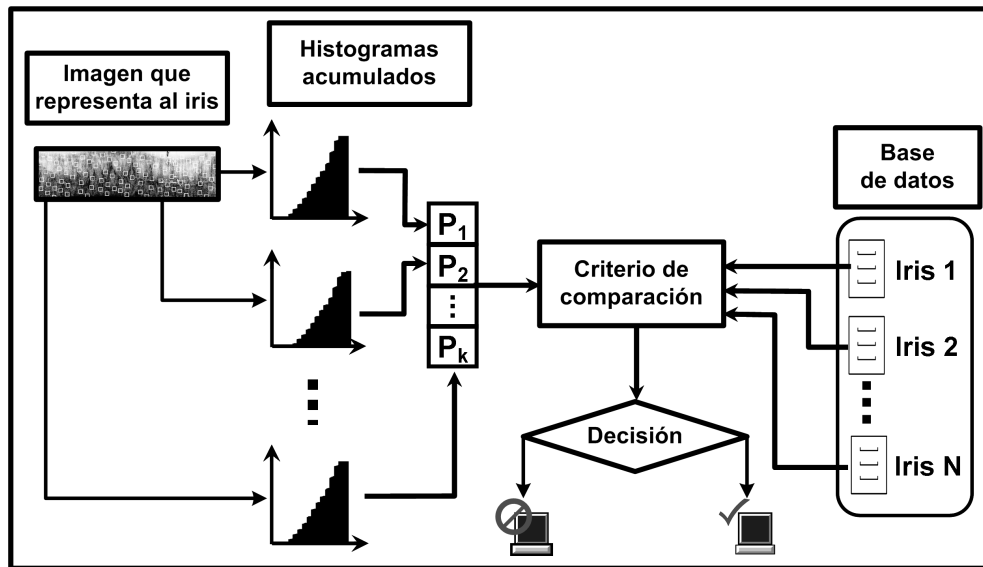


FIGURA 4.17: Descripción del modo verificación del método de muestreo.

#### 4.7.4. Método de Muestreo en modo verificación

El iris muestra que solicite la verificación será muestreado de acuerdo a los métodos vistos en la sección 4.6. De este procedimiento se obtienen subimágenes de  $m \times n$  tamaño de los cuales será obtenido el histograma acumulado de cada uno de ellos. El iris será comparado contra un iris plantilla que se encuentra representado por un conjunto de histogramas acumulados, cada histograma será comparado bin a bin para obtener la diferencia entre ellos de acuerdo a la ecuación 4.9. La resolución del sistema es determinar si son o no iguales. Un ejemplo de su implementación se muestra en el esquema 4.18).

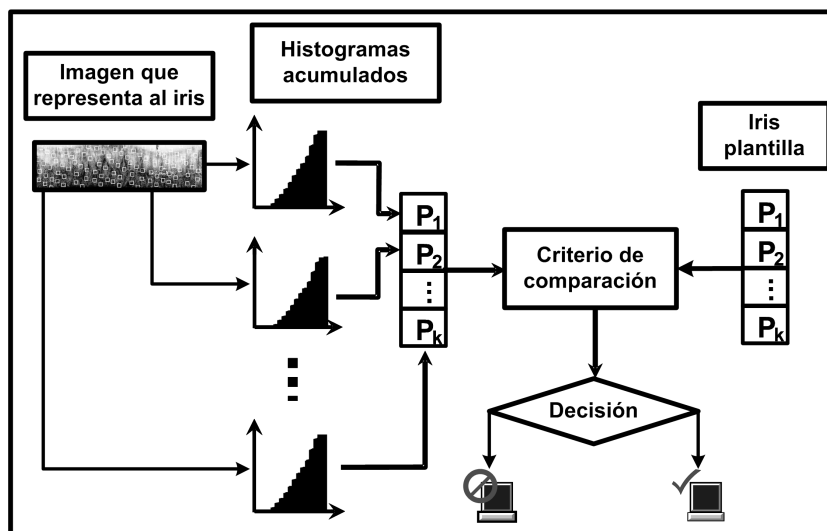


FIGURA 4.18: Descripción del modo verificación del método de muestreo.



# Resultados Experimentales y optimización

---

## 5.1. Introducción

En este capítulo son presentados los resultados aplicando los algoritmos propuestos en la sección 4, donde fue planteada la teoría necesaria para explicar los resultados experimentales y las optimizaciones obtenidas por recocido simulado.

Se emplearon dos bases de datos, Miles Research [37] fue con la primera que se trabajó, fue empleada para probar la eficiencia de los algoritmos con pocas muestras y con deficiencias de iluminación. Para comprobar los resultados obtenidos por Miles Research se empleo la base de datos UBIRIS [36]; ésta es de dominio público e incorpora muestras que presentan ruido por deficiencias en la iluminación, mal enfoque y reflejos.

De acuerdo a la explicación en las secciones 4.5 y 4.7 los experimentos serán probados los modos de operación de un sistema biométrico: identificación y verificación. En la sección 5.2 se ofrece una breve explicación de las condiciones en que fueron realizados los experimentos para cada base de datos.

En la sección 5.3 son descritas cada una de las bases de datos, algunos problemas de ruido que presentan y el número de localizaciones exitosas; Miles Research contiene muestras que no tiene las condiciones en que fueron capturadas las imágenes ni el número de muestras de cada iris, la selección de los iris iguales se realizó seleccionándolas manualmente sin ningún tipo de algoritmo computacional. UBIRIS en cambio es una base de datos mucho más grande donde se especifican las condiciones en que fueron capturadas las imágenes y el número de muestras de cada iris. De acuerdo con [36] es una base de datos que les son incorporadas muestras con ruido a diferencia de CASIA o UPOL que son bases de datos limpias.

En la tercera parte de la sección 5.3 se incorpora una evaluación de las bases de datos donde son especificados los problemas que presenta cada base de datos y en las que fue posible realizar una correcta localización de los límites límbico y pupilar. De acuerdo a [44], [9], [8] y [11] en un sistema real cuando se capturan las una serie de imágenes del iris, se selecciona la que tiene mejor enfoque empleando el algoritmos como el explicado en la sección 3.2.1. Para intentar compensar esa selección, se eliminan muestras con ruido, teniendo como resultado una clasificación de las bases de datos de acuerdo a su tamaño o número de muestras, mientras más pequeña sea, menores muestras con ruido presenta.

En la sección 5.4 son presentados los resultados en modo identificación se apoya de gráficos para comparar el resultado entre los métodos explicados en la sección 4.6 para cada base de datos. Los resultados detallados son mostrados los apéndices A y B, ahí viene especificado la proporción de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos



calculados de acuerdo a la explicación proporcionada en la sección 4.5.2.

La última sección presenta los resultados en modo verificación, en esta parte no fue posible aplicar todos los algoritmos explicados en la sección 4.6. Se apoya de las curvas de distribución genuino-impostor para observar la confiabilidad de la toma de decisión de cada método propuesto y se grafican las curvas de Características Operativas del Receptor (curvas ROC) para representar todos los puntos de operación del umbral.

## 5.2. Descripción de los experimentos

### 5.2.1. Experimentos modo identificación

Los experimentos fueron realizados dividiendo la base de datos en tres partes:

1. Base de datos ficticia: Para crear una base de datos ficticia, de la base de datos Miles se tomaron 30% de los iris que son diferentes, mientras que para la base de datos UBIRIS Se tomaron el 75% de los iris diferentes. Se obtuvo el patrón de cada uno de ellos de acuerdo al método implementado.
2. Iris para umbrales: Se tomaron muestras para crear los umbrales para la aceptación o rechazo de una nueva muestra que llegue al sistema. Las muestras fueron un total de 0.04% de las muestras que fueron bien localizadas en la base de datos Miles y en UBIRIS se tomaron el 0.05%. De esas muestras, la mitad de ellas pertenecían a la base de datos y la otra mitad no pertenecían.
3. Iris para pruebas: Los iris restantes se usarán para representar las muestras que lleguen al sistema solicitando identificación.

Cada que se realizó un nuevo experimento se cambiaron las muestras de la base de datos, los umbrales y las pruebas.

Los experimentos muestran dos resultados debido a la forma en que fueron implementados:

- *Resultados prácticos:* son denominados a los experimentos que intentan apearse al modo operacional si este fuese implementado en la vida real, donde las muestras de los umbrales son independientes de las pruebas y no son mezcladas para mejorar los resultados presentados.
- *Resultados teóricos:* los umbrales y las pruebas se unen obteniendo el umbral que dé los mejores resultados para las muestras que se tienen.

Se debe aclarar que los resultados prácticos sólo darán la eficiencia máxima que pueden ser obtenidas por las muestras y el método implementado, mientras menor sea la diferencia entre los resultados prácticos y teóricos, mayor será la confiabilidad del método propuesto.

### 5.2.2. Experimentos modo verificación

Los experimentos se realizaron comparando un iris contra todos los de la base de datos. Los resultados presentados se dan con la decisión entre pares de iris iguales y pares de iris diferentes para obtener la curva de genuino e impostor. Al trazar las curvas se puede detectar la independencia de la prueba de verificación y la desviación estándar de cada distribución, proporcionando información sobre las calidad de las imágenes que pueden verse afectadas por deficiencias de iluminación, reflejos u otro tipo de ruido como lo menciona [11].

Para poder realizar la evaluación del desempeño se graficarán las curvas de Características Operativas del Receptor mejor conocidas como curvas ROC (Receiver Operating Characteristic). La curva mostrará la sensibilidad de cada método en función del umbral, graficando la proporción de falsos positivos contra falsos negativos. Las ventajas de analizar la eficiencia con una curva ROC es que muestra la ejecución global en un rango completo de umbral, logrando apreciar la sensibilidad del método analizado.

### 5.3. Bases de datos

#### 5.3.1. Base de datos Miles Research

La base de datos Miles Research la conforman imágenes de iris en formato JPG a color de 1024x768 píxeles, se cuentan con 332 muestras de iris de diferentes colores, las imágenes presentan variaciones de iluminación, obstrucciones como pestañas y tomadas en diferentes sesiones. Algunos ejemplos de las localizaciones fallidas se muestran en la figura (Ver figura 5.1). Uno de principales problemas que se pueden observar es la iluminación en el centro de la pupila, se presentan cuatro reflejos de luces los cuales impiden al correcta determinación del límite pupilar.

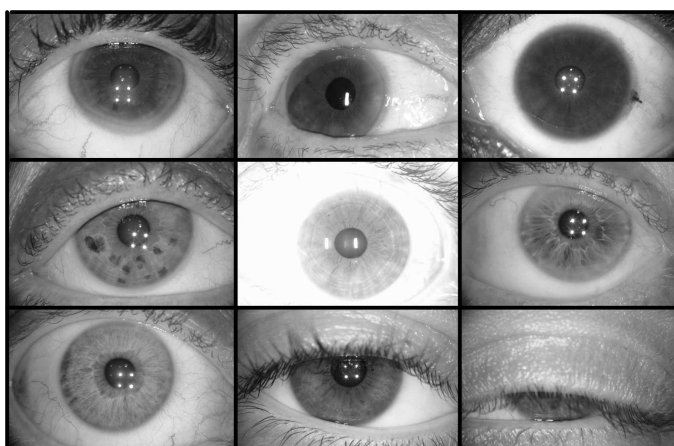


FIGURA 5.1: Ejemplos de localizaciones fallidas de la base de datos Miles Research.

Las pruebas realizadas arrojan que se localizaron 178 iris cuando las imágenes no fueron ecualizadas por histogramas, al llevar acabo el preprocesamiento el número de muestras que fueron localizadas correctamente fueron 213 es decir un 64.5% y se componen de 119 iris diferentes y donde al menos se cuenta con una muestra de cada una de ellos.

#### 5.3.2. Base de datos UBIRIS

Esta base de datos fue realizada por el Departamento de Informática de la Universidad de Beira de Portugal (UBIRIS)[36]. Se compone de 241 personas y fueron tomadas en dos sesiones, para los experimentos se usaron las imágenes de la sesión uno que se compone de 1213 iris.

Las imágenes de los iris fueron tomadas durante el mes de Septiembre del 2004 y a diferencia de otras bases de datos como la CASIA y la UPOL se incorporaron imágenes con algunos factores de ruido.

Las características presentadas por la base de datos son las siguientes:

- **Modelo de la cámara:** Nikon E5700.
- **Software:** E5700 v1.0
- **Representación del color:** RBG.
- **Longitud focal:** 71 mm.
- **Tiempo de exposición:** 1/30 segundos.
- **ISO speed:** ISO-200.
- **Tamaño de las imágenes:** 2560x1704 pixeles.
- **Resolución:** 300 dpi.
- **Formato:** JPEG.

Todas las imágenes fueron clasificadas por tres parámetros (“Enfoque”, “Reflejo” y “Visibilidad del iris”) y en una escala de tres valores (“Bueno”, “Regular” y “Malo”). La clasificación estadística es detallada en la tabla 5.1.

TABLA 5.1: Clasificación del ruido presentado por la base de datos UBIRIS

Parámetro	Bueno	Regular	Malo
<b>Enfoque</b>	73.83 %	17.53 %	8.63 %
<b>Reflejo</b>	58.87 %	36.78 %	4.34 %
<b>Visibilidad del iris</b>	36.73 %	47.83 %	15.44 %

Los creadores de la base de datos realizaron un preprocesamiento de las imágenes para que pudieran ser distribuidas en Internet y se redujeron las imágenes de tamaño a 800x600 teniendo aún la calidad suficiente para ejecutar los algoritmos.

El número de iris que fueron localizados correctamente sin realizar la ecualización de histogramas fueron 456 muestras, mientras que una vez ecualizadas se lograron obtener 1013 localizaciones exitosas que equivale a un 83.51 %. En la figura 5.2 se muestran algunos ejemplos de localizaciones fallidas debido a que los ojos estaban parcialmente abiertos, que estaban mal enfocadas o presentaban algún tipo de coloración en la esclera. De los 241 iris diferentes sólo fueron tomados 230 iris, donde al menos una de las muestras fue localizada con éxito.

### 5.3.3. Evaluación de la base de datos.

#### Ruidos presentados en la base de datos Miles Research

Las muestras totales de esta base de datos son 332 y fueron localizadas correctamente 213 muestras, es decir que más de la tercera parte fueron mal localizadas, la mayor parte de las imágenes presentan ruido por la iluminación, lo que indica que no fue un ambiente muy controlado el empleado para tomar las muestras.

1. **Iluminación:** Uno de los principales problemas de la base de datos radica en la iluminación con la que fueron tomadas, en la figura 5.3(a) se pueden observar cuatro ejemplos de iris. Para la iluminación de la pupila se presentan en algunas muestras 4 reflejos y en otras dos, estos reflejos no tan sólo se encuentran en la pupila, abarcan una pequeña área del iris y no se cuenta con algún patrón para delimitarlo en la etapa de normalización. El tipo de iluminación no es especificada en la base de datos, sin embargo muchos de los iris presentan dilatación de la pupila, si se dilata la pupila en gran medida se puede llegar a perder información importante del iris.



FIGURA 5.2: Ejemplos de localizaciones fallidas de la base de datos UBIRIS.

2. **Obstrucciones:** Este ruido es inevitable cuando el sistema es probado en modo real, el ser humano parpadea inevitablemente al momento de tomar la foto del iris y hay algunas imágenes que localizaron a pesar de esa obstrucción. En la figura 5.3 se muestra algunos ejemplos en donde los iris fueron localizados exitosamente.

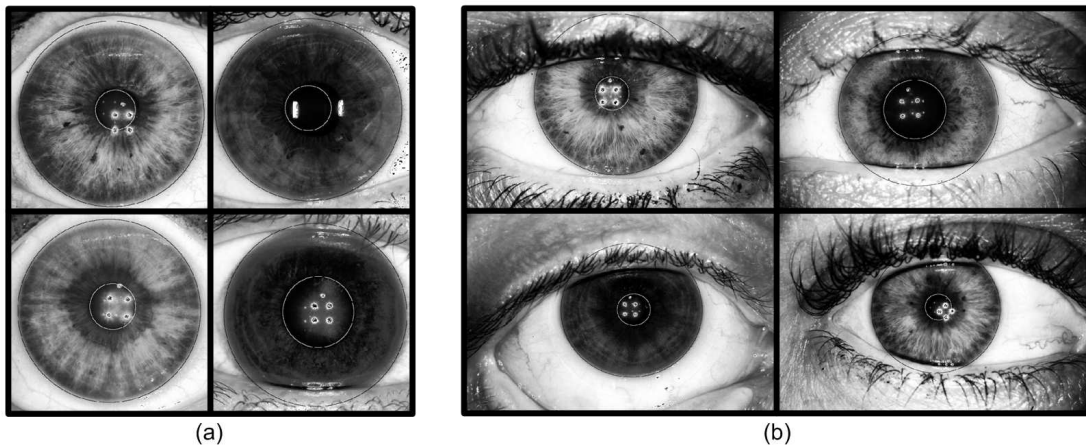


FIGURA 5.3: Ejemplos de iris con ruido de la base de datos Miles Research. a)Ejemplos con problemas de iluminación. b)Ejemplos con obstrucciones.

### Ruidos presentados en la base de datos UBIRIS

Los ruidos descritos en la tabla 5.1 se encuentran presentes en las imágenes de la base de datos localizadas con éxito y vuelven más difícil la aplicación exitosa de los métodos de reconocimiento.

1. **Enfoque:** la mayor parte de las imágenes que se encontraban desenfocadas fueron localizadas con éxito pero es uno de los ruidos que difícil de cuantificar, en la figura 5.4 (a) se muestran tres diferentes iris con tres grados de mal enfoque.
2. **Obstrucciones:** En la figura 5.4 (b) se muestra tres imágenes del mismo iris, donde el primero presenta el iris sin ninguna obstrucción y las dos siguientes imágenes con la obstrucción de los párpados.

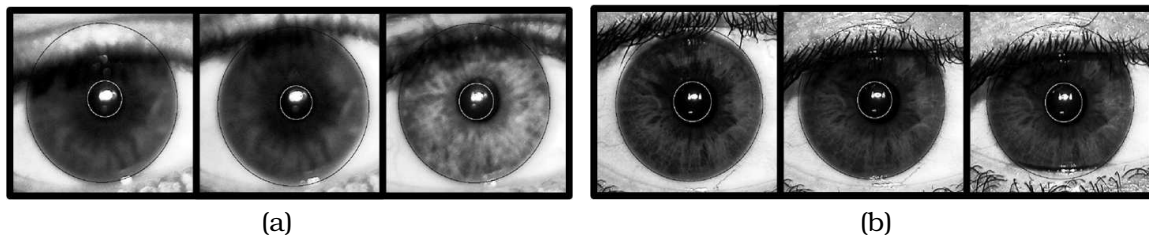


FIGURA 5.4: Ejemplos de iris con ruido de la base de datos UBIRIS. a) Ejemplos de imágenes desenfocadas. b) Ejemplos de un mismo iris con obstrucciones.

3. **Reflejo:** Otro ruido presentado es un reflejo mostrado en la parte inferior derecha del iris producto de la iluminación al momento de tomar la imagen. En la figura 5.5 se muestran cuatro diferentes iris que muestran el reflejo.

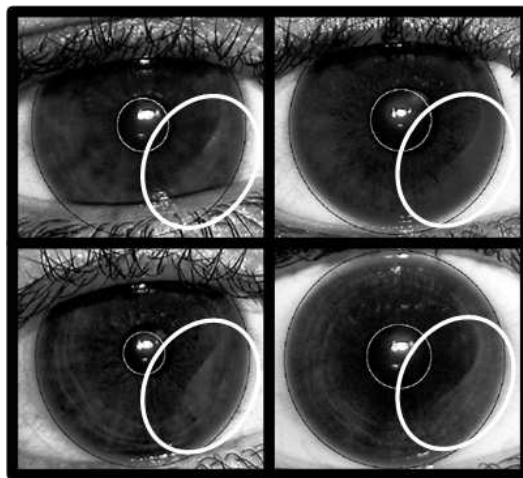


FIGURA 5.5: Ejemplos de iris con reflejo de la base de datos UBIRIS.

#### 5.3.4. Eliminación del ruido de las bases de datos

Debido a que el algoritmo no será implementado desde la adquisición de las imágenes no se tiene una etapa de evaluación del enfoque de las imágenes, sin embargo se usa una selección previa de la base de datos para eliminar las imágenes que presentan ruido ya sea por reflejos, mal enfoque o que el iris se encuentre parcialmente cerrado.

Para llevar a cabo esta evaluación sólo se realizará una comparación entre las muestras del mismo iris con el fin de no alterar la objetividad del método e interferir con las pruebas de evaluación.

La selección se realizó con las muestras del mismo iris mediante la comparación de las cintas por medio de la ecuación 5.1.

$$DM_{kl} = \sum_{j=1}^n \sum_{j=1}^m |I_k(i, j) - I_l(i, j)| \quad (5.1)$$

donde  $DM_{kl}$  es la diferencia del nivel de grises entre la cinta  $k$  y la cinta  $l$ ,  $I_k(i, j)$  y  $I_l(i, j)$  representa la imagen de la cinta  $k$  y  $l$  respectivamente,  $n$  es el número de renglones y  $m$  es el número de columnas de la cinta de la figura 4.7.

Los resultados de la evaluación son presentados en la tabla 5.2, se manejarán por porcentajes con respecto al total de iris que les fueron localizados correctamente los límites del iris y la pupila.

TABLA 5.2: Preselección de las bases de datos

<b>Base de datos Miles</b>		<b>Base de datos UBIRIS</b>	
Porcentaje	No. de iris	Porcentaje	No. de iris
100 %	213	100 %	1013
75 %	160	90 %	912
		80 %	811
		70 %	710
		60 %	608
		50 %	507

## 5.4. Resultados de los experimentos en modo identificación

En esta sección sólo se mostrará el resumen de los resultados obtenidos en los experimentos en modo verificación, los datos completos obtenidos en los experimentos se pueden consultar en el anexo A para la base de datos Miles Research y en el anexo B para la base de datos UBIRIS.

### 5.4.1. Condiciones experimentales para los métodos propuestos

A continuación se describen las condiciones en que se realizarán las pruebas para cada uno de los métodos y para cada una de las bases de datos.

#### Método de muestreo

Primero se realizaron pruebas para mostrar cuál de los tres muestreos presenta mejores resultados, dichos parámetros son el aleatorio simple, aleatorio sin reposición y Hammersley. Para cada base de datos pueden variarse tres parámetros:

1. Número de subimágenes: Es el total de zonas muestreadas sobre la cinta que representa al iris en coordenadas rectangulares.
2. Altura de las subimágenes.
3. Largo de las subimágenes.

Por cuestiones de simplificación se trabajó con subimágenes cuadradas, para poder variar dos parámetros. Fueron seleccionados tamaños de subimágenes de 5x5, 15x15 y 25x25 píxeles; el número de subimágenes variarán en 100, 150 y 200.

Debido a que la naturaleza aleatoria, cada combinación se realizará cinco veces y todos los resultados mostrarán el promedio de las pruebas.

### Método ARX y la optimización de sus parámetros por recocido simulado

Antes de realizar las pruebas del método ARX se optimizaron los  $na$ 's y  $nb$ 's teniendo como objetivo minimizar el valor de la ecuación 4.15. Los resultados de dicho experimento se muestran en la figura 5.6. Los factores de sintonización empleados fueron los mismos que los mostrados en la sección 5.4.3.

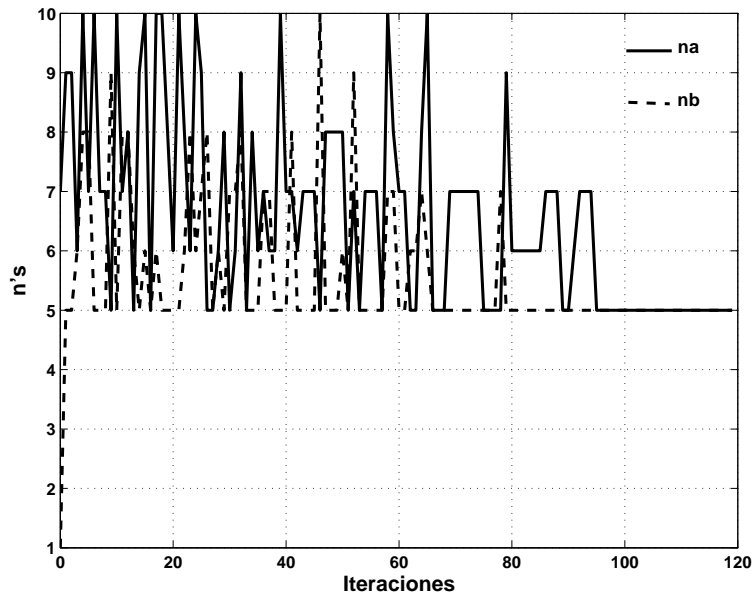


FIGURA 5.6: Optimización de los parámetros del método ARX por medio del algoritmo de recocido simulado.

Las condiciones para los experimentos serán con los valores de  $na = 5$  y  $nb = 5$  los cuales son sustituidos en la ecuación 4.6 que sirven para describir al modelo que representará al iris; los valores son los obtenidos por el método de recocido simulado que se muestra en la figura 5.6.

### Base de datos Miles Research

De acuerdo a la tabla 5.2 se tienen dos clasificaciones de la base de datos, la primera es de 100% la cual tiene todos los iris a los que les fue correctamente localizados los límites del iris y la pupila. De esa base de datos se realizó una selección de las muestras para quitar aquellas que tenían ruido obteniendo la base de datos de 75% la cual tiene 160 muestras de las 213 que tenía originalmente. En la tabla 5.3 se presentan el número de muestras empleadas en cada fase del experimento.

TABLA 5.3: Clasificación del número de iris ocupados en cada sección del experimento de la base de datos Miles Research.

Porcentaje	Base	Umbral	Pruebas
100%	36	10	167
75%	36	8	116

### Base de datos UBIRIS

Como se observa en la tabla 5.2 esta base de datos tiene 5 clasificaciones de acuerdo las muestras que fueron eliminadas por el ruido, es decir la base de datos al 100 % contiene todas las imágenes a las que les fueron localizados correctamente el límite límbico y pupilar y la base de datos del 50 % les fueron sustraídas las imágenes que presentaban mayor ruido y alteraban el funcionamiento de los métodos. El número de iris que se emplearon en la base ficticia de iris, para crear los umbrales y las pruebas se muestran en la tabla 5.4.

TABLA 5.4: Clasificación del número de iris ocupados en cada sección del experimento de la base de datos UBIRIS.

Porcentaje	Base	Umbral	Pruebas
100 %	173	51	789
90 %	173	46	693
80 %	173	41	597
70 %	173	36	501
60 %	173	31	404
50 %	173	26	308

#### 5.4.2. Resultados experimentales en modo identificación

En esta sección se muestran los mejores resultados de cada método, para cada base de datos serán graficados los datos obtenidos al comparar el método ARX y los tres métodos de muestreo: aleatorio con reposición o simple, aleatorio sin reposición y Hammersley. Para la base de datos UBIRIS se emplearán los métodos de optimización explicados en la sección 4.6 y serán graficados los resultados.

#### Resultados para la base de datos Miles Research

Al observar los resultados de la gráfica 5.7 se puede notar que hay mucha diferencia en los resultados obtenidos. En eje  $Y$  se grafica la efectividad de cada método y en el eje  $X$  se presenta la clasificación de cada base de datos de acuerdo a la tabla 5.3.

Los mejores resultados con la base de datos de 100 % son obtenidos por el muestreo hammerley con un 96.95 %, se puede apreciar que el método ARX es sin duda el que presenta el resultado más bajo. Los resultados mostrados para la base de datos se observa que los tres métodos de muestreo obtuvieron el 100 %, sin embargo el que logra mayores combinaciones con el 100 % de efectividad es nuevamente el muestreo Hammersley con ocho de nueve combinaciones posibles.

#### Resultados para la base de datos UBIRIS

Los resultados son mostrados de acuerdo a la clasificación de la tabla 5.4, las pruebas son mostradas para 100 %, 90 %, 80 % y 70 %, se omitieron las dos últimas porque los resultados eran muy próximos a 100 % de desempeño, lo que impide lograr una comparación entre los métodos propuestos.

De las primeras conclusiones es el bajo desempeño obtenido por el modelo ARX, sin embargo se aprecia que conforme la base de datos va siendo más limpia, la distancia entre efectividades con los métodos de muestreo va disminuyendo.



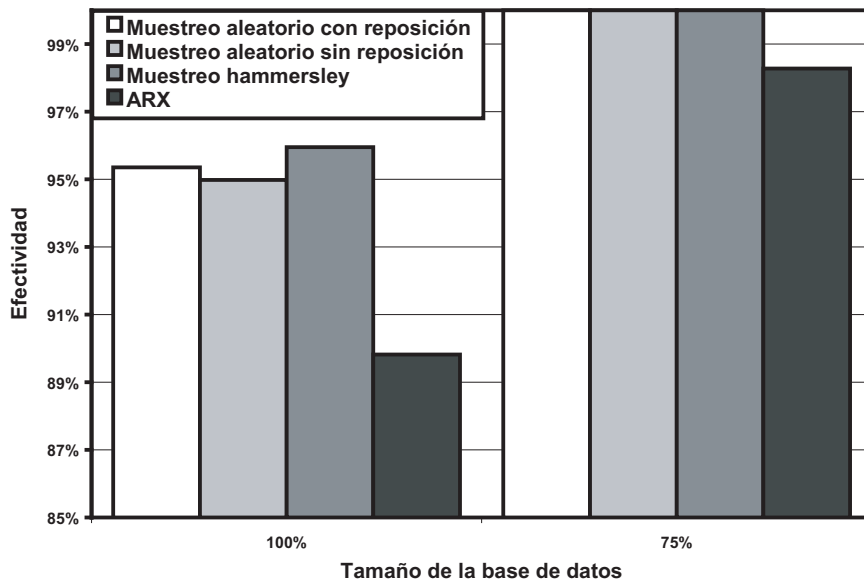


FIGURA 5.7: Resultados con mejores desempeños en cada método propuesto para la base de datos Miles Research, en el eje Y se grafica la efectividad y en el eje X el tamaño de la base de datos de acuerdo a la tabla 5.3

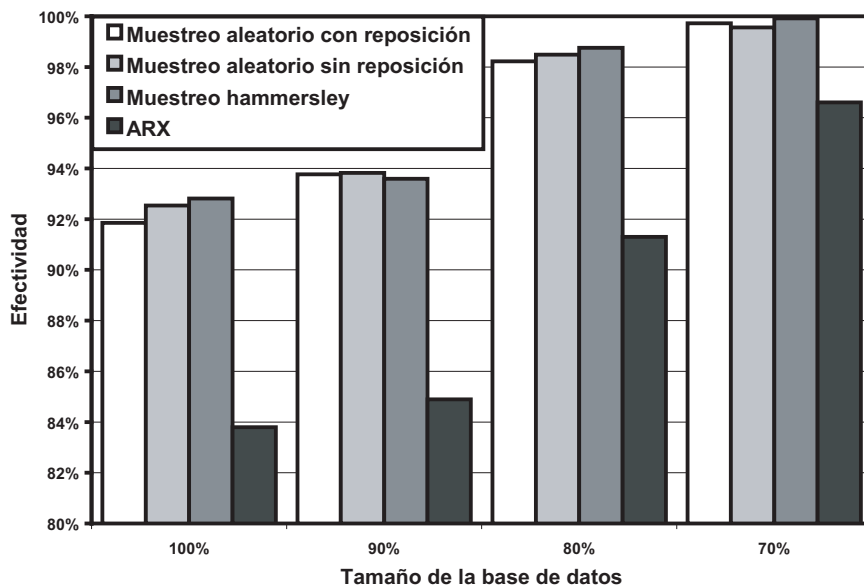


FIGURA 5.8: Resultados con mejores desempeños en cada método propuesto para la base de datos UBIRIS, en el eje Y se grafica la efectividad y en el eje X el tamaño de la base de datos de acuerdo a la tabla 5.4

En tres de las cuatro clasificaciones realizadas el muestreo de Hammersley es el que obtuvo mejores resultados, por lo tanto son comprobadas las suposiciones realizadas en la sección 4.6 donde se expuso que el muestreo hammerley lograba representar mejor las características del iris.

Sin embargo el propósito del trabajo de investigación es enfocarse a la base de datos del 100% que contiene las muestras con ruido y se tiene una efectividad de 92.82%.

### 5.4.3. Optimización del método de muestreo por recocido simulado

El método de muestreo al que se le buscarán los parámetros óptimos será el Hammersley, el cual dio los mejores resultados para ambas bases de datos. El método de recocido simulado fue ejecutado con la base de datos UBIRIS.

Debido a la naturaleza del recocido simulado cada que se realiza una iteración se debe realizar una nueva prueba con distintos parámetros, el realizar una corrida con todos los iris implicaría muchísimo tiempo computacional. Para efectos de la sintonización se trabajaron con 50 iris diferentes que comprendían un conjunto de 208 muestras.

Los factores que se varían al momento de realizar el recocido simulado son la temperatura modificando el factor de decrecimiento  $\alpha$ , la amplitud de la modificación en cada parámetro a optimizar y el número de iteraciones. Las combinaciones fueron resultado de los siguientes parámetros:

- Factor de decrecimiento de la temperatura ( $\alpha$ ): 0.5, 0.65, 0.8 y 0.95
- Amplitud del cambio de cada parámetro: Los parámetros a modificar son el número de subimágenes, la altura y el ancho de la subimagen sin embargo se asumió que todas los parámetros iban a modificarse con la misma amplitud. Los valores introducidos en la sintonización son: 5, 10 y 15 unidades.

El número de iteraciones fue debido al comportamiento de las gráficas, cada que se realizaban las experimentaciones en la mayoría de los casos se estabilizaba con las primeras 20 iteraciones. Por lo tanto se estableció que cada 50 iteraciones se reiniciarán los factores de la temperatura y los parámetros que se modifican en el muestreo Hammersley. Al reiniciar los parámetros el único cambio es que la inicialización de éstos son los óptimos de la primera corrida; la idea es que el algoritmo no quede atrapado en algún óptimo local. Algunos ejemplos de las corridas experimentales se muestran en las gráficas de la figura 5.9.

De estas pruebas se aprovecharon y se tomaron las ocho mejores combinaciones las cuales se presentan en la tabla 5.5.

TABLA 5.5: Combinaciones de parámetros de muestreo Hammersley obtenidos por recocido simulado con 50 iris diferentes.

Número de subimágenes	113	128	178	184	46	57	73	99
Altura de las subimágenes	15	29	11	14	21	31	21	5
Largo de las subimágenes	32	5	5	5	20	5	5	15

Al final las pruebas fueron realizadas 5 veces con 250 iteraciones. Los factores que sintonizaron al recocido simulado fue con un factor de decrecimiento de temperatura de 0.65 y la amplitud de cambio de 10 unidades en cada parámetro a optimizar. La figura 5.10 muestra la corrida final la cuál se hizo con toda la base de datos.

Las combinaciones obtenidas por el método de recocido simulado con toda la base de datos son mostradas en la tabla 5.6.

TABLA 5.6: Combinaciones de parámetros de muestreo Hammersley obtenidos por recocido simulado con toda la base de datos.

Número de subimágenes	169	150	141	172	127	156	149	115
Altura de las subimágenes	29	16	35	42	38	33	44	48
Largo de las subimágenes	98	5	13	5	5	5	16	13

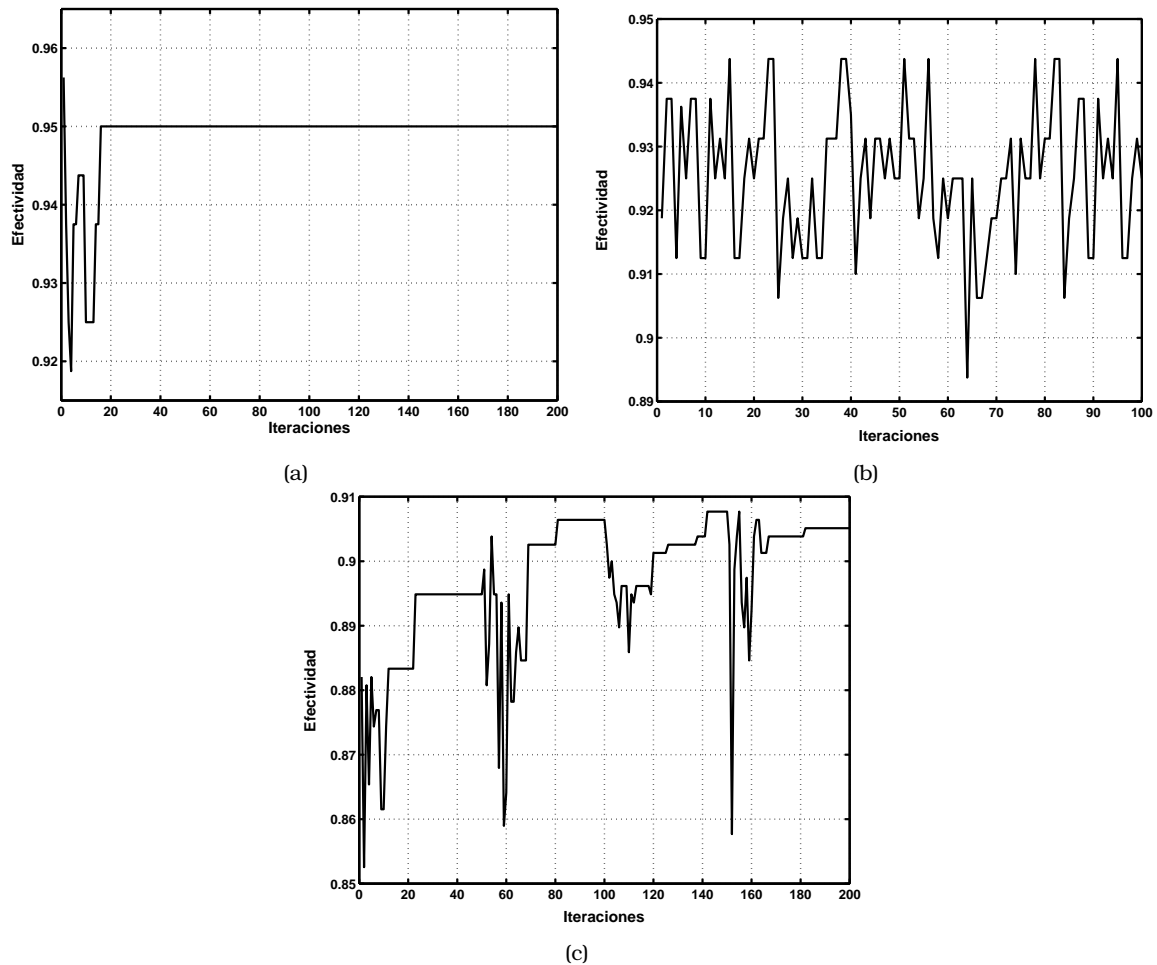


FIGURA 5.9: Ejemplos de los resultados de las corridas experimentales para encontrar los factores óptimos de recocido simulado. a) Sin reiniciar las iteraciones se puede observar que la efectividad se queda en un óptimo local. b) La gráfica ejemplifica la importancia del factor de decremento de la temperatura, el cual influye en que tan rápido llegue a la respuesta, sin embargo puede ocurrir este comportamiento donde no se logra la estabilidad de una respuesta óptima. c) En esta gráfica el comportamiento es el correcto, sin embargo la efectividad es muy baja.

Los resultados que se obtuvieron muestran las mejores combinaciones locales, más no se asegura que sea la mejor combinación global. Una vez obtenidos los parámetros óptimos se realizaron experimentos con las combinaciones de las tablas 5.5 y 5.6.

#### 5.4.4. Resultados del método de muestreo Hammersley optimizado por el algoritmo de recocido simulado

Los resultados mostrados en la figura 5.11 muestran los mejores cuatro resultados obtenidos por el algoritmo de recocido simulado. El algoritmo de recocido simulado fue implementado en el método de muestreo Hammersley y la explicación de la manera en que fue realizado el experimento se reporta en la sección 5.4.3.

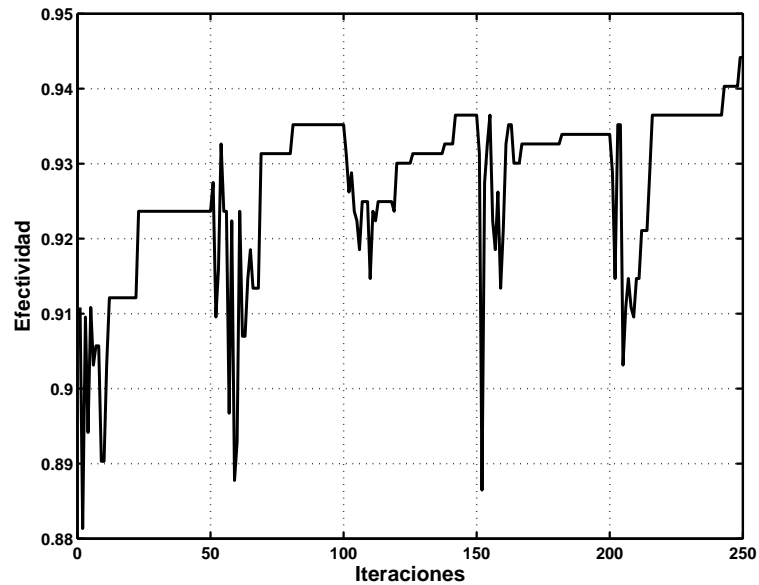


FIGURA 5.10: Resultados del algoritmo de recocido simulado con el método de muestreo Hammersley, se gráfica el número de iteraciones contra la efectividad. Para cada iteración se probó con las 1013 muestras que pertenecen a la base de datos y modificando cada uno de los parámetros.

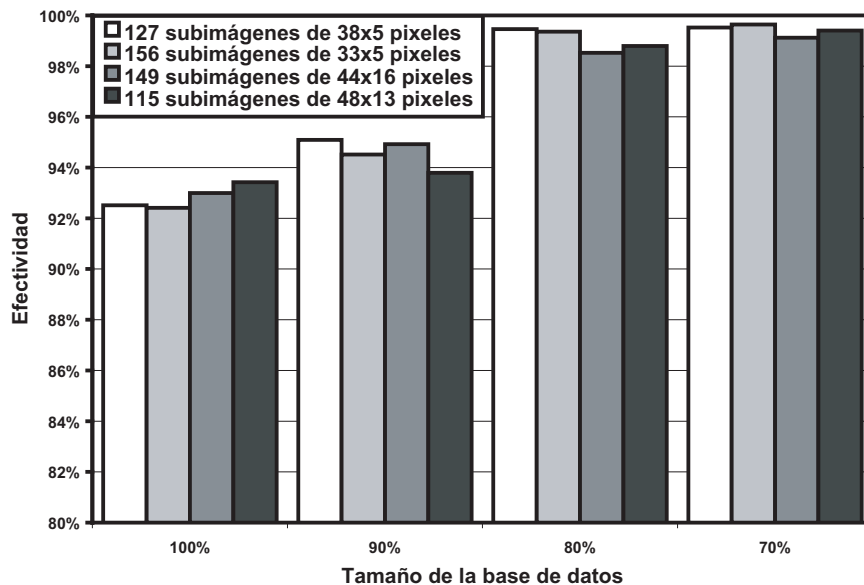


FIGURA 5.11: Resultados con mejores desempeños por el método de muestreo Hammersley optimizado con recocido simulado para la base de datos UBIRIS, en el eje Y se grafica la efectividad y en el eje X el tamaño de la base de datos de acuerdo a la tabla 5.4

Los resultados son muy parejos para todos los casos, la mejor combinación para la base de datos de 100% es de 115 subimágenes de 48x13 píxeles y se obtiene una efectividad de 93.43%. A comparación de los resultados mostrados en la figura 5.8 para la base de datos de 100% donde se obtiene una efectividad de 92.82%, el desempeño mejora pero apenas algunos puntos porcentuales.

### 5.4.5. Condiciones del método de muestreo Hammersley factorial

Este método como se explicó en la sección 4.6 consta de dos muestreos, en el primero se busca que el muestreo sea rápido sin perder el desempeño. Las combinaciones para el primer muestreo se obtuvieron de los parámetros optimizados por recocido simulado y los valores seleccionados fueron las cuatro combinaciones con el tiempo de identificación más pequeño; dichas combinaciones se encuentran en la tabla 5.7. El tiempo es tomado de todos los iris que fueron ocupados para las pruebas con la base del 100%.

TABLA 5.7: Combinaciones de parámetros de muestreo Hammersley obtenidos por recocido con los menores tiempos de identificación, los cuales serán empleados en el primer muestreo Hammersley factorial.

Primer muestreo	1	2	3	4
Número de subimágenes	46	57	73	99
Altura de las subimágenes	21	31	21	5
Largo de las subimágenes	20	5	5	15
Tiempo de identificación	159	178	207	309

Para los valores del segundo muestreo se emplearon los valores obtenidos en la tabla 5.6 debido a que en general son lo que mostraron mejores resultados.

TABLA 5.8: Combinaciones de parámetros de muestreo Hammersley obtenidos por recocido con los menores tiempos de identificación, los cuales serán empleados en el segundo muestreo Hammersley factorial.

Segundo muestreo	1	2	3	4	5	6	7	8
Número de subimágenes	169	150	141	172	127	156	149	115
Altura de las subimágenes	29	16	35	42	38	33	44	48
Largo de las subimágenes	98	5	13	5	5	5	16	13
Tiempo de identificación	383	693	357	458	301	516	506	368

### 5.4.6. Resultados del método de muestreo Hammersley factorial

Los últimos experimentos que fueron probados en el modo de identificación fueron los correspondientes al método de muestreo Hammersley factorial, para seleccionar los parámetros se emplearon los resultados obtenidos en el recocido simulado. Para el primer muestreo se buscó que fuera rápido y efectivo, su objetivo es eliminar las muestras de la base de datos que tienen diferencias más grandes con respecto a la que llegó al sistema. El segundo muestreo será más fino, por lo regular con mayor número de subimágenes y con un tiempo mayor de procesamiento; éste es quien da el veredicto final sobre el resultado de la identificación.

La figura 5.12 muestra los resultados del desempeño de todas las combinaciones que fueron sometidas a experimentación por el método de muestreo Hammersley factorial con la base de datos UBIRIS de 100%. La denotación en el eje  $X$  es por medio de dos números: (i,j) donde  $i$  es el primer muestreo de acuerdo a la tabla 5.7 y  $j$  es el segundo muestreo de acuerdo a la tabla 5.8.

Los resultados obtenidos son muy buenos, todas las combinaciones superan el 99.6% de efectividad. Los resultados más altos son de 99.92% arrojados por la combinación de un primer muestreo de 73 subimágenes de 21x5 pixeles, el segundo muestreo fue de 150 subimágenes de 16x5 pixeles y 141 subimágenes de 35x13 pixeles.

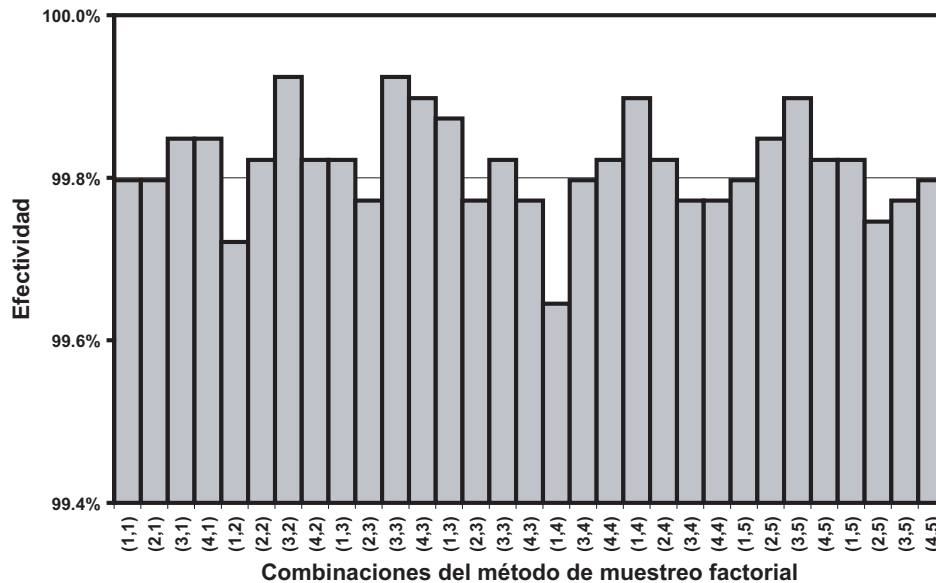


FIGURA 5.12: Resultados del muestreo Hammersley factorial con la base de datos UBIRIS de 100 %, en el eje Y se grafica la efectividad y en el eje X todas las combinaciones realizadas para los experimentos, se encuentra denotado por (i,j) donde i es el primer muestreo de acuerdo a la tabla 5.7 y j es el segundo muestreo de acuerdo a la tabla 5.8

## 5.5. Resultados experimentales en modo verificación

Sólo se mostrarán algunas gráficas de los resultados obtenidos, los datos de todos los experimentos se pueden consultar en el anexo C para la base de datos Miles Research y en el anexo D para la base de datos UBIRIS.

Por toda la información que puede ser generada las pruebas sólo fueron realizadas para la mejor respuesta obtenida por recocido simulado en modo identificación, 115 subimágenes de 48x13 pixeles de tamaño. Para el ARX también se emplea la mejor respuesta obtenida en la sección 5.4.

### 5.5.1. Resumen de los resultados en modo verificación

#### Base de datos Miles Research

Para presentar la eficiencia de cada método se grafico la respuesta de la efectividad de cada método de muestreo contra el umbral, en la figura 5.13 se muestran los resultados. Para el método ARX fue representado en otra gráfica debido a que el umbral tiene un diferente rango de operación, los resultados son mostrados en la figura 5.14

Para el método de muestreo con la base de datos del 100 % (figura 5.13.a) los mejores resultados son obtenidos por el método aleatorio con reposición, sin embargo cabe notar que la curva tiene pliegues por el número reducido de muestras que tiene esta base de datos.

Cuando se grafican la base de datos del 75 % (figura 5.13.b) los mayores resultados de efectividad son logrados por el muestreo Hammersley.

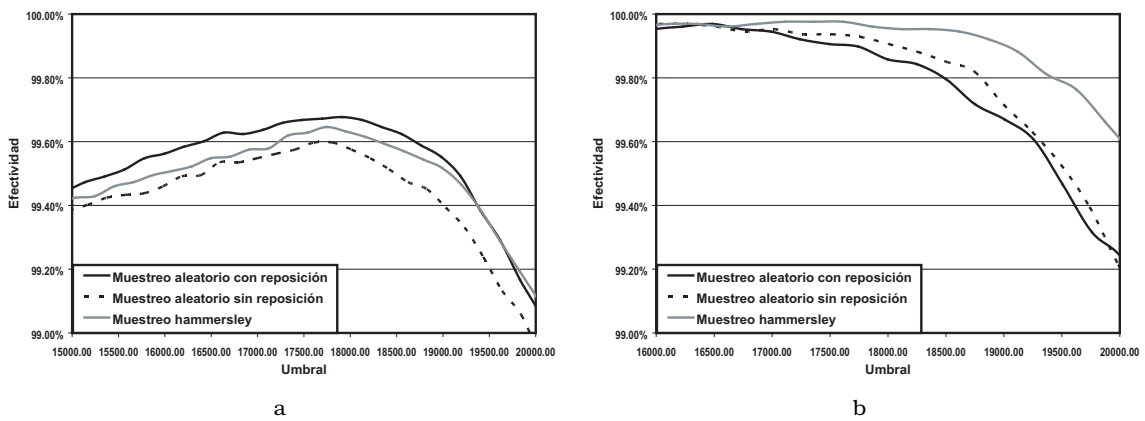


FIGURA 5.13: Efectividad por los métodos de muestreo en modo identificación variando el umbral. a) 100 % del tamaño de la base de datos. b) 75 % del tamaño de la base de datos.

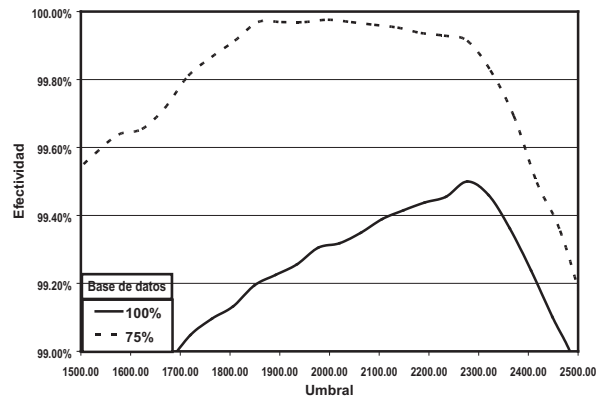


FIGURA 5.14: Efectividad variando el umbral por el método ARX y para las bases de datos de 100 % y 75 %.

En el método ARX se puede apreciar que para la base del 100 % no se logran tan buenos resultados como los mostrados en la figura 5.13.a, sin embargo para la base de datos del 100 % los resultados obtenidos son similares a los obtenidos por el método de muestreo en la figura 5.13.b.

### Base de datos UBIRIS

En la figura 5.15 se muestran los resultados para el método de muestreo variando el umbral y en la figura 5.14 son presentados los resultados del método ARX.

Para el método de muestreo con la base de datos del 100 % (figura 5.15.a) los mejores resultados son obtenidos por el método Hammersley, sin embargo se aprecia que los resultados obtenidos por el muestreo sin reposición son similares a los obtenidos por hammerley.

Cuando se grafican la base de datos del 60 % (figura 5.15.b) los mayores resultados de efectividad son logrados por el muestreo Hammersley y se observa una mayor diferencia que para la base de datos del 100 %.

En el método ARX reafirma el hecho de que al eliminar las muestras de ruido su respuesta se asemeja a la obtenida por el método de muestreo y se observa claramente el comparar la gráfica de la figura 5.16 contra las gráficas de la figura 5.15.

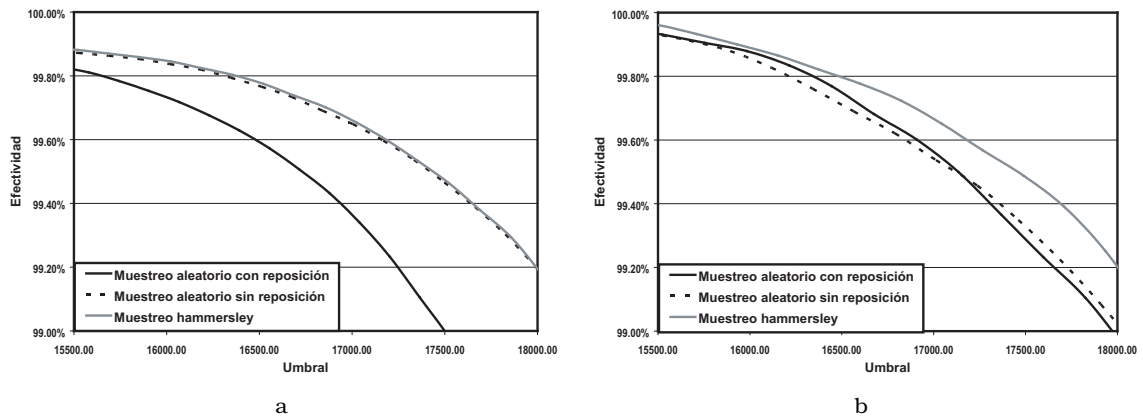


FIGURA 5.15: Efectividad por los métodos de muestreo en modo identificación variando el umbral. a) 100% del tamaño de la base de datos. b) 60% del tamaño de la base de datos.

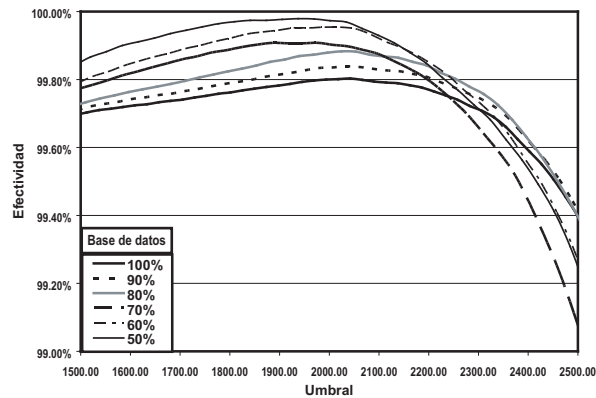


FIGURA 5.16: Efectividad variando el umbral por el método ARX y para las bases de datos de 100%, 90%, 80%, 70%, 60% y 50%.

### 5.5.2. Resultados en la Base de datos Miles Research

En general el comportamiento de las gráficas de distribución genuino-impostor van tomando una forma más ideal conforme se quitan las muestras de ruido como se muestra en la figura 5.17, es decir cuando se tiene mucho ruido las distribuciones se empalman y se tiene mayor probabilidad de cometer error; cuando se les quita esas muestras que presentan ruido las distribuciones impostor-genuino se separan haciendo un reconocimiento más confiable.

Otra característica que se muestra en la figura 5.17 las desviaciones estándar de las distribuciones van disminuyendo y van teniendo un comportamiento más similar a una distribución binomial.

La distribución impostor-genuino que cabe recalcar es la del método ARX mostrada en la figura 5.18 donde su gráfica de genuino presenta una menor desviación estándar en comparación de los métodos de muestreo, lo cual es una de las razones que en modo verificación allá fallado conforme las muestras presentaban mayor ruido.



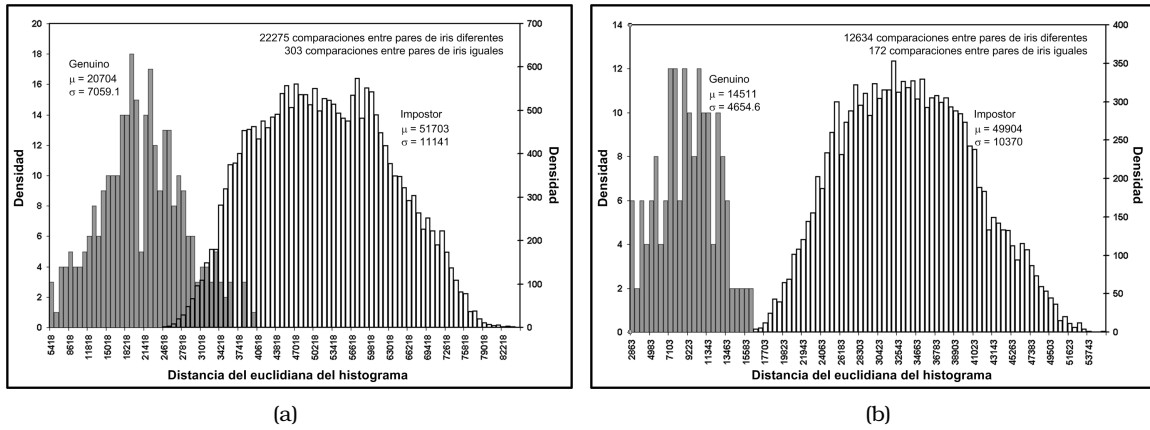


FIGURA 5.17: Distribución genuino-impositor por el método de muestreo Hammersley. a) Base de datos de tamaño 100 %. b) Base de datos de tamaño 75 %.

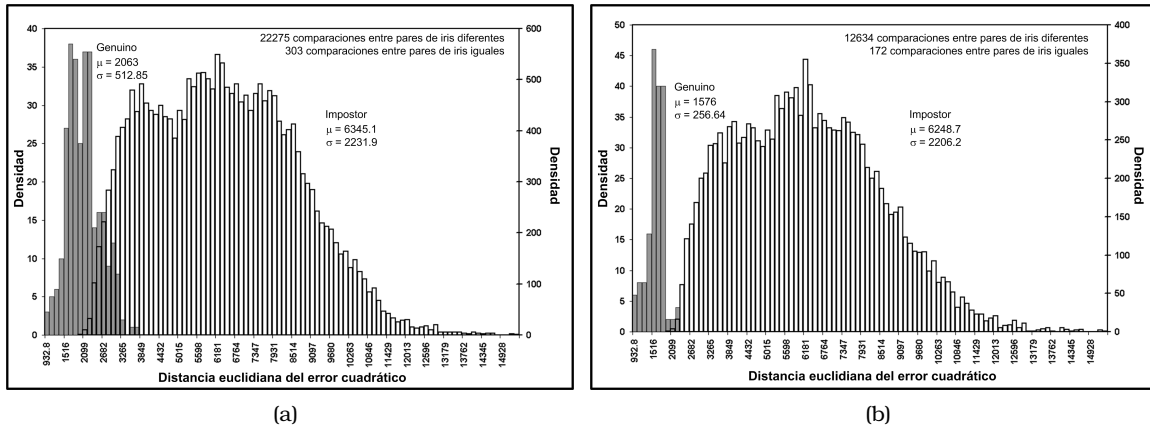


FIGURA 5.18: Distribución genuino-impositor por el método ARX. a) Base de datos de tamaño 100 %. b) Base de datos de tamaño 75 %.

De los métodos de muestreo presentados el que obtiene mejores resultados es el Hammersley debido a que su distribución se encuentra más uniforme que los otros muestreos y una manera más práctica de comparar a cada uno de los métodos es mediante una curva ROC. La curva ROC de la figura 5.19 (a) es para toda la base de dato sin eliminar las muestras de ruido, se puede observar la superioridad del método Hammersley; sin embargo cuando las muestras de iris que contienen ruido son eliminadas, la gráfica ROC de la figura 5.19 (b) muestra que los métodos son muy similares y casi no existen diferencias en su desempeño.

**5.5.3. Resultados en la Base de datos UBIRIS**

Al igual que en la Base de datos Miles Research, cuando se les quita las muestras que presentan ruido las distribuciones impostor-genuino se separan haciendo un reconocimiento más confiable.

Debido a que se tienen mas muestras se puede estudiar mejor el comportamiento de los parámetros que se observan en las distribuciones, como la media y desviación estándar de cada una de ellas. La figura 5.20 presenta el comportamiento de la diferencia entre las medias, la desviación estándar de la distribución del genuino y la del impostor.

De acuerdo a la figura 5.20 se puede apreciar que la diferencia entre la media del genuino y la media del impostor se van separando, es la razón por la cuál se obtiene mejores resultados.

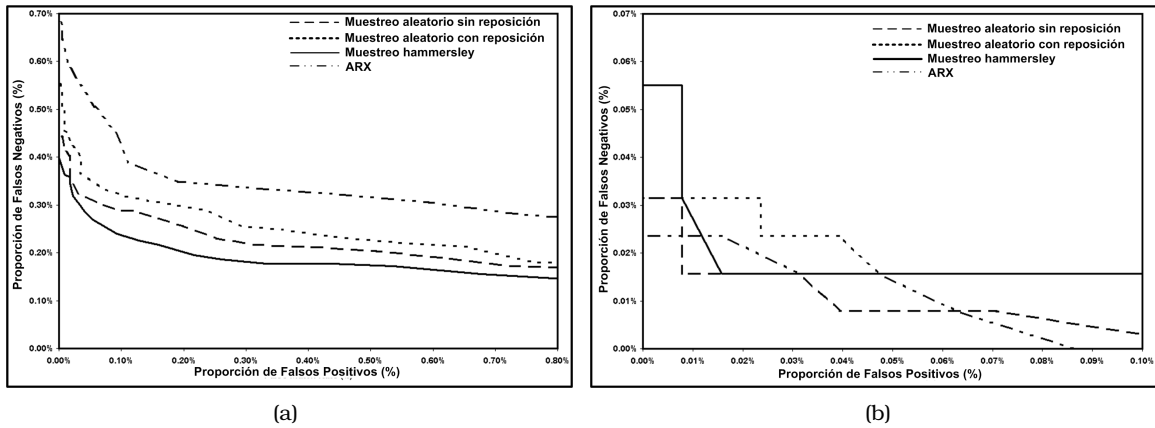


FIGURA 5.19: Curva ROC donde se comparan todos los métodos de verificación. a) Base de datos de tamaño 100 %. b) Base de datos de tamaño 75 %.

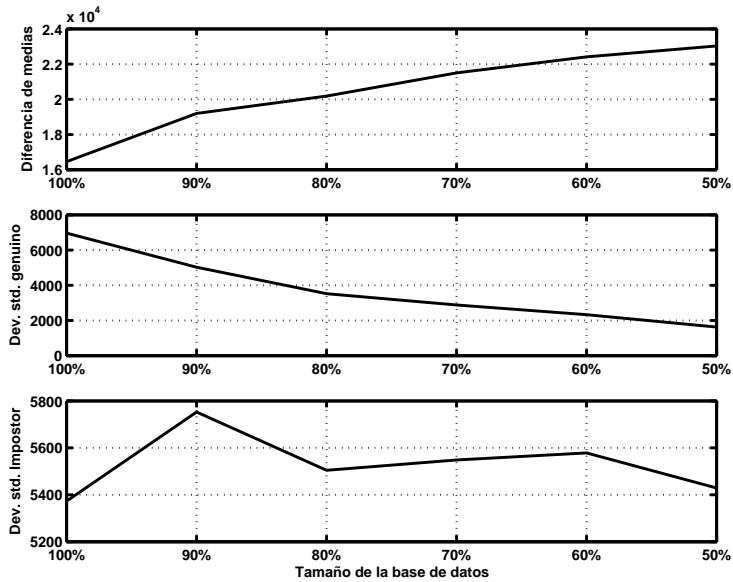


FIGURA 5.20: Comportamiento de los parámetros de las distribuciones genuino-impostor con respecto al tamaño de la base de datos.

En la gráfica de la desviación estándar del genuino se observa una pendiente muy pronunciada en el cambio de 100 % a 80 %, y se observa en los resultados que es cuando se tienen cambios más drásticos en los desempeños.

Por último la gráfica de la desviación estándar del impostor no presenta grandes variaciones, lo cual apoya el argumento de que al disminuir el tamaño de la base de datos sólo se están quitando las muestras que contienen ruido.

La gráfica de la figura 5.21 (a) muestra distribución genuino-impostor de la base de datos de 100 % y se observa claramente el empalme de las distribuciones, el seleccionar un umbral que de el mejor desempeño se caerían en muchos errores de falsos positivos y falsos negativos, haciendolo un método poco confiable por todo el ruido que presentan las muestras de los iris.

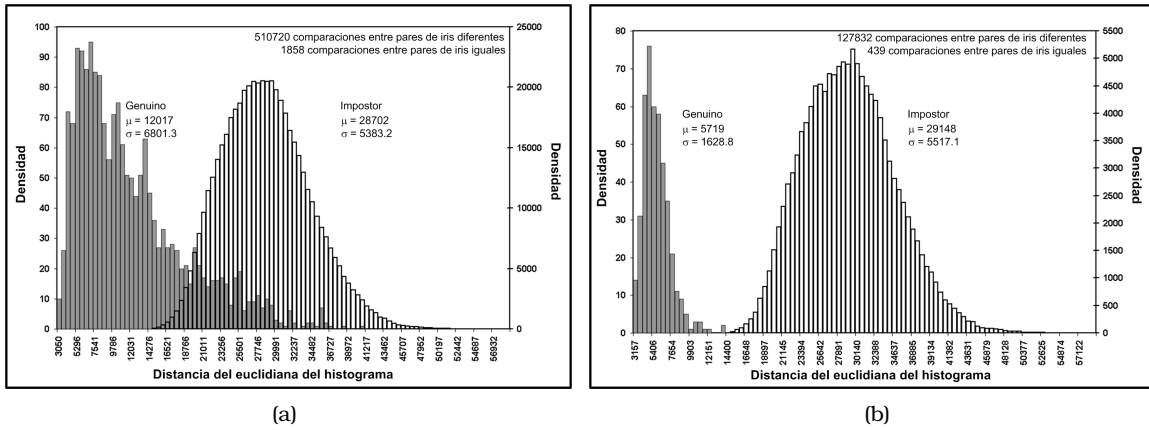


FIGURA 5.21: Distribución genuino-impostor por el método de muestreo Hammersley. a) Base de datos de tamaño 100 %. b) Base de datos de tamaño 50 %.

Las gráficas de distribución por si solas no permiten comparar la eficiencia entre los métodos propuestos, la comparación no es tan directa como en verificación. Para las curvas ROC se compararon en la figura 5.22 donde la gráfica (a) es para la base de 100 % y en la (b) se muestra su extremo de 60 %. No se grafica la de 50 % debido que para el muestreo aleatorio sin reposición y Hammersley la respuesta se encuentra pegada al eje  $x$  y  $y$ .

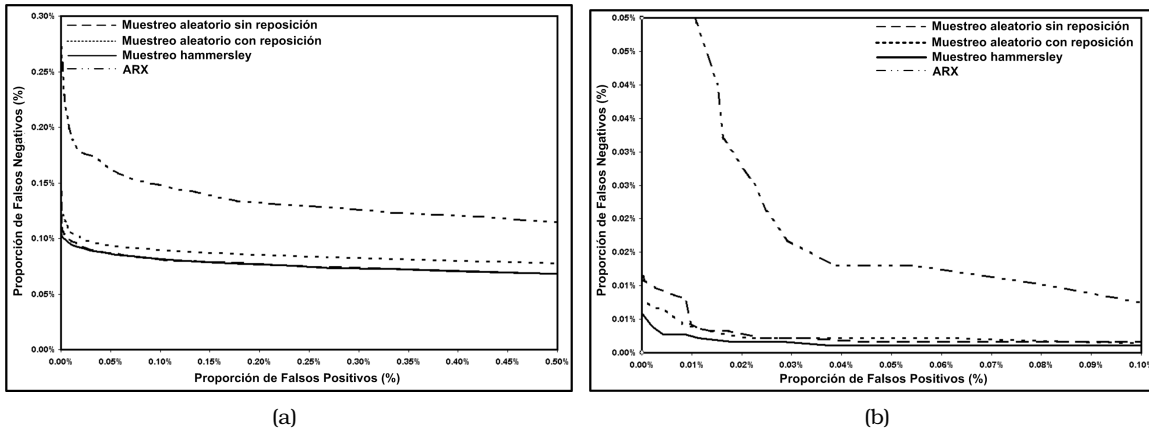


FIGURA 5.22: Curva ROC donde se comparan todos los métodos de verificación. a) Base de datos de tamaño 100 %. b) Base de datos de tamaño 90 %.

# Conclusiones y trabajos a futuro

---

## 6.1. Introducción

Como resultados de este trabajo de investigación, basados en el capítulo 1 donde se plantearon los objetivos de la investigación y en el capítulo 5 donde se muestran los resultados obtenidos del mismo se puede concluir que los objetivos del trabajo de investigación fueron cubiertos: dos métodos para el reconocimiento automático del iris y se logró diseñar e implementar cada idea propuesta obteniendo resultados muy prometedores. Al final se obtuvo un diseño de un nuevo sistema que extrae y compara las características del iris que es robusto ante el ruido presentado por una deficiente iluminación, desenfoque y/o reflejos.

## 6.2. Resumen de los resultados del trabajo de investigación

Este trabajo de investigación ha presentado un sistema de reconocimiento automático de personas basado en el análisis del iris usando métodos estadísticos. Para probar los resultados se emplearon dos bases de datos: Miles Research [37] y UBIRIS [36].

En la etapa de localización de los límites límbico y pupilar fue presentado un algoritmo que empleó la ecualización por histograma para modificar el contraste de la imagen donde se busca que todos los valores de grises tengan el mismo número de píxeles, es decir que la imagen contenga el mismo número de píxeles claros y oscuros. El objetivo de implementar la ecualización por histograma es para contrarrestar las deficiencias de iluminación que puedan presentarse en un diseño funcional. Para llevar a cabo la segmentación del iris se utilizó el operador integro-diferencial de Daugman [9],[8],[11] obteniendo resultados aceptables.

En la base de datos Miles Research fueron localizados exitosamente un 64.5%, cabe mencionar que muchas de las muestras se encontraban con los ojos parcialmente cerrados o con severos problemas de iluminación como fue explicado en la sección 5.3.1. Los resultados para la base de datos UBIRIS fue de un 83.51% de localizaciones exitosas. De acuerdo a [36] se realizó un experimento para obtener la exactitud de la segmentación de varios métodos con la base de datos UBIRIS; los resultados de la segmentación de esta investigación son inferiores al obtenido por el algoritmo de Daugman, el cual tuvo un 93.53% de localizaciones exitosas, sin embargo se obtienen resultados similares a los otros métodos probados.

La normalización se implementó totalmente del algoritmo de Daugman [9],[8],[11]. Como resultado se obtiene que todos los iris son representados por una imagen de 200X720 píxeles, la única implementación en esta etapa es el postprocesamiento a la imagen, nuevamente se emplea ecualización por histogramas para resaltar las características del iris y obtener mejores resultados en la etapa de extracción de características.

Las aportaciones importantes fueron realizadas en la tercera etapa, que es la extracción de características. Fueron presentadas dos propuestas: representar los rasgos del iris por un método de muestreo estadístico y por medio de un modelo estocástico, como lo es el ARX.

Para el método de muestreo se realizaron varias aportaciones, de la imagen normalizada se toman subimágenes de tamaño  $n \times m$  donde cada subimagen representará su histograma acumulado. Para muestrear la imagen normalizada que representa al iris se probaron tres métodos:

1. **Muestreo Aleatorio con Reposición:** La idea principal es garantizar que todos los elementos de la imagen tengan la misma oportunidad de ser incluidos en el subconjunto muestreado.
2. **Muestreo Aleatorio sin Reposición:** Se hace una modificación al concepto original, una vez que es muestreada una subimagen, ninguno de sus elementos podrá ser muestreado nuevamente para evitar información innecesaria.
3. **Muestreo Hammersley [14],[1]:** Es un método derivado de los métodos de cuasi-Monte Carlo, los cuales emplean un algoritmo determinístico para generar sus números pseudo-aleatorios ofreciendo un grado de uniformidad más alta.

Los resultados mostrados en la sección 5.4 comprueban que los mejores resultados los obtiene el muestreo Hammersley por las características con que es generado; este muestreo logra capturar las principales características realizando un muestreo muy uniforme a diferencia de los otros dos métodos.

Sin embargo los modelos fueron comparados con parámetros arbitrarios para su desempeño y se introdujo la idea de implementar el algoritmo de optimización de recocido simulado con el método de muestreo Hammersley; el objetivo fue maximizar el desempeño modificando el número de subimágenes y el tamaño de las mismas.

Los resultados obtenidos por recocido simulado fueron los esperados, se logró mejorar el desempeño teniendo varias combinaciones que presentan resultados similares.

Como última herramienta propuesta se implementó el método de muestreo factorial, la idea principal de este método es disminuir el tiempo de búsqueda en modo identificación realizando dos muestreos: el primer muestreo tiene la característica de ser rápido y efectivo, rechazando las muestras de la base de datos que definitivamente no pueden ser candidatas a ser comparadas con el iris que llegó al sistema; el segundo muestreo es más fino, siendo un muestreo más lento pero más efectivo. Los valores empleados para el muestreo factorial son los obtenidos de la optimización por recocido simulado. Sin duda los resultados mejoraron en gran medida, las combinaciones propuestas superan el 99.7% de efectividad y el mejor desempeño fue de 99.92% ocupando la base de datos con todas las muestras que fueron correctamente localizadas.

En el método ARX la representación del iris es modelada por este método estocástico. El recocido simulado fue empleado para optimizar los valores de  $n$ 's y  $m$ 's que representaran mejor al iris. Los resultados obtenidos no fueron tan buenos como el método de muestreo, sin embargo cuando la base de datos no muestra ruido los resultados pueden ser comparados por los obtenidos en el método de muestreo.

### 6.3. Sugerencias de investigaciones futuras

El sistema presentado en esta investigación aún áreas de mejora:

- En la etapa de adquisición de datos puede realizarse una parametrización de los iris de acuerdo a la raza. Esto ayudaría a no pasar al iris por las siguientes fases o bien desechar desde un inicio muestras de la base de datos que no pueden ser idénticas.
- La etapa de localización de límites límbico y pupilar aún presenta resultados pobres de 64.5% para la base de datos Miles Research y 83.51% para la base de datos UBIRIS, pueden mejorarse las localizaciones incorporando alguna técnica de preprocesamiento o alguna combinación con el operador integro-diferencial como lo es la transformada de Hough usada por [28]. La implementación de este algoritmo es demasiado lenta por lo que puede buscarse un algoritmo que se ejecute en un menor tiempo con mejores resultados.
- En la etapa de normalización sería interesante incorporar un algoritmo que eliminara zonas que presentará ruido. Una idea en la que se puede profundizar es en el análisis estadístico de texturas de primer y segundo orden, al observar una imagen con ruido la pestaña y/o párpados tienen la característica de que presentan una textura homogénea delimitada por una línea oscura.
- Una vez localizados el iris y la pupila en la imagen, implementar algún algoritmo que sea capaz de seleccionar entre las imágenes que fueron correctamente segmentadas.
- Para el método de muestreo se empleó la caracterización de la imagen por medio de su histograma acumulado; la manera en que se concluyó la elección de ese método fue después de probar todos los estadísticos de primer orden: media, desviación estándar, curtosis y entropía. La desventaja de representar cada subimagen con un histograma acumulado es la información que tiene que ser almacenada, por lo que puede implementarse alguna otra propiedad para representarse.
- Para el método ARX aún pueden realizarse muchas mejoras, en cuanto a tiempos y resultados; ya no se investigó a profundidad modelar el iris variando el factor de olvido, implementando variables instrumentales o modelarlo como un ARMAX.
- Las implementaciones de los programas fueron realizadas en MATLAB los cuales afectan mucho los tiempos de ejecución de cada etapa; se sugiere programarlo en C para mejorarlos.



# Resultados de la base de datos Miles Research (Identificación)

---

## Condiciones experimentales para los métodos de muestreo

Los resultados presentados en el anexo fueron probados con subimágenes de 5x5, 15x15 y 25x25 píxeles; el número de subimágenes variarán en 100, 150 y 200.

Debido a que la naturaleza aleatoria, cada combinación se realizará cinco veces y todos los resultados mostrarán el promedio de las pruebas.

## Condiciones experimentales para el modelo ARX

Las condiciones para los experimentos serán con los valores de  $na = 5$  y  $nb = 5$  los cuales son sustituidos en la ecuación 4.6 que sirven para describir al modelo que representará al iris; los valores son los obtenidos por el método de recocido simulado que se muestra en la figura 5.6.

## Descripción de la base de datos

De acuerdo a la tabla 5.2 se tienen dos clasificaciones de la base de datos: la primera es de 100% la cual tiene todos los iris que fueron correctamente segmentados y la de 75% que les fueron eliminadas muestras con ruido. En la tabla A.1 se presentan el número de muestras empleadas en cada fase del experimento.

TABLA A.1: Clasificación del número de iris ocupados en cada sección del experimento de la base de datos Miles Research.

Porcentaje	Base	Umbral	Pruebas
100 %	36	10	167
75 %	36	8	116



## A.1. Método de muestreo aleatorio con reposición

Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos. **PVN:** Proporción de Verdaderos Negativos.  
**PPF:** Proporción de Falsos Positivos. **PFN:** Proporción de Falsos Positivos.

TABLA A.2: Resultados por el método de muestreo aleatorio con reposición para 100% y 75% del tamaño de la base de datos

Tamaño de la subimagen	Tamaño de la base de datos Miles Research											
	100 subimágenes				150 subimágenes				200 subimágenes			
	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)
Práctico	PVP:	12.88%	11.31%	10.40%	8.87%	12.14%	12.98%	12.42%	11.68%	16.43%		
	PVN:	78.16%	80.12%	71.19%	79.78%	80.60%	82.37%	73.03%	83.31%	76.79%		
	PPF:	5.26%	7.02%	16.15%	8.59%	5.36%	2.14%	12.17%	1.91%	5.36%		
	PFN:	3.70%	1.55%	2.27%	2.76%	1.90%	2.50%	2.39%	3.10%	1.43%		
Teórico	Efectividad:	91.05%	91.43%	81.59%	88.64%	92.74%	95.36%	85.45%	95.00%	93.21%		
	Desv. Std:	5.03%	2.37%	12.33%	7.23%	6.63%	2.36%	9.65%	2.68%	9.06%		
	PVP:	15.15%	12.32%	12.35%	10.79%	13.82%	14.92%	12.99%	13.82%	17.27%		
	PVN:	79.43%	83.84%	84.15%	85.71%	83.47%	82.71%	82.72%	82.80%	80.13%		
Práctico	PVP:	6.52%	7.52%	7.71%	5.51%	6.87%	5.65%	6.67%	3.09%	8.75%		
	PVN:	92.97%	90.43%	91.25%	94.14%	92.62%	94.18%	93.33%	96.91%	91.25%		
	PPF:	0.00%	2.05%	0.34%	0.34%	0.17%	0.17%	0.00%	0.00%	0.00%		
	PFN:	0.52%	0.00%	0.69%	0.00%	0.34%	0.00%	0.00%	0.00%	0.00%		
Teórico	Efectividad:	99.48%	97.95%	98.97%	99.66%	99.49%	99.83%	100.00%	100.00%	100.00%		
	Desv. Std:	0.77%	2.46%	1.42%	0.77%	0.47%	0.38%	0.00%	0.00%	0.00%		
	PVP:	9.84%	10.32%	11.13%	8.39%	10.00%	8.55%	9.52%	6.13%	11.45%		
	PVN:	90.16%	89.68%	88.87%	91.61%	90.00%	91.45%	90.48%	93.87%	88.55%		
Práctico	PVP:	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%		
	PVN:	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%		
	PPF:	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%		
	PFN:	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%		
Teórico	Efectividad:	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%		
	Desv. Std:	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%		
	PVP:	10.369	11.28	13.021	18.295	18.473	20.493	24.922	29.575	27.455		
	PVN:	10.369	11.28	13.021	18.295	18.473	20.493	24.922	29.575	27.455		

## A.2. Método de muestreo aleatorio sin reposición

Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos. **PVN:** Proporción de Verdaderos Negativos.  
**PFP:** Proporción de Falsos Positivos. **PFN:** Proporción de Falsos Positivos.

TABLA A.3: Resultados por el método de muestreo aleatorio sin reposición para 100 % y 75 % del tamaño de la base de datos

Tamaño de la subimagen	Tamaño de la base de datos Miles Research											
	100 subimágenes				150 subimágenes				200 subimágenes			
	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)		5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)		5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	
Práctico	PVP:	8.71%	16.12%	14.43%	9.06%	11.60%	13.69%	11.81%	12.29%	11.22%		
	PVN:	81.03%	74.58%	79.72%	80.10%	82.07%	74.76%	73.76%	80.67%	83.77%		
	PFP:	5.13%	6.80%	3.69%	7.02%	3.82%	10.24%	13.24%	4.54%	3.47%		
	PFN:	5.13%	2.51%	2.15%	3.82%	2.51%	1.31%	1.20%	2.50%	1.55%		
Teórico	Efectividad:	89.74%	90.70%	94.16%	89.16%	93.67%	88.45%	85.57%	92.96%	94.99%		
	Desv. Std:	4.33%	4.65%	2.98%	5.82%	5.12%	9.42%	4.37%	5.29%	3.98%		
	PVP:	10.76%	17.85%	16.07%	10.55%	13.53%	14.92%	11.33%	14.58%	12.90%		
	PVN:	83.37%	79.10%	81.45%	84.48%	83.53%	82.60%	84.49%	83.05%	84.39%		
Práctico	PFP:	0.70%	0.45%	0.02%	0.61%	0.31%	0.34%	0.37%	0.11%	0.70%		
	PFN:	5.17%	2.60%	2.46%	4.37%	2.63%	2.15%	3.81%	2.26%	2.02%		
	Efectividad:	94.13%	96.95%	97.52%	95.03%	97.06%	97.52%	95.82%	97.63%	97.29%		
	Desv. Std:	1.17%	1.81%	1.10%	3.28%	1.35%	0.65%	0.85%	1.57%	2.44%		
Teórico	Tiempo (s)	13.345	14.018	16.528	21.601	23.252	35.827	32.374	34.131	32.556		
	PVP:	5.13%	8.24%	5.31%	7.89%	7.18%	7.69%	8.93%	4.62%	5.13%		
	PVN:	94.19%	91.41%	94.52%	91.60%	92.82%	92.31%	90.56%	95.39%	94.87%		
	PFP:	0.68%	0.34%	0.00%	0.51%	0.00%	0.00%	0.34%	0.00%	0.00%		
Práctico	PFN:	0.00%	0.00%	0.17%	0.00%	0.00%	0.00%	0.17%	0.00%	0.00%		
	Efectividad:	99.32%	99.66%	99.83%	99.49%	100.00%	100.00%	99.48%	100.00%	100.00%		
	Desv. Std:	1.53%	0.47%	0.38%	0.76%	0.00%	0.00%	0.77%	0.00%	0.00%		
	PVP:	8.06%	10.97%	8.39%	10.65%	10.00%	10.48%	11.77%	7.58%	8.06%		
Teórico	PVN:	91.94%	89.03%	91.61%	89.36%	90.00%	89.52%	88.23%	92.42%	91.94%		
	PFP:	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%		
	PFN:	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%		
	Efectividad:	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%		
Teórico	Desv. Std:	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%		
	Tiempo (s)	10.337	11.292	11.863	17.455	18.479	18.567	29.751	30.839	27.914		

### A.3. Método de muestreo Hammerlsey

Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos.    **PVN:** Proporción de Verdaderos Negativos.  
**PPF:** Proporción de Falsos Positivos.    **PFN:** Proporción de Falsos Positivos.

TABLA A.4: Resultados por el método de muestreo hammersley para 100 % y 75 % del tamaño de la base de datos

Tamaño de la subimagen	Tamaño de la base de datos Miles Research														
	100 subimágenes			150 subimágenes			200 subimágenes			100%			75%		
	5x5 (pixeles)	15x15 (pixeles)	25x25 (pixeles)	5x5 (pixeles)	15x15 (pixeles)	25x25 (pixeles)	5x5 (pixeles)	15x15 (pixeles)	25x25 (pixeles)	5x5 (pixeles)	15x15 (pixeles)	25x25 (pixeles)	5x5 (pixeles)	15x15 (pixeles)	25x25 (pixeles)
Práctico	PVP:	8.71%	16.12%	14.43%	9.06%	11.60%	13.69%	11.81%	12.29%	11.22%					
	PVN:	81.03%	74.58%	79.72%	80.10%	82.07%	74.76%	73.76%	80.67%	83.77%					
	PPF:	5.13%	6.80%	3.69%	7.02%	3.82%	10.24%	13.24%	4.54%	3.47%					
	PFN:	5.13%	2.51%	2.15%	3.82%	2.51%	1.31%	1.20%	2.50%	1.55%					
	Efectividad:	89.74%	90.70%	94.16%	89.16%	93.67%	88.45%	85.57%	92.96%	94.99%					
Teórico	Desv. Std:	4.33%	4.65%	2.98%	5.82%	5.12%	9.42%	4.37%	5.29%	3.98%					
	PVP:	10.76%	17.85%	16.07%	10.55%	13.53%	14.92%	11.33%	14.58%	12.90%					
	PVN:	83.37%	79.10%	81.45%	84.48%	83.53%	82.60%	84.49%	83.05%	84.39%					
	PPF:	0.70%	0.45%	0.02%	0.61%	0.31%	0.34%	0.37%	0.11%	0.70%					
	PFN:	5.17%	2.60%	2.46%	4.37%	2.63%	2.15%	3.81%	2.26%	2.02%					
Práctico	Efectividad:	94.13%	96.95%	97.52%	95.03%	97.06%	97.52%	95.82%	97.63%	97.29%					
	Desv. Std:	1.17%	1.81%	1.10%	3.28%	1.35%	0.65%	0.85%	1.57%	2.44%					
	Tiempo (s)	13.345	14.018	16.528	21.601	23.252	36.827	32.374	34.131	32.556					
	PVP:	5.13%	8.24%	5.31%	7.89%	7.18%	7.69%	8.93%	4.62%	5.13%					
	PVN:	94.19%	91.41%	94.52%	91.60%	92.82%	92.31%	90.56%	95.39%	94.87%					
Teórico	PPF:	0.68%	0.34%	0.00%	0.51%	0.00%	0.00%	0.34%	0.00%	0.00%					
	PFN:	0.00%	0.00%	0.17%	0.00%	0.00%	0.00%	0.17%	0.00%	0.00%					
	Efectividad:	99.32%	99.66%	99.83%	99.49%	100.00%	100.00%	99.48%	100.00%	100.00%					
	Desv. Std:	1.53%	0.47%	0.38%	0.76%	0.00%	0.00%	0.77%	0.00%	0.00%					
	PVP:	8.06%	10.97%	8.39%	10.65%	10.00%	10.48%	11.77%	7.58%	8.06%					
Práctico	PVN:	91.94%	89.03%	91.61%	89.36%	90.00%	89.52%	88.23%	92.42%	91.94%					
	PPF:	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%					
	PFN:	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%					
	Efectividad:	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%					
	Desv. Std:	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%					
Teórico	Tiempo (s)	10.337	11.292	11.853	17.455	18.479	18.567	29.751	30.839	27.914					

### A.4. Método de muestreo hammesley con parámetros optimizados por el método de recocido simulado

Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos.    **PVN:** Proporción de Verdaderos Negativos.  
**PFPP:** Proporción de Falsos Positivos.    **PFNN:** Proporción de Falsos Positivos.

TABLA A.5: Resultados por el método de muestreo hammersley con parámetros optimizados por el método de recocido simulado con 50 iris para 100% y 75% del tamaño de la base de datos

No. subimágenes	Tamaño de la subimagen	Tamaño de la base de datos Miles Research									
		100%					75%				
		113	128	178	184	46	57	73	99	5x15	5x15
Práctico	PVP:	10.98%	10.48%	17.53%	13.11%	11.31%	8.24%	6.55%	16.82%		
	PVN:	79.00%	84.28%	73.40%	74.49%	81.55%	80.40%	87.62%	75.79%		
	PFPP:	7.99%	3.81%	5.74%	9.76%	5.12%	8.50%	1.19%	4.88%		
	PFNN:	2.03%	1.43%	3.33%	2.63%	2.02%	2.86%	4.64%	2.51%		
	Efectividad:	<b>89.99%</b>	<b>94.76%</b>	<b>90.93%</b>	<b>87.61%</b>	<b>92.86%</b>	<b>88.64%</b>	<b>94.17%</b>	<b>92.61%</b>		
Teórico	Desv. Std:	<b>7.61%</b>	<b>4.80%</b>	<b>6.30%</b>	<b>16.46%</b>	<b>6.06%</b>	<b>15.48%</b>	<b>1.36%</b>	<b>2.80%</b>		
	PVP:	12.74%	11.50%	18.72%	14.21%	12.52%	11.12%	10.06%	17.66%		
	PVN:	84.55%	85.79%	76.54%	81.84%	84.09%	86.51%	85.65%	78.50%		
	PFPP:	0.31%	0.09%	0.87%	0.42%	0.43%	0.16%	0.91%	0.37%		
	PFNN:	2.40%	2.63%	3.88%	3.53%	2.96%	2.22%	3.39%	3.47%		
Práctico	Efectividad:	<b>97.29%</b>	<b>97.29%</b>	<b>95.26%</b>	<b>96.05%</b>	<b>96.61%</b>	<b>97.63%</b>	<b>95.71%</b>	<b>96.16%</b>		
	Desv. Std:	<b>0.93%</b>	<b>2.31%</b>	<b>1.72%</b>	<b>3.07%</b>	<b>0.80%</b>	<b>1.29%</b>	<b>0.51%</b>	<b>1.89%</b>		
	Tiempo (s)	<b>19.833</b>	<b>20.59</b>	<b>30.33</b>	<b>32.435</b>	<b>10.123</b>	<b>9.2112</b>	<b>11.693</b>	<b>15.1</b>		
	PVP:	9.08%	5.13%	6.16%	3.94%	7.56%	7.36%	7.22%	4.10%		
	PVN:	90.92%	94.87%	93.85%	96.06%	92.27%	91.96%	92.43%	95.73%		
Teórico	PFPP:	0.00%	0.00%	0.00%	0.00%	0.17%	0.68%	0.34%	0.00%		
	PFNN:	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.17%		
	Efectividad:	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>99.83%</b>	<b>99.32%</b>	<b>99.66%</b>	<b>99.83%</b>		
	Desv. Std:	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.38%</b>	<b>0.94%</b>	<b>0.77%</b>	<b>0.38%</b>		
	PVP:	11.77%	8.06%	9.03%	6.94%	10.32%	10.13%	10.00%	7.26%		
Práctico	PVN:	88.23%	91.94%	90.97%	93.07%	89.68%	89.71%	90.00%	92.74%		
	PFPP:	0.00%	0.00%	0.00%	0.00%	0.00%	0.13%	0.00%	0.00%		
	PFNN:	0.00%	0.00%	0.00%	0.00%	0.00%	0.04%	0.00%	0.00%		
	Efectividad:	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>99.84%</b>	<b>100.00%</b>	<b>100.00%</b>		
	Desv. Std:	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.36%</b>	<b>0.00%</b>	<b>0.00%</b>		
Teórico	Tiempo (s)	<b>13.832</b>	<b>14.549</b>	<b>21.319</b>	<b>21.826</b>	<b>6.8334</b>	<b>6.1668</b>	<b>7.9494</b>	<b>10.481</b>		



## A.5. Método ARX

Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos.    **PVN:** Proporción de Verdaderos Negativos.  
**PFP:** Proporción de Falsos Positivos.    **PFN:** Proporción de Falsos Positivos.

TABLA A.7: Resultados por el método ARX para 100% y 75% del tamaño de la base de datos con  $na = 5$ ,  $nb = 5$  y  $grid = 4$ .

grid	4x4		5x5		8x8		10x10		
	100% Miles	75% Miles	100% Miles	75% Miles	100% Miles	75% Miles	100% Miles	75% Miles	
na=5, nb=5	PVP:	12.58%	0.86%	2.40%	1.72%	14.50%	1.72%	2.40%	1.72%
	PVN:	76.05%	97.41%	84.43%	93.10%	74.77%	92.24%	87.43%	91.38%
	PFP:	0.60%	0.00%	0.00%	0.00%	4.33%	0.00%	0.00%	0.00%
	PFN:	10.78%	1.72%	13.17%	5.17%	6.40%	6.03%	10.18%	6.90%
Práctico	Efectividad:	<b>88.62%</b>	<b>98.28%</b>	<b>86.83%</b>	<b>94.83%</b>	<b>89.27%</b>	<b>93.97%</b>	<b>89.82%</b>	<b>93.10%</b>
	PVP:	19.21%	5.65%	14.12%	9.68%	9.58%	10.48%	8.76%	5.65%
	PVN:	72.32%	94.36%	80.23%	90.32%	83.83%	88.71%	85.03%	93.55%
	PFP:	3.39%	0.00%	2.26%	0.00%	1.20%	0.81%	0.28%	0.81%
Teórico	PFN:	5.08%	0.00%	3.39%	0.00%	5.39%	0.00%	5.93%	0.00%
	Efectividad:	<b>91.53%</b>	<b>100.00%</b>	<b>94.35%</b>	<b>100.00%</b>	<b>93.41%</b>	<b>99.19%</b>	<b>93.79%</b>	<b>99.19%</b>
	Tiempo (s)	<b>397.64</b>	<b>247.8</b>	<b>235.87</b>	<b>177.18</b>	<b>108.97</b>	<b>72.635</b>	<b>65.234</b>	<b>47.368</b>



---

## Apéndice B

# Resultados de la base de datos UBIRIS (Identificación)

---

Se probaron las mismas condiciones que en el apéndice A, pero con esta base de datos fue probado el algoritmo de optimización por recocido simulado y el muestreo Hammersley factorial.

### Condiciones experimentales para la optimización del método de muestreo para recocido simulado

Las combinaciones que se usaron fueron obtenidas en la sección 5.4.3 para recocido simulado fueron:

TABLA B.1: Combinaciones de parámetros de muestreo Hammersley obtenidos por recocido simulado

Número de subimágenes	113	128	178	184	46	57	73	99
Altura de las subimágenes	15	29	11	14	21	31	21	5
Largo de las subimágenes	32	5	5	5	20	5	5	15
Número de subimágenes	169	150	141	172	127	156	149	115
Altura de las subimágenes	29	16	35	42	38	33	44	48
Largo de las subimágenes	98	5	13	5	5	5	16	13

### Condiciones experimentales para el método de muestreo Hammersley factorial

Las combinaciones para el primer muestreo son:

TABLA B.2: Combinaciones de parámetros de muestreo Hammersley obtenidos por recocido con los menores tiempos de identificación, los cuales serán empleados en el primer muestreo Hammersley factorial.

Primer muestreo	1	2	3	4
Número de subimágenes	46	57	73	99
Altura de las subimágenes	21	31	21	5
Largo de las subimágenes	20	5	5	15

Las combinaciones para el segundo muestreo son:



TABLA B.3: Combinaciones de parámetros de muestreo Hammersley obtenidos por recocido con los menores tiempos de identificación, los cuales serán empleados en el segundo muestreo Hammersley factorial.

Segundo muestreo	1	2	3	4	5	6	7	8
Número de subimágenes	169	150	141	172	127	156	149	115
Altura de las subimágenes	29	16	35	42	38	33	44	48
Largo de las subimágenes	98	5	13	5	5	5	16	13

### Descripción de la base de datos

Como se observa en la tabla 5.2 esta base de datos tiene 5 clasificaciones de acuerdo las muestras que fueron eliminadas por el ruido, es decir la base de datos al 100% contiene todas las imágenes a las que les fueron localizados correctamente el límite límbico y pupilar y la base de datos del 50% les fueron sustraídas las imágenes que presentaban mayor ruido y alteraban el funcionamiento de los métodos. El número de iris que se emplearon en la base ficticia de iris, para crear los umbrales y las pruebas se muestran en la tabla B.4.

TABLA B.4: Clasificación del número de iris ocupados en cada sección del experimento de la base de datos UBIRIS.

Porcentaje	Base	Umbral	Pruebas
100%	173	51	789
90%	173	46	693
80%	173	41	597
70%	173	36	501
60%	173	31	404
50%	173	26	308

## B.1. Método de muestreo aleatorio con reposición

Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos. **PVN:** Proporción de Verdaderos Negativos.

**PFP:** Proporción de Falsos Positivos. **PFN:** Proporción de Falsos Positivos.

TABLA B.5: Resultados por el método de muestreo aleatorio con reposición para 100% y 90% del tamaño de la base de datos

Tamaño de la subimagen	Tamaño de la base de datos UBIRIS											
	100 subimágenes				150 subimágenes				200 subimágenes			
	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)
Práctico	PVP:	54.04%	53.73%	54.54%	55.43%	54.62%	53.78%	56.14%	56.09%	55.43%	56.09%	55.43%
	PVN:	34.54%	36.27%	36.83%	34.77%	36.42%	36.60%	34.06%	35.31%	36.42%	35.31%	36.42%
	PFP:	2.31%	1.14%	0.94%	1.40%	0.79%	0.61%	1.42%	0.94%	1.14%	0.94%	1.14%
	PFN:	9.11%	8.86%	7.69%	8.40%	8.17%	9.01%	8.38%	7.67%	7.01%	8.38%	7.01%
Teórico	Efectividad:	<b>88.58%</b>	<b>90.00%</b>	<b>91.37%</b>	<b>90.20%</b>	<b>91.04%</b>	<b>90.38%</b>	<b>90.20%</b>	<b>91.40%</b>	<b>91.85%</b>	<b>91.40%</b>	<b>91.85%</b>
	Desv. Std:	<b>1.14%</b>	<b>1.36%</b>	<b>0.77%</b>	<b>1.13%</b>	<b>1.76%</b>	<b>1.38%</b>	<b>3.43%</b>	<b>1.13%</b>	<b>1.45%</b>	<b>1.13%</b>	<b>1.45%</b>
	PVP:	54.66%	54.75%	55.73%	54.75%	56.06%	55.66%	56.99%	55.71%	55.25%	55.71%	55.25%
	PVN:	35.32%	36.39%	36.98%	36.06%	36.73%	36.73%	34.84%	36.37%	37.73%	36.37%	37.73%
Teórico	PFP:	2.70%	2.49%	2.12%	1.51%	2.01%	2.01%	1.87%	1.40%	1.30%	1.40%	1.30%
	PFN:	7.32%	6.37%	5.17%	7.68%	5.20%	5.61%	6.29%	6.53%	5.73%	6.53%	5.73%
	Efectividad:	<b>89.98%</b>	<b>91.14%</b>	<b>92.72%</b>	<b>90.81%</b>	<b>92.79%</b>	<b>92.38%</b>	<b>91.83%</b>	<b>92.07%</b>	<b>92.98%</b>	<b>92.07%</b>	<b>92.98%</b>
	Desv. Std:	<b>0.47%</b>	<b>1.06%</b>	<b>0.31%</b>	<b>0.59%</b>	<b>0.70%</b>	<b>0.84%</b>	<b>1.63%</b>	<b>0.73%</b>	<b>0.99%</b>	<b>0.73%</b>	<b>0.99%</b>
Práctico	Tiempo (s)	<b>350.45</b>	<b>556.08</b>	<b>572.02</b>	<b>888.8</b>	<b>894.27</b>	<b>919.26</b>	<b>1290.6</b>	<b>1273.8</b>	<b>1315.7</b>	<b>1273.8</b>	<b>1315.7</b>
	PVP:	61.13%	60.49%	61.01%	63.00%	63.61%	63.12%	63.75%	63.03%	63.64%	63.03%	63.64%
	PVN:	29.24%	31.40%	30.77%	29.47%	30.16%	29.93%	28.72%	29.90%	29.55%	29.90%	29.55%
	PFP:	2.11%	0.52%	1.24%	1.85%	0.92%	1.33%	1.36%	0.63%	1.39%	0.63%	1.39%
Teórico	PFN:	7.53%	7.59%	6.98%	5.69%	5.31%	5.63%	6.18%	6.44%	5.43%	6.44%	5.43%
	Efectividad:	<b>90.36%</b>	<b>91.89%</b>	<b>91.78%</b>	<b>92.47%</b>	<b>93.77%</b>	<b>93.05%</b>	<b>92.47%</b>	<b>92.93%</b>	<b>93.19%</b>	<b>92.93%</b>	<b>93.19%</b>
	Desv. Std:	<b>4.22%</b>	<b>1.82%</b>	<b>1.83%</b>	<b>2.81%</b>	<b>1.58%</b>	<b>0.56%</b>	<b>1.60%</b>	<b>2.37%</b>	<b>1.51%</b>	<b>2.37%</b>	<b>1.51%</b>
	PVP:	62.30%	62.70%	61.80%	62.67%	63.34%	62.92%	65.96%	64.80%	63.82%	64.80%	63.82%
Teórico	PVN:	30.18%	31.27%	31.97%	30.65%	30.89%	31.12%	28.39%	30.49%	30.18%	30.49%	30.18%
	PFP:	2.46%	1.91%	1.29%	1.91%	1.48%	1.52%	3.04%	1.31%	2.08%	1.31%	2.08%
	PFN:	5.06%	4.12%	4.93%	4.78%	4.29%	4.44%	2.62%	3.40%	3.92%	3.40%	3.92%
	Efectividad:	<b>92.48%</b>	<b>93.96%</b>	<b>93.78%</b>	<b>93.32%</b>	<b>94.23%</b>	<b>94.05%</b>	<b>94.34%</b>	<b>95.29%</b>	<b>93.99%</b>	<b>95.29%</b>	<b>93.99%</b>
Teórico	Desv. Std:	<b>1.41%</b>	<b>1.66%</b>	<b>1.83%</b>	<b>2.55%</b>	<b>1.64%</b>	<b>0.97%</b>	<b>1.07%</b>	<b>1.06%</b>	<b>1.53%</b>	<b>1.06%</b>	<b>1.53%</b>
	Tiempo (s)	<b>348.05</b>	<b>345.66</b>	<b>360.16</b>	<b>586.15</b>	<b>593.64</b>	<b>619.49</b>	<b>863.27</b>	<b>873.98</b>	<b>913.34</b>	<b>873.98</b>	<b>913.34</b>

Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos. **PVN:** Proporción de Verdaderos Negativos.  
**PPF:** Proporción de Falsos Positivos. **PFN:** Proporción de Falsos Positivos.

TABLA B.6: Resultados por el método de muestreo aleatorio con reposición para 80% y 70% del tamaño de la base de datos

Tamaño de la subimagen	Tamaño de la base de datos UBIRIS														
	100 subimágenes			150 subimágenes			200 subimágenes			80%			70%		
	5x5 (pixeles)	15x15 (pixeles)	25x25 (pixeles)	5x5 (pixeles)	15x15 (pixeles)	25x25 (pixeles)	5x5 (pixeles)	15x15 (pixeles)	25x25 (pixeles)	5x5 (pixeles)	15x15 (pixeles)	25x25 (pixeles)	5x5 (pixeles)	15x15 (pixeles)	25x25 (pixeles)
Práctico	PVP:	67.75%	66.17%	67.22%	67.99%	67.82%	67.75%	67.75%	67.75%	66.98%	67.15%	69.63%	66.98%	67.15%	69.63%
	PVN:	28.29%	30.40%	28.32%	30.24%	30.20%	29.33%	29.33%	29.33%	29.60%	30.30%	28.29%	29.60%	30.30%	28.29%
	PPF:	0.91%	0.70%	0.97%	0.30%	0.30%	0.70%	0.70%	0.70%	0.50%	0.40%	0.67%	0.50%	0.40%	0.67%
	PFN:	3.05%	2.72%	3.49%	1.48%	1.68%	2.21%	2.21%	2.21%	2.92%	2.15%	1.41%	2.92%	2.15%	1.41%
Teórico	Efectividad:	<b>96.04%</b>	<b>96.58%</b>	<b>95.54%</b>	<b>98.22%</b>	<b>98.02%</b>	<b>97.08%</b>	<b>97.08%</b>	<b>97.08%</b>	<b>96.58%</b>	<b>97.45%</b>	<b>97.92%</b>	<b>96.58%</b>	<b>97.45%</b>	<b>97.92%</b>
	Desv. Std:	<b>1.58%</b>	<b>0.77%</b>	<b>2.55%</b>	<b>1.10%</b>	<b>1.32%</b>	<b>1.12%</b>	<b>1.12%</b>	<b>1.12%</b>	<b>1.36%</b>	<b>1.32%</b>	<b>1.12%</b>	<b>1.36%</b>	<b>1.32%</b>	<b>1.12%</b>
	PVP:	67.10%	65.61%	67.21%	67.09%	67.20%	67.21%	67.21%	67.21%	67.63%	66.99%	68.50%	67.63%	66.99%	68.50%
	PVN:	29.49%	31.79%	29.62%	31.50%	31.39%	30.38%	30.38%	30.38%	30.55%	31.42%	29.91%	30.55%	31.42%	29.91%
Práctico	PVP:	1.11%	0.56%	1.07%	0.32%	0.40%	0.97%	0.97%	0.97%	0.86%	0.56%	0.44%	0.86%	0.56%	0.44%
	PVN:	2.31%	2.04%	2.10%	1.09%	1.01%	1.44%	1.44%	1.44%	0.96%	1.04%	1.16%	0.96%	1.04%	1.16%
	Efectividad:	<b>96.58%</b>	<b>97.40%</b>	<b>96.83%</b>	<b>98.59%</b>	<b>98.59%</b>	<b>97.59%</b>	<b>97.59%</b>	<b>97.59%</b>	<b>98.18%</b>	<b>98.40%</b>	<b>98.40%</b>	<b>98.18%</b>	<b>98.40%</b>	<b>98.40%</b>
	Desv. Std:	<b>1.14%</b>	<b>0.66%</b>	<b>1.52%</b>	<b>0.91%</b>	<b>0.62%</b>	<b>0.80%</b>	<b>0.80%</b>	<b>0.80%</b>	<b>1.32%</b>	<b>0.81%</b>	<b>0.83%</b>	<b>1.32%</b>	<b>0.81%</b>	<b>0.83%</b>
Teórico	Tiempo (s)	<b>287.92</b>	<b>301.12</b>	<b>309.59</b>	<b>499.68</b>	<b>511.11</b>	<b>532</b>	<b>532</b>	<b>532</b>	<b>739.12</b>	<b>751.67</b>	<b>778.82</b>	<b>739.12</b>	<b>751.67</b>	<b>778.82</b>
	PVP:	64.75%	66.95%	67.23%	69.22%	67.67%	67.03%	67.03%	67.03%	68.30%	68.86%	67.51%	68.30%	68.86%	67.51%
	PVN:	31.78%	31.58%	31.26%	30.06%	31.58%	31.74%	31.74%	31.74%	31.26%	30.10%	32.22%	31.26%	30.10%	32.22%
	PPF:	0.24%	0.40%	0.44%	0.04%	0.36%	0.12%	0.12%	0.12%	0.28%	0.04%	0.00%	0.28%	0.04%	0.00%
Teórico	PPF:	3.23%	1.08%	1.08%	0.68%	0.40%	1.12%	1.12%	1.12%	0.16%	1.00%	0.28%	0.16%	1.00%	0.28%
	Efectividad:	<b>96.53%</b>	<b>98.52%</b>	<b>98.48%</b>	<b>99.28%</b>	<b>99.24%</b>	<b>98.76%</b>	<b>98.76%</b>	<b>98.76%</b>	<b>99.56%</b>	<b>98.96%</b>	<b>99.72%</b>	<b>99.56%</b>	<b>98.96%</b>	<b>99.72%</b>
	Desv. Std:	<b>2.75%</b>	<b>1.39%</b>	<b>1.44%</b>	<b>0.64%</b>	<b>0.43%</b>	<b>0.71%</b>	<b>0.71%</b>	<b>0.71%</b>	<b>0.48%</b>	<b>1.45%</b>	<b>0.30%</b>	<b>0.48%</b>	<b>1.45%</b>	<b>0.30%</b>
	PVP:	65.82%	66.50%	66.57%	68.39%	66.67%	66.71%	66.71%	66.71%	67.19%	68.28%	66.47%	67.19%	68.28%	66.47%
Teórico	PVN:	32.43%	32.98%	32.87%	31.24%	32.92%	32.81%	32.81%	32.81%	32.70%	31.42%	33.38%	32.70%	31.42%	33.38%
	PPF:	0.79%	0.20%	0.06%	0.19%	0.22%	0.26%	0.26%	0.26%	0.07%	0.05%	0.03%	0.07%	0.05%	0.03%
	PFN:	0.96%	0.32%	0.50%	0.18%	0.19%	0.22%	0.22%	0.22%	0.04%	0.25%	0.12%	0.04%	0.25%	0.12%
	Efectividad:	<b>98.25%</b>	<b>99.48%</b>	<b>99.44%</b>	<b>99.63%</b>	<b>99.59%</b>	<b>99.52%</b>	<b>99.52%</b>	<b>99.52%</b>	<b>99.89%</b>	<b>99.70%</b>	<b>99.85%</b>	<b>99.89%</b>	<b>99.70%</b>	<b>99.85%</b>
Teórico	Desv. Std:	<b>1.46%</b>	<b>0.24%</b>	<b>0.43%</b>	<b>0.48%</b>	<b>0.28%</b>	<b>0.45%</b>	<b>0.45%</b>	<b>0.45%</b>	<b>0.17%</b>	<b>0.28%</b>	<b>0.16%</b>	<b>0.17%</b>	<b>0.28%</b>	<b>0.16%</b>
	Tiempo (s)	<b>361.15</b>	<b>362.7</b>	<b>373.69</b>	<b>582.1</b>	<b>585.32</b>	<b>610.67</b>	<b>610.67</b>	<b>610.67</b>	<b>834.06</b>	<b>862.55</b>	<b>871.01</b>	<b>834.06</b>	<b>862.55</b>	<b>871.01</b>

Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos. **PVN:** Proporción de Verdaderos Negativos.  
**PFP:** Proporción de Falsos Positivos. **PFN:** Proporción de Falsos Positivos.

TABLA B.7: Resultados por el método de muestreo aleatorio con reposición para 50% del tamaño de la base de datos

Tamaño de la subimagen	100 subimágenes			150 subimágenes			200 subimágenes			UBIRIS		
	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	50%		
										5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)
<b>PVP:</b>	12.88%	11.31%	10.40%	8.87%	12.14%	12.98%	12.42%	11.68%	16.43%			
<b>PVN:</b>	78.16%	80.12%	71.19%	79.78%	80.60%	82.37%	73.03%	83.31%	76.79%			
<b>PFP:</b>	5.26%	7.02%	16.15%	8.59%	5.36%	2.14%	12.17%	1.91%	5.36%			
<b>PFN:</b>	3.70%	1.55%	2.27%	2.76%	1.90%	2.50%	2.39%	3.10%	1.43%			
<b>Efectividad:</b>	<b>91.05%</b>	<b>91.43%</b>	<b>81.59%</b>	<b>88.64%</b>	<b>92.74%</b>	<b>95.36%</b>	<b>85.45%</b>	<b>95.00%</b>	<b>93.21%</b>			
<b>Desv. Std:</b>	<b>5.03%</b>	<b>2.37%</b>	<b>12.33%</b>	<b>7.23%</b>	<b>6.63%</b>	<b>2.36%</b>	<b>9.65%</b>	<b>2.68%</b>	<b>9.06%</b>			
<b>PVP:</b>	15.15%	12.32%	12.35%	10.79%	13.82%	14.92%	12.99%	13.82%	17.27%			
<b>PVN:</b>	79.43%	83.84%	84.15%	85.71%	83.47%	82.71%	82.72%	82.80%	80.13%			
<b>PFP:</b>	2.04%	1.13%	1.05%	0.51%	0.38%	0.00%	0.44%	0.37%	0.10%			
<b>PFN:</b>	3.38%	2.71%	2.46%	2.99%	2.34%	2.37%	3.85%	3.02%	2.50%			
<b>Efectividad:</b>	<b>94.58%</b>	<b>96.16%</b>	<b>96.50%</b>	<b>96.50%</b>	<b>97.29%</b>	<b>97.63%</b>	<b>95.71%</b>	<b>96.61%</b>	<b>97.40%</b>			
<b>Desv. Std:</b>	<b>1.86%</b>	<b>1.76%</b>	<b>2.31%</b>	<b>2.86%</b>	<b>1.22%</b>	<b>1.35%</b>	<b>1.94%</b>	<b>2.15%</b>	<b>1.53%</b>			
<b>Tiempo (s)</b>	<b>13.315</b>	<b>14.729</b>	<b>16.432</b>	<b>28.954</b>	<b>24.455</b>	<b>27.349</b>	<b>35.559</b>	<b>36.725</b>	<b>38.918</b>			

## B.2. Método de muestreo aleatorio sin reposición

Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos.    **PVN:** Proporción de Verdaderos Negativos.  
**PFP:** Proporción de Falsos Positivos.      **PFN:** Proporción de Falsos Positivos.

TABLA B.8: Resultados por el método de muestreo aleatorio sin reposición para 100% y 90% del tamaño de la base de datos

Tamaño de la subimagen	Tamaño de la base de datos UBIRIS											
	100 subimágenes				150 subimágenes				200 subimágenes			
	5x5 (pixeles)	15x15 (pixeles)	25x25 (pixeles)		5x5 (pixeles)	15x15 (pixeles)	25x25 (pixeles)		5x5 (pixeles)	15x15 (pixeles)	25x25 (pixeles)	
Práctico	PVP:	51.95%	55.86%	55.64%	53.71%	57.59%	55.71%		54.65%	56.47%	55.56%	
	PVN:	35.64%	36.19%	36.22%	35.56%	34.52%	36.83%		34.95%	34.01%	36.57%	
	PFP:	1.93%	1.12%	0.66%	1.85%	1.04%	0.99%		1.65%	2.11%	0.56%	
	PFN:	10.48%	6.83%	7.49%	8.88%	6.85%	6.47%		8.76%	7.41%	7.31%	
	Efectividad:	<b>87.59%</b>	<b>92.06%</b>	<b>91.85%</b>	<b>89.26%</b>	<b>92.11%</b>	<b>92.54%</b>		<b>89.59%</b>	<b>90.48%</b>	<b>92.13%</b>	
Teórico	Desv. Std:	<b>2.67%</b>	<b>1.38%</b>	<b>0.74%</b>	<b>0.67%</b>	<b>1.36%</b>	<b>0.79%</b>		<b>1.93%</b>	<b>1.39%</b>	<b>0.99%</b>	
	PVP:	53.22%	56.02%	56.10%	54.04%	57.17%	55.31%		55.12%	56.18%	55.79%	
	PVN:	36.14%	37.34%	36.81%	36.53%	35.76%	37.86%		35.98%	36.20%	37.54%	
	PFP:	2.91%	1.30%	1.52%	2.35%	1.32%	1.36%		2.12%	1.18%	0.79%	
	PFN:	7.73%	5.34%	5.57%	7.08%	5.76%	5.48%		6.79%	6.44%	5.88%	
Práctico	Efectividad:	<b>89.35%</b>	<b>93.36%</b>	<b>92.90%</b>	<b>90.57%</b>	<b>92.93%</b>	<b>93.17%</b>		<b>91.09%</b>	<b>92.38%</b>	<b>93.33%</b>	
	Desv. Std:	<b>0.99%</b>	<b>1.35%</b>	<b>0.46%</b>	<b>0.94%</b>	<b>1.05%</b>	<b>0.73%</b>		<b>1.84%</b>	<b>0.12%</b>	<b>0.66%</b>	
	Tiempo (s)	<b>388.91</b>	<b>396.65</b>	<b>413.71</b>	<b>662.08</b>	<b>671.59</b>	<b>704.2</b>		<b>970.05</b>	<b>990.64</b>	<b>1029.5</b>	
	PVP:	59.71%	63.95%	61.62%	57.29%	64.07%	62.66%		64.24%	64.94%	62.86%	
	PVN:	29.93%	29.00%	30.25%	30.45%	28.80%	29.44%		28.63%	28.89%	30.27%	
Teórico	PFP:	1.33%	1.44%	0.98%	0.95%	0.72%	0.89%		2.08%	1.04%	0.69%	
	PFN:	9.03%	5.60%	7.16%	11.31%	6.41%	7.01%		5.05%	5.14%	6.18%	
	Efectividad:	<b>89.64%</b>	<b>92.96%</b>	<b>91.86%</b>	<b>87.73%</b>	<b>92.87%</b>	<b>92.09%</b>		<b>92.87%</b>	<b>93.82%</b>	<b>93.13%</b>	
	Desv. Std:	<b>2.49%</b>	<b>1.66%</b>	<b>1.80%</b>	<b>3.16%</b>	<b>3.52%</b>	<b>4.07%</b>		<b>2.28%</b>	<b>1.56%</b>	<b>1.84%</b>	
	PVP:	61.71%	64.80%	63.89%	62.54%	65.93%	64.59%		65.02%	65.07%	63.99%	
Práctico	PVN:	30.10%	29.32%	29.94%	29.20%	29.23%	30.16%		28.86%	30.08%	30.79%	
	PFP:	2.57%	2.40%	2.54%	3.41%	1.57%	1.51%		3.13%	1.15%	1.50%	
	PFN:	5.63%	3.48%	3.63%	4.84%	3.27%	3.74%		2.99%	3.69%	3.73%	
	Efectividad:	<b>91.80%</b>	<b>94.13%</b>	<b>93.83%</b>	<b>91.74%</b>	<b>95.15%</b>	<b>94.75%</b>		<b>93.88%</b>	<b>95.15%</b>	<b>94.78%</b>	
	Desv. Std:	<b>1.87%</b>	<b>1.27%</b>	<b>0.90%</b>	<b>1.28%</b>	<b>1.26%</b>	<b>1.25%</b>		<b>1.93%</b>	<b>0.86%</b>	<b>0.61%</b>	
Tiempo (s)	<b>355.67</b>	<b>352.84</b>	<b>363.6</b>	<b>606.75</b>	<b>621.24</b>	<b>625.71</b>		<b>863.82</b>	<b>865.67</b>	<b>895.47</b>		

Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos.      **PVN:** Proporción de Verdaderos Negativos.  
**PPF:** Proporción de Falsos Positivos.      **PFN:** Proporción de Falsos Positivos.

TABLA B.9: Resultados por el método de muestreo aleatorio sin reposición para 80% y 70% del tamaño de la base de datos

Tamaño de la subimagen	Tamaño de la base de datos UBIRIS											
	100 subimágenes				150 subimágenes				200 subimágenes			
	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)
<b>Práctico</b>	PVP:	68.02%	67.55%	67.85%	68.32%	68.66%	67.62%	67.69%	67.95%	69.40%		
	PVN:	28.29%	29.60%	29.77%	28.86%	29.66%	29.20%	30.27%	30.54%	28.29%		
	PPF:	1.17%	0.54%	0.81%	0.70%	0.70%	0.84%	0.37%	0.27%	1.31%		
	PFN:	2.52%	2.32%	1.58%	2.11%	0.97%	2.35%	1.68%	1.24%	1.01%		
	<b>Efectividad:</b>	<b>96.31%</b>	<b>97.15%</b>	<b>97.62%</b>	<b>97.18%</b>	<b>98.32%</b>	<b>96.81%</b>	<b>97.95%</b>	<b>98.49%</b>	<b>97.69%</b>		
<b>Teórico</b>	Desv. Std:	1.00%	1.63%	0.51%	1.65%	0.34%	2.17%	1.68%	0.86%	0.69%		
	PVP:	67.18%	67.15%	67.38%	68.23%	67.56%	67.35%	67.23%	67.34%	68.12%		
	PVN:	29.75%	30.84%	31.11%	29.86%	31.38%	30.71%	31.43%	31.72%	30.03%		
	PPF:	1.09%	0.60%	0.74%	1.05%	0.28%	0.67%	0.52%	0.34%	0.91%		
	PFN:	1.98%	1.41%	0.77%	0.86%	0.78%	1.27%	0.83%	0.60%	0.94%		
<b>Efectividad:</b>	<b>96.93%</b>	<b>97.99%</b>	<b>98.49%</b>	<b>98.09%</b>	<b>98.93%</b>	<b>98.06%</b>	<b>98.65%</b>	<b>99.06%</b>	<b>98.15%</b>			
Desv. Std:	0.88%	0.96%	0.48%	0.65%	0.45%	0.97%	0.42%	0.33%	0.55%			
<b>Tiempo (s)</b>	<b>293.83</b>	<b>306.46</b>	<b>308.43</b>	<b>511.68</b>	<b>526.81</b>	<b>536.04</b>	<b>736.44</b>	<b>741.44</b>	<b>767.85</b>			
<b>Práctico</b>	PVP:	66.95%	67.07%	68.82%	66.11%	67.51%	66.83%	67.27%	67.39%	67.07%		
	PVN:	31.98%	31.74%	30.38%	32.46%	32.02%	32.26%	32.30%	31.82%	31.70%		
	PPF:	0.16%	0.00%	0.36%	0.08%	0.16%	0.32%	0.12%	0.12%	0.40%		
	PFN:	0.92%	1.20%	0.44%	1.36%	0.32%	0.60%	0.32%	0.68%	0.84%		
	<b>Efectividad:</b>	<b>98.92%</b>	<b>98.80%</b>	<b>99.20%</b>	<b>98.56%</b>	<b>99.52%</b>	<b>99.08%</b>	<b>99.56%</b>	<b>99.20%</b>	<b>98.76%</b>		
<b>Teórico</b>	Desv. Std:	1.41%	1.62%	0.37%	1.68%	0.27%	0.39%	0.26%	0.80%	0.55%		
	PVP:	66.43%	66.66%	67.78%	66.05%	66.48%	65.94%	66.25%	66.74%	66.35%		
	PVN:	33.12%	32.86%	31.81%	33.51%	33.30%	33.69%	33.57%	33.15%	33.17%		
	PPF:	0.21%	0.10%	0.22%	0.20%	0.07%	0.06%	0.03%	0.00%	0.13%		
	PFN:	0.24%	0.38%	0.19%	0.25%	0.15%	0.32%	0.16%	0.11%	0.36%		
<b>Efectividad:</b>	<b>99.56%</b>	<b>99.51%</b>	<b>99.59%</b>	<b>99.55%</b>	<b>99.78%</b>	<b>99.63%</b>	<b>99.81%</b>	<b>99.89%</b>	<b>99.52%</b>			
Desv. Std:	0.17%	0.68%	0.42%	0.17%	0.20%	0.26%	0.13%	0.17%	0.31%			
<b>Tiempo (s)</b>	<b>247.46</b>	<b>255.24</b>	<b>262.75</b>	<b>422.36</b>	<b>429.67</b>	<b>446.5</b>	<b>612.86</b>	<b>629.31</b>	<b>652.2</b>			

Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos.    **PVN:** Proporción de Verdaderos Negativos.  
**PPF:** Proporción de Falsos Positivos.      **PFN:** Proporción de Falsos Positivos.

TABLA B.10: Resultados por el método de muestreo aleatorio sin reposición para 50% del tamaño de la base de datos

Tamaño de la subimagen	UBIRIS								
	100 subimágenes			150 subimágenes			200 subimágenes		
	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)
<b>PVP:</b>	63.25%	63.25%	63.70%	63.70%	64.81%	63.83%	62.79%	64.94%	61.43%
<b>PVN:</b>	36.69%	36.62%	36.17%	36.30%	35.20%	36.04%	37.21%	35.07%	38.57%
<b>PPF:</b>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<b>PFN:</b>	0.06%	0.13%	0.13%	0.00%	0.00%	0.13%	0.00%	0.00%	0.00%
<b>Efectividad:</b>	<b>99.94%</b>	<b>99.87%</b>	<b>99.87%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>99.87%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>
<b>Desv. Std:</b>	<b>0.15%</b>	<b>0.18%</b>	<b>0.18%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.18%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>
<b>PVP:</b>	62.28%	62.34%	62.75%	62.64%	63.65%	62.87%	61.80%	63.77%	60.54%
<b>PVN:</b>	37.73%	37.67%	37.25%	37.37%	36.35%	37.13%	38.20%	36.23%	39.46%
<b>PPF:</b>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<b>PFN:</b>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
<b>Efectividad:</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>
<b>Desv. Std:</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>
<b>Tiempo (s)</b>	149.72	154.9	163.17	258.58	263.6	273	379.61	384.76	398.98

### B.3. Método de muestreo Hammerlsey

Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos. **PVN:** Proporción de Verdaderos Negativos.

**PFP:** Proporción de Falsos Positivos. **PFN:** Proporción de Falsos Positivos.

TABLA B.11: Resultados por el método de muestreo Hammersley para 100% y 90% del tamaño de la base de datos

Tamaño de la subimagen	Tamaño de la base de datos UBIRIS											
	100 subimágenes				150 subimágenes				200 subimágenes			
	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)		5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)		5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	
Práctico	PVP:	53.48%	53.60%	57.01%	54.95%	57.06%	57.13%		57.46%	58.15%	55.99%	
	PVN:	35.05%	35.99%	34.87%	33.96%	34.80%	35.69%		33.15%	33.43%	36.09%	
	PFP:	1.17%	0.36%	1.78%	1.42%	0.79%	0.79%		1.29%	2.51%	0.53%	
	PFN:	10.31%	10.05%	6.35%	9.67%	7.36%	6.40%		8.10%	5.91%	7.39%	
Teórico	Efectividad:	<b>88.53%</b>	<b>89.59%</b>	<b>91.88%</b>	<b>88.91%</b>	<b>91.85%</b>	<b>92.82%</b>		<b>90.61%</b>	<b>91.57%</b>	<b>92.08%</b>	
	Desv. Std:	<b>2.46%</b>	<b>1.71%</b>	<b>1.42%</b>	<b>0.91%</b>	<b>1.04%</b>	<b>1.70%</b>		<b>1.34%</b>	<b>0.94%</b>	<b>2.11%</b>	
	PVP:	55.19%	55.58%	56.68%	55.18%	57.80%	56.57%		58.27%	56.95%	56.52%	
	PVN:	35.29%	36.54%	36.46%	35.30%	35.20%	37.07%		33.85%	35.81%	37.12%	
Teórico	PFP:	2.38%	1.23%	1.61%	1.39%	1.71%	0.64%		2.01%	1.35%	1.02%	
	PFN:	7.14%	6.65%	5.25%	8.13%	5.29%	5.71%		5.87%	5.88%	5.33%	
	Efectividad:	<b>90.47%</b>	<b>92.12%</b>	<b>93.14%</b>	<b>90.48%</b>	<b>93.00%</b>	<b>93.64%</b>		<b>92.12%</b>	<b>92.76%</b>	<b>93.64%</b>	
	Desv. Std:	<b>1.03%</b>	<b>0.49%</b>	<b>1.11%</b>	<b>0.71%</b>	<b>0.33%</b>	<b>1.08%</b>		<b>0.52%</b>	<b>0.91%</b>	<b>0.55%</b>	
Práctico	Tiempo (s)	<b>370</b>	<b>379.67</b>	<b>398.25</b>	<b>629.11</b>	<b>646.68</b>	<b>663.13</b>		<b>932.19</b>	<b>950.81</b>	<b>980.8</b>	
	PVP:	61.50%	63.09%	63.43%	65.05%	64.79%	63.95%		61.67%	64.82%	64.50%	
	PVN:	29.18%	29.24%	29.52%	28.11%	28.60%	29.38%		29.84%	28.77%	28.63%	
	PFP:	2.31%	1.56%	0.98%	2.48%	1.73%	1.15%		0.49%	1.79%	2.02%	
Teórico	PFN:	7.01%	6.12%	6.06%	4.36%	4.88%	5.51%		7.99%	4.62%	4.85%	
	Efectividad:	<b>90.68%</b>	<b>92.32%</b>	<b>92.96%</b>	<b>93.16%</b>	<b>93.39%</b>	<b>93.33%</b>		<b>91.52%</b>	<b>93.59%</b>	<b>93.13%</b>	
	Desv. Std:	<b>1.92%</b>	<b>1.72%</b>	<b>1.79%</b>	<b>1.70%</b>	<b>1.25%</b>	<b>1.30%</b>		<b>1.61%</b>	<b>1.66%</b>	<b>1.00%</b>	
	PVP:	63.38%	63.38%	65.00%	65.16%	64.51%	64.74%		64.56%	65.35%	64.20%	
Teórico	PVN:	29.18%	30.47%	29.43%	29.13%	30.14%	29.69%		29.65%	29.73%	30.18%	
	PFP:	3.52%	1.57%	2.40%	2.67%	1.45%	2.14%		1.93%	2.05%	1.71%	
	PFN:	3.92%	4.57%	3.18%	3.04%	3.91%	3.44%		3.86%	2.88%	3.92%	
	Efectividad:	<b>92.56%</b>	<b>93.85%</b>	<b>94.42%</b>	<b>94.29%</b>	<b>94.64%</b>	<b>94.42%</b>		<b>94.21%</b>	<b>95.07%</b>	<b>94.37%</b>	
Teórico	Desv. Std:	<b>0.60%</b>	<b>1.13%</b>	<b>0.72%</b>	<b>0.81%</b>	<b>0.50%</b>	<b>0.61%</b>		<b>0.69%</b>	<b>0.72%</b>	<b>0.98%</b>	
	Tiempo (s)	<b>332.74</b>	<b>335.19</b>	<b>342.24</b>	<b>563.98</b>	<b>571.13</b>	<b>594.92</b>		<b>825.73</b>	<b>840.11</b>	<b>865.98</b>	



Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos. **PVN:** Proporción de Verdaderos Negativos.  
**PPF:** Proporción de Falsos Positivos. **PFN:** Proporción de Falsos Positivos.

TABLA B.12: Resultados por el método de muestreo Hammersley para 80% y 70% del tamaño de la base de datos

Tamaño de la subimagen	Tamaño de la base de datos UBIRIS														
	100 subimágenes			150 subimágenes			200 subimágenes			80%			70%		
	5x5 (pixeles)	15x15 (pixeles)	25x25 (pixeles)	5x5 (pixeles)	15x15 (pixeles)	25x25 (pixeles)	5x5 (pixeles)	15x15 (pixeles)	25x25 (pixeles)	5x5 (pixeles)	15x15 (pixeles)	25x25 (pixeles)	5x5 (pixeles)	15x15 (pixeles)	25x25 (pixeles)
Práctico	PVP:	66.44%	69.30%	68.39%	68.36%	68.99%	68.76%	69.06%	68.02%	67.11%					
	PVN:	30.30%	28.52%	29.46%	29.77%	29.77%	28.69%	29.53%	30.67%	30.74%					
	PPF:	0.30%	1.07%	0.81%	0.40%	0.44%	1.14%	0.50%	0.27%	0.44%					
	PFN:	2.95%	1.11%	1.34%	1.48%	0.81%	1.41%	0.91%	1.04%	1.71%					
Teórico	Efectividad:	<b>96.75%</b>	<b>97.82%</b>	<b>97.85%</b>	<b>98.12%</b>	<b>98.76%</b>	<b>97.45%</b>	<b>98.59%</b>	<b>98.69%</b>	<b>97.85%</b>					
	Desv. Std:	<b>0.89%</b>	<b>0.97%</b>	<b>0.65%</b>	<b>0.51%</b>	<b>0.56%</b>	<b>0.69%</b>	<b>0.37%</b>	<b>0.69%</b>	<b>0.59%</b>					
	PVP:	66.68%	68.23%	67.29%	68.09%	67.87%	67.99%	68.09%	67.18%	66.54%					
	PVN:	31.19%	30.48%	31.05%	30.76%	31.12%	30.50%	31.00%	31.94%	31.76%					
Práctico	PVP:	67.63%	66.99%	67.07%	67.71%	66.55%	68.10%	69.14%	67.51%	67.82%					
	PVN:	31.38%	31.78%	32.06%	31.70%	31.90%	30.94%	30.78%	32.26%	31.22%					
	PPF:	0.72%	0.00%	0.08%	0.24%	0.08%	0.40%	0.04%	0.00%	0.00%					
	PFN:	0.28%	1.24%	0.80%	0.36%	1.48%	0.56%	0.04%	0.24%	0.96%					
Teórico	Efectividad:	<b>99.00%</b>	<b>98.76%</b>	<b>99.12%</b>	<b>99.40%</b>	<b>98.44%</b>	<b>99.04%</b>	<b>99.92%</b>	<b>99.76%</b>	<b>99.04%</b>					
	Desv. Std:	<b>0.90%</b>	<b>0.64%</b>	<b>0.18%</b>	<b>0.42%</b>	<b>1.30%</b>	<b>0.67%</b>	<b>0.18%</b>	<b>0.17%</b>	<b>0.59%</b>					
	PVP:	66.37%	66.93%	66.30%	66.82%	66.65%	67.03%	67.87%	66.56%	67.39%					
	PVN:	33.11%	32.92%	33.21%	33.04%	33.09%	32.56%	32.09%	33.45%	32.46%					
Práctico	PVP:	66.37%	66.93%	66.30%	66.82%	66.65%	67.03%	67.87%	66.56%	67.39%					
	PVN:	33.11%	32.92%	33.21%	33.04%	33.09%	32.56%	32.09%	33.45%	32.46%					
	PPF:	0.19%	0.07%	0.12%	0.11%	0.09%	0.03%	0.01%	0.00%	0.01%					
	PFN:	0.34%	0.07%	0.36%	0.04%	0.17%	0.38%	0.02%	0.00%	0.14%					
Teórico	Efectividad:	<b>99.48%</b>	<b>99.85%</b>	<b>99.52%</b>	<b>99.85%</b>	<b>99.74%</b>	<b>99.59%</b>	<b>99.96%</b>	<b>100.00%</b>	<b>99.85%</b>					
	Desv. Std:	<b>0.40%</b>	<b>0.24%</b>	<b>0.10%</b>	<b>0.24%</b>	<b>0.21%</b>	<b>0.24%</b>	<b>0.08%</b>	<b>0.00%</b>	<b>0.16%</b>					
	PVP:	66.37%	66.93%	66.30%	66.82%	66.65%	67.03%	67.87%	66.56%	67.39%					
	PVN:	33.11%	32.92%	33.21%	33.04%	33.09%	32.56%	32.09%	33.45%	32.46%					
Tiempo (s)	240	245.53	251.93	398.86	407.93	422.01	594.42	604.21	621.46						

Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos. **PVN:** Proporción de Verdaderos Negativos.  
**PFP:** Proporción de Falsos Positivos. **PFN:** Proporción de Falsos Positivos.

TABLA B.13: Resultados por el método de muestreo Hamersley para 50% del tamaño de la base de datos

Tamaño de la subimagen	100 subimágenes			150 subimágenes			200 subimágenes			UBIRIS		
	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)	50%		
										5x5 (píxeles)	15x15 (píxeles)	25x25 (píxeles)
<b>PVP:</b>	63.51%	65.20%	63.31%	62.79%	62.79%	63.64%	64.22%	65.07%	63.51%			
<b>PVN:</b>	36.49%	34.81%	36.69%	37.21%	37.21%	36.36%	35.78%	34.94%	36.49%			
<b>PFP:</b>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%			
<b>PFN:</b>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%			
<b>Efectividad:</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>			
<b>Desv. Std:</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>			
<b>PVP:</b>	62.46%	64.01%	62.28%	61.80%	61.80%	62.58%	63.11%	63.89%	62.46%			
<b>PVN:</b>	37.55%	35.99%	37.73%	38.20%	38.20%	37.43%	36.89%	36.11%	37.55%			
<b>PFP:</b>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%			
<b>PFN:</b>	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%			
<b>Efectividad:</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>			
<b>Desv. Std:</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>			
<b>Tiempo (s)</b>	<b>143.62</b>	<b>148.09</b>	<b>150.82</b>	<b>246.71</b>	<b>250.84</b>	<b>260.35</b>	<b>364.87</b>	<b>368.98</b>	<b>384.13</b>			

### B.4. Método de muestreo hammesley con parámetros optimizados por el método de recocido simulado

Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos.    **PVN:** Proporción de Verdaderos Negativos.  
**FPF:** Proporción de Falsos Positivos.      **PFN:** Proporción de Falsos Positivos.

TABLA B.14: Resultados por el método de muestreo Hammersley con parámetros optimizados por el método de recocido simulado con 50 iris para 100% y 90% del tamaño de la base de datos

No. subimágenes Tamaño de la subimagen	Tamaño de la base de datos UBIRIS															
	100%					90%										
	113 15x32 (pixeles)	128 29x5 (pixeles)	178 11x5 (pixeles)	184 14x5 (pixeles)	46 21x20 (pixeles)	57 31x5 (pixeles)	73 21x5 (pixeles)	99 5x15 (pixeles)	113 15x32 (pixeles)	128 29x5 (pixeles)	178 11x5 (pixeles)	184 14x5 (pixeles)	46 21x20 (pixeles)	57 31x5 (pixeles)	73 21x5 (pixeles)	99 5x15 (pixeles)
Práctico	PVP:	55.51%	57.39%	58.02%	59.11%	51.90%	55.53%	55.31%	53.93%							
	PVN:	35.31%	34.59%	34.19%	32.61%	36.98%	35.48%	35.61%	36.40%							
	FPF:	1.45%	1.14%	0.89%	2.46%	0.89%	1.17%	0.23%	0.79%							
	PFN:	7.74%	6.88%	6.90%	5.81%	10.23%	7.82%	8.86%	8.88%							
	Efectividad:	<b>90.81%</b>	<b>91.98%</b>	<b>92.21%</b>	<b>91.73%</b>	<b>88.88%</b>	<b>91.02%</b>	<b>90.91%</b>	<b>90.33%</b>							
Teórico	Desv. Std:	<b>1.64%</b>	<b>1.47%</b>	<b>1.69%</b>	<b>1.02%</b>	<b>2.27%</b>	<b>2.15%</b>	<b>0.52%</b>	<b>1.16%</b>							
	PVP:	56.32%	57.24%	57.64%	57.93%	53.50%	56.32%	57.33%	55.44%							
	PVN:	35.89%	36.05%	35.26%	34.69%	37.41%	35.94%	35.21%	36.56%							
	FPF:	2.30%	1.21%	1.14%	1.81%	1.76%	2.10%	1.98%	1.89%							
	PFN:	5.49%	5.50%	5.95%	5.57%	7.33%	5.63%	5.48%	6.11%							
Práctico	Efectividad:	<b>92.21%</b>	<b>93.29%</b>	<b>92.90%</b>	<b>92.62%</b>	<b>90.90%</b>	<b>92.26%</b>	<b>92.55%</b>	<b>92.00%</b>							
	Desv. Std:	<b>1.17%</b>	<b>1.37%</b>	<b>1.23%</b>	<b>0.46%</b>	<b>0.57%</b>	<b>1.11%</b>	<b>0.89%</b>	<b>0.60%</b>							
	Tiempo (s)	<b>335.2</b>	<b>475.21</b>	<b>660.55</b>	<b>625.31</b>	<b>159.06</b>	<b>178.4</b>	<b>206.54</b>	<b>309.23</b>							
	PVP:	62.37%	66.35%	66.67%	65.17%	62.54%	65.02%	63.72%	64.88%							
	PVN:	30.39%	28.80%	28.08%	28.83%	30.71%	29.35%	29.44%	27.91%							
Teórico	FPF:	0.89%	1.04%	1.44%	1.39%	1.18%	2.45%	1.44%	1.90%							
	PFN:	6.35%	3.81%	3.81%	4.62%	5.57%	3.17%	5.40%	5.31%							
	Efectividad:	<b>92.76%</b>	<b>95.15%</b>	<b>94.75%</b>	<b>94.00%</b>	<b>93.25%</b>	<b>94.37%</b>	<b>93.16%</b>	<b>92.79%</b>							
	Desv. Std:	<b>0.60%</b>	<b>0.95%</b>	<b>0.99%</b>	<b>1.19%</b>	<b>1.56%</b>	<b>0.75%</b>	<b>1.72%</b>	<b>1.00%</b>							
	PVP:	62.96%	66.02%	67.08%	65.93%	63.12%	63.76%	63.61%	66.05%							
Práctico	PVN:	31.25%	29.78%	28.76%	29.53%	31.14%	31.37%	30.24%	28.38%							
	FPF:	1.29%	1.31%	2.12%	1.98%	1.98%	1.62%	1.85%	2.75%							
	PFN:	4.51%	2.88%	2.04%	2.57%	3.75%	3.25%	4.29%	2.83%							
	Efectividad:	<b>94.21%</b>	<b>95.81%</b>	<b>95.83%</b>	<b>95.45%</b>	<b>94.26%</b>	<b>95.13%</b>	<b>93.86%</b>	<b>94.42%</b>							
	Desv. Std:	<b>0.66%</b>	<b>0.55%</b>	<b>0.61%</b>	<b>0.94%</b>	<b>1.25%</b>	<b>0.44%</b>	<b>1.37%</b>	<b>0.78%</b>							
Teórico	Tiempo (s)	<b>188.81</b>	<b>205.94</b>	<b>286.11</b>	<b>332.62</b>	<b>101.38</b>	<b>104.12</b>	<b>137.44</b>	<b>162.31</b>							



Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos.    **PVN:** Proporción de Verdaderos Negativos.  
**PPF:** Proporción de Falsos Positivos.      **PFN:** Proporción de Falsos Positivos.

TABLA B.16: Resultados por el método de muestreo Hammersley con parámetros optimizados por el método de recocido simulado con 230 iris para 100% y 90% del tamaño de la base de datos

No. subimágenes Tamaño de la subimagen	Tamaño de la base de datos UBIRIS									
	100%					90%				
	169 29x98 (píxeles)	150 16x5 (píxeles)	141 35x13 (píxeles)	172 42x5 (píxeles)	127 38x5 (píxeles)	156 33x5 (píxeles)	149 44x16 (píxeles)	115 48x13 (píxeles)		
<b>Práctico</b>	PVP: 58.20%	53.68%	57.18%	57.69%	57.92%	58.53%	59.29%	58.38%		
	PVN: 33.86%	36.62%	35.08%	34.32%	34.59%	33.88%	33.71%	35.05%		
	PPF: 1.57%	2.34%	0.28%	1.09%	0.43%	1.29%	2.23%	0.84%		
	PFN: 6.37%	7.36%	7.46%	6.90%	7.06%	6.29%	4.77%	5.74%		
	<b>Efectividad: 92.06%</b>	<b>90.31%</b>	<b>92.26%</b>	<b>92.01%</b>	<b>92.51%</b>	<b>92.41%</b>	<b>93.00%</b>	<b>93.43%</b>		
	<b>Desv. Std: 0.80%</b>	<b>0.82%</b>	<b>1.11%</b>	<b>1.41%</b>	<b>1.12%</b>	<b>1.61%</b>	<b>1.25%</b>	<b>0.42%</b>		
<b>Teórico</b>	PVP: 57.36%	54.11%	57.87%	57.69%	57.85%	58.18%	57.38%	57.07%		
	PVN: 35.64%	37.18%	35.73%	35.79%	35.61%	35.35%	36.69%	37.07%		
	PPF: 1.14%	3.06%	1.04%	1.02%	0.80%	1.13%	0.66%	0.29%		
	PFN: 5.86%	5.65%	5.37%	5.50%	5.75%	5.35%	5.27%	5.57%		
	<b>Efectividad: 93.00%</b>	<b>91.29%</b>	<b>93.60%</b>	<b>93.48%</b>	<b>93.45%</b>	<b>93.52%</b>	<b>94.07%</b>	<b>94.15%</b>		
	<b>Desv. Std: 0.44%</b>	<b>0.65%</b>	<b>0.68%</b>	<b>0.43%</b>	<b>0.60%</b>	<b>0.84%</b>	<b>0.47%</b>	<b>0.76%</b>		
	<b>Tiempo (s) 488.488</b>	<b>729.55</b>	<b>486.948</b>	<b>639.738</b>	<b>406.34</b>	<b>547.22</b>	<b>578.29</b>	<b>457.24</b>		
<b>Práctico</b>	PVP: 64.91%	60.12%	66.52%	66.96%	66.58%	66.38%	66.38%	64.70%		
	PVN: 28.80%	31.05%	28.98%	28.92%	28.51%	28.14%	28.54%	29.09%		
	PPF: 1.47%	2.25%	0.92%	1.07%	1.30%	1.67%	1.36%	1.15%		
	PFN: 4.82%	6.58%	3.58%	3.06%	3.61%	3.81%	3.72%	5.05%		
	<b>Efectividad: 93.71%</b>	<b>91.17%</b>	<b>95.50%</b>	<b>95.87%</b>	<b>95.09%</b>	<b>94.52%</b>	<b>94.92%</b>	<b>93.80%</b>		
	<b>Desv. Std: 0.80%</b>	<b>0.52%</b>	<b>0.73%</b>	<b>0.93%</b>	<b>0.76%</b>	<b>0.90%</b>	<b>0.93%</b>	<b>2.04%</b>		
<b>Teórico</b>	PVP: 65.13%	59.87%	66.00%	66.13%	66.06%	65.77%	65.94%	65.25%		
	PVN: 29.59%	32.29%	29.87%	30.36%	29.50%	30.04%	29.68%	30.26%		
	PPF: 1.94%	2.33%	1.29%	0.88%	1.60%	1.03%	1.48%	1.27%		
	PFN: 3.34%	5.52%	2.86%	2.64%	2.84%	3.16%	2.91%	3.22%		
	<b>Efectividad: 94.72%</b>	<b>92.15%</b>	<b>95.86%</b>	<b>96.48%</b>	<b>95.56%</b>	<b>95.81%</b>	<b>95.61%</b>	<b>95.51%</b>		
	<b>Desv. Std: 0.25%</b>	<b>0.59%</b>	<b>0.56%</b>	<b>0.83%</b>	<b>0.39%</b>	<b>0.92%</b>	<b>0.46%</b>	<b>0.73%</b>		
	<b>Tiempo (s) 430.144</b>	<b>402.36</b>	<b>417.142</b>	<b>542.146</b>	<b>354.574</b>	<b>312.29</b>	<b>302.45</b>	<b>250.04</b>		

Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos.      **PVN:** Proporción de Verdaderos Negativos.  
**PPF:** Proporción de Falsos Positivos.        **PFN:** Proporción de Falsos Positivos.

TABLA B.17: Resultados por el método de muestreo Hammersley con parámetros optimizados por el método de recocido simulado con 230 iris para 80% y 70% del tamaño de la base de datos

No. subimágenes	Tamaño de la subimagen	Tamaño de la base de datos UBIRIS									
		80%					70%				
		169	150	141	172	127	156	149	115		
Práctico	PVP:	69.33%	65.00%	69.70%	68.56%	69.13%	70.24%	69.60%	68.29%		
	PVN:	29.66%	29.53%	28.79%	30.47%	30.34%	29.13%	28.93%	30.50%		
	PPF:	0.17%	1.01%	0.64%	0.07%	0.13%	0.17%	0.87%	0.27%		
	PFN:	0.84%	4.46%	0.87%	0.91%	0.40%	0.47%	0.60%	0.94%		
	Efectividad:	<b>98.99%</b>	<b>94.53%</b>	<b>98.49%</b>	<b>99.03%</b>	<b>99.46%</b>	<b>99.36%</b>	<b>98.52%</b>	<b>98.79%</b>		
	Desv. Std:	<b>0.62%</b>	<b>0.35%</b>	<b>1.33%</b>	<b>1.17%</b>	<b>0.30%</b>	<b>0.28%</b>	<b>0.62%</b>	<b>1.11%</b>		
	Teórico	PVP:	68.45%	64.50%	68.78%	67.87%	67.93%	69.11%	68.16%	67.54%	
		PVN:	30.80%	30.71%	30.56%	31.72%	31.69%	30.58%	31.02%	31.71%	
		PPF:	0.36%	1.14%	0.22%	0.09%	0.06%	0.08%	0.10%	0.33%	
		PFN:	0.39%	3.65%	0.44%	0.31%	0.31%	0.23%	0.71%	0.42%	
Efectividad:		<b>99.25%</b>	<b>95.20%</b>	<b>99.34%</b>	<b>99.59%</b>	<b>99.62%</b>	<b>99.69%</b>	<b>99.18%</b>	<b>99.25%</b>		
Desv. Std:		<b>0.49%</b>	<b>0.09%</b>	<b>0.26%</b>	<b>0.28%</b>	<b>0.18%</b>	<b>0.25%</b>	<b>0.26%</b>	<b>0.62%</b>		
Tiempo (s)		<b>382.91</b>	<b>692.9</b>	<b>356.554</b>	<b>458.194</b>	<b>301.202</b>	<b>515.98</b>	<b>505.85</b>	<b>368.01</b>		
Práctico		PVP:	68.62%	66.27%	68.70%	67.03%	67.78%	69.58%	66.87%	67.19%	
		PVN:	30.98%	31.42%	30.62%	32.34%	31.74%	30.06%	32.26%	32.22%	
		PPF:	0.00%	0.76%	0.04%	0.00%	0.00%	0.00%	0.04%	0.00%	
	PFN:	0.40%	1.56%	0.64%	0.64%	0.48%	0.36%	0.84%	0.60%		
	Efectividad:	<b>99.60%</b>	<b>97.69%</b>	<b>99.32%</b>	<b>99.36%</b>	<b>99.52%</b>	<b>99.64%</b>	<b>99.12%</b>	<b>99.40%</b>		
	Desv. Std:	<b>0.79%</b>	<b>0.73%</b>	<b>0.54%</b>	<b>0.59%</b>	<b>0.44%</b>	<b>0.26%</b>	<b>0.90%</b>	<b>0.71%</b>		
	Teórico	PVP:	67.72%	65.24%	67.97%	66.48%	67.02%	68.60%	66.26%	66.52%	
		PVN:	32.25%	32.82%	31.96%	33.52%	32.94%	31.40%	33.48%	33.41%	
		PPF:	0.01%	0.55%	0.00%	0.00%	0.02%	0.00%	0.01%	0.00%	
		PFN:	0.03%	1.39%	0.07%	0.00%	0.02%	0.00%	0.26%	0.07%	
Efectividad:		<b>99.96%</b>	<b>98.06%</b>	<b>99.92%</b>	<b>100.00%</b>	<b>99.96%</b>	<b>100.00%</b>	<b>99.74%</b>	<b>99.92%</b>		
Desv. Std:		<b>0.08%</b>	<b>0.58%</b>	<b>0.10%</b>	<b>0.00%</b>	<b>0.08%</b>	<b>0.00%</b>	<b>0.21%</b>	<b>0.10%</b>		
Tiempo (s)		<b>337.216</b>	<b>581.12</b>	<b>315.018</b>	<b>384.34</b>	<b>249.304</b>	<b>433.45</b>	<b>425.57</b>	<b>308.17</b>		



Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos.

**PVN:** Proporción de Verdaderos Negativos.

**PFP:** Proporción de Falsos Positivos.

**PFN:** Proporción de Falsos Positivos.

**1er:** Combinaciones del primer muestreo.

**2do:** Combinaciones del segundo muestreo.

TABLA B.19: Resultados por el método de muestreo Hammersley factorial para 100% y 90% del tamaño de la base de datos (continuación).

No. subimágenes	149 subimágenes de 44-16 pixeles					115 subimágenes de 48x13 pixeles					2do.	
	Tamaño de la subimagen	46	57	73	99	46	57	73	99	1er.		
		41x20 (pixeles)	31x5 (pixeles)	21x5 (pixeles)	5x15 (pixeles)	41x20 (pixeles)	31x5 (pixeles)	21x5 (pixeles)	5x15 (pixeles)	99	5x15 (pixeles)	
Práctico	PVP:	41.19%	38.76%	38.86%	36.27%	39.47%	40.96%	38.58%	40.41%			
	PVN:	58.60%	61.09%	61.04%	63.55%	60.36%	58.78%	61.19%	59.39%			
	PFP:	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%			
	PFN:	0.20%	0.15%	0.10%	0.18%	0.18%	0.25%	0.23%	0.20%			
	Efectividad:	<b>99.80%</b>	<b>99.85%</b>	<b>99.90%</b>	<b>99.82%</b>	<b>99.82%</b>	<b>99.75%</b>	<b>99.77%</b>	<b>99.80%</b>			
Teórico	Desv. Std:	<b>0.11%</b>	<b>0.14%</b>	<b>0.14%</b>	<b>0.11%</b>	<b>0.11%</b>	<b>0.06%</b>	<b>0.11%</b>	<b>0.11%</b>			
	PVP:	41.19%	38.76%	38.86%	36.27%	39.62%	40.96%	38.58%	40.41%			
	PVN:	58.60%	61.09%	61.04%	63.55%	60.21%	58.78%	61.19%	59.39%			
	PFP:	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%			
	PFN:	0.20%	0.15%	0.10%	0.18%	0.17%	0.25%	0.23%	0.20%			
Práctico	Efectividad:	<b>99.80%</b>	<b>99.85%</b>	<b>99.90%</b>	<b>99.82%</b>	<b>99.83%</b>	<b>99.75%</b>	<b>99.77%</b>	<b>99.80%</b>			
	Desv. Std:	<b>0.11%</b>	<b>0.14%</b>	<b>0.14%</b>	<b>0.11%</b>	<b>0.11%</b>	<b>0.06%</b>	<b>0.11%</b>	<b>0.11%</b>			
	Tiempo (s)	217.44	251.06	257.01	281.4	253.31	283.24	353.66	516.35			
	PVP:	37.92%	40.78%	42.22%	43.32%	44.53%	42.05%	44.21%	38.82%			
	PVN:	62.08%	59.22%	57.78%	56.65%	55.47%	57.95%	55.79%	61.18%			
Teórico	PFP:	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%			
	PFN:	0.00%	0.00%	0.00%	0.03%	0.00%	0.00%	0.00%	0.00%			
	Efectividad:	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>99.97%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>			
	Desv. Std:	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.06%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>			
	PVP:	38.12%	41.06%	42.35%	43.32%	44.63%	42.11%	44.36%	39.19%			
Teórico	PVN:	61.88%	58.94%	57.65%	56.65%	55.37%	57.89%	55.64%	60.81%			
	PFP:	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%			
	PFN:	0.00%	0.00%	0.00%	0.03%	0.00%	0.00%	0.00%	0.00%			
	Efectividad:	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>99.97%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>			
	Desv. Std:	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.06%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>	<b>0.00%</b>			
Tiempo (s)	193.73	212.55	228.39	269.47	224.4	253.77	314.13	451.09				



## B.6. Método ARX

Nomenclatura:

**PVP:** Proporción de Verdaderos Positivos.      **PVN:** Proporción de Verdaderos Negativos.  
**PFP:** Proporción de Falsos Positivos.        **PFN:** Proporción de Falsos Positivos.  
**1er:** Combinaciones del primer muestreo.    **2do:** Combinaciones del segundo muestreo.

TABLA B.20: Resultados por el método ARX para 100%, 90%, 70%, 60% y 50% del tamaño de la base de datos con  $na = 5$ ,  $nb = 5$  y  $grid = 4$ .

na=5, nb=5		100%	90%	80%	70%	60%	50%	grid
		UBIRIS	UBIRIS	UBIRIS	UBIRIS	UBIRIS	UBIRIS	
Práctico	PVP:	47.34%	60.58%	64.72%	67.47%	67.16%	63.31%	4x4
	PVN:	36.46%	24.32%	26.59%	29.14%	29.88%	36.36%	
	PFP:	4.18%	6.62%	2.17%	3.19%	2.47%	0.00%	
	PFN:	12.03%	8.49%	6.52%	0.20%	0.49%	0.32%	
	<b>Efectividad:</b>	<b>83.80%</b>	<b>84.89%</b>	<b>91.30%</b>	<b>96.61%</b>	<b>97.04%</b>	<b>99.68%</b>	
Teórico	PVP:	49.88%	61.91%	66.38%	65.92%	64.98%	62.35%	4x4
	PVN:	34.76%	23.48%	27.04%	30.73%	32.95%	37.35%	
	PFP:	7.14%	9.00%	3.06%	2.98%	0.61%	0.07%	
	PFN:	8.21%	5.62%	3.53%	0.37%	1.46%	0.22%	
	<b>Efectividad:</b>	<b>84.64%</b>	<b>85.39%</b>	<b>93.42%</b>	<b>96.65%</b>	<b>97.93%</b>	<b>99.70%</b>	
<b>Tiempo (s)</b>		<b>9434.6</b>	<b>8247.8</b>	<b>6962.1</b>	<b>5872.2</b>	<b>4726.3</b>	<b>3672.3</b>	
Práctico	PVP:	47.72%	52.66%	64.88%	62.87%	61.48%	63.96%	5x5
	PVN:	37.47%	31.66%	26.76%	30.94%	34.82%	34.09%	
	PFP:	4.18%	2.73%	5.02%	0.80%	0.74%	0.32%	
	PFN:	10.63%	12.95%	3.34%	5.39%	2.96%	1.62%	
	<b>Efectividad:</b>	<b>85.19%</b>	<b>84.32%</b>	<b>91.64%</b>	<b>93.81%</b>	<b>96.30%</b>	<b>98.05%</b>	
Teórico	PVP:	47.86%	56.56%	63.79%	62.57%	62.76%	64.07%	5x5
	PVN:	38.45%	31.26%	27.90%	31.66%	35.17%	34.73%	
	PFP:	4.17%	4.74%	5.02%	1.49%	1.61%	0.90%	
	PFN:	9.52%	7.44%	3.29%	4.28%	0.46%	0.30%	
	<b>Efectividad:</b>	<b>86.31%</b>	<b>87.82%</b>	<b>91.69%</b>	<b>94.23%</b>	<b>97.93%</b>	<b>98.80%</b>	
<b>Tiempo (s)</b>		<b>10211</b>	<b>8998.1</b>	<b>7746.7</b>	<b>6476.2</b>	<b>5225.3</b>	<b>3988.5</b>	
Práctico	PVP:	35.19%	51.37%	55.18%	64.87%	62.22%	61.04%	8x8
	PVN:	41.90%	33.38%	30.27%	26.75%	31.36%	33.12%	
	PFP:	1.52%	4.75%	3.18%	4.99%	2.72%	5.19%	
	PFN:	21.39%	10.50%	11.37%	3.39%	3.70%	0.65%	
	<b>Efectividad:</b>	<b>77.09%</b>	<b>84.75%</b>	<b>85.45%</b>	<b>91.62%</b>	<b>93.58%</b>	<b>94.16%</b>	
Teórico	PVP:	44.29%	51.15%	57.48%	61.64%	62.45%	59.28%	8x8
	PVN:	38.93%	33.83%	29.20%	30.73%	31.80%	36.23%	
	PFP:	6.19%	5.41%	5.60%	2.79%	3.37%	2.99%	
	PFN:	10.60%	9.61%	7.73%	4.84%	2.38%	1.50%	
	<b>Efectividad:</b>	<b>83.21%</b>	<b>84.98%</b>	<b>86.68%</b>	<b>92.36%</b>	<b>94.25%</b>	<b>95.51%</b>	
<b>Tiempo (s)</b>		<b>4365.6</b>	<b>3939.1</b>	<b>3415.3</b>	<b>2772</b>	<b>2315.1</b>	<b>1712.3</b>	
Práctico	PVP:	42.66%	40.72%	59.20%	56.29%	58.03%	46.10%	10x10
	PVN:	35.44%	36.12%	20.07%	31.14%	26.91%	39.61%	
	PFP:	10.51%	3.31%	17.06%	5.79%	8.40%	0.00%	
	PFN:	11.39%	19.86%	3.68%	6.79%	6.67%	14.29%	
	<b>Efectividad:</b>	<b>78.10%</b>	<b>76.84%</b>	<b>79.26%</b>	<b>87.43%</b>	<b>84.94%</b>	<b>85.71%</b>	
Teórico	PVP:	38.39%	44.66%	52.67%	58.85%	55.40%	52.40%	10x10
	PVN:	41.73%	35.72%	30.88%	29.80%	31.26%	37.73%	
	PFP:	5.42%	4.60%	7.05%	8.19%	5.29%	2.99%	
	PFN:	14.46%	15.02%	9.40%	3.17%	8.05%	6.89%	
	<b>Efectividad:</b>	<b>80.12%</b>	<b>80.38%</b>	<b>83.54%</b>	<b>88.64%</b>	<b>86.67%</b>	<b>90.12%</b>	
<b>Tiempo (s)</b>		<b>2927.2</b>	<b>2590.8</b>	<b>2203.4</b>	<b>1789.1</b>	<b>1486.1</b>	<b>1135.3</b>	

# Resultados de la base de datos Miles Research (Verificación)

Para los métodos de muestreo se utilizó el mejor resultado que se obtuvo en modo identificación el cual fue de 115 parches de tamaño 48x13 píxeles. Para el ARX se realizaron los experimentos con unos valores de  $na = 5$  y  $nb = 5$ .

## C.1. Método de muestreo aleatorio con reposición

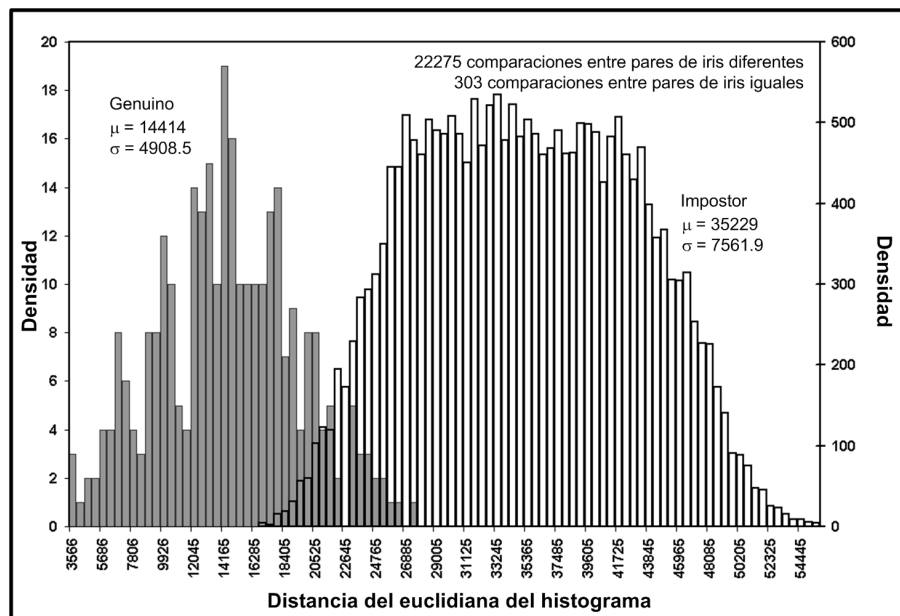


FIGURA C.1: Distribución genuino-impostor por el método de muestreo aleatorio con reposición para una base de datos de tamaño 100 %.

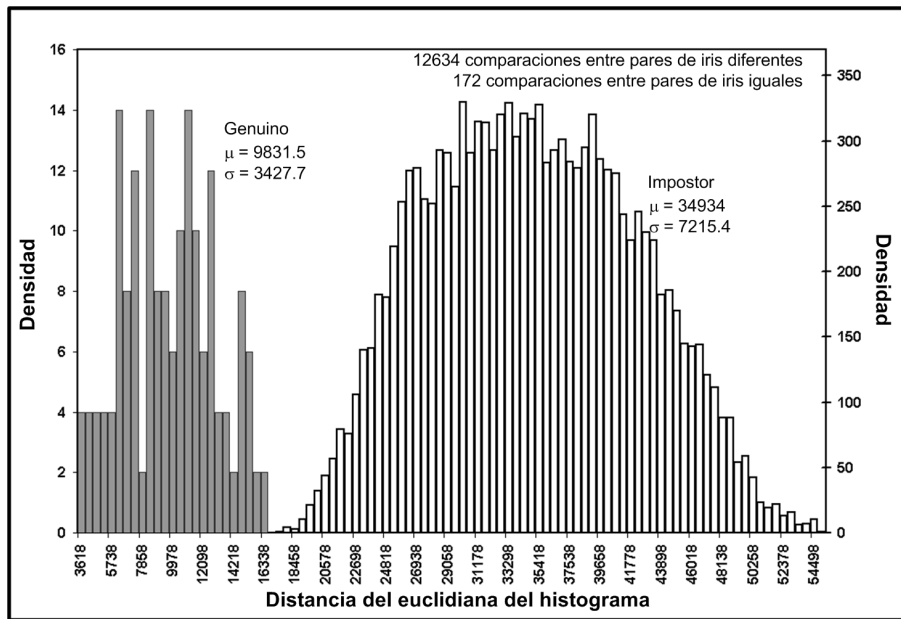


FIGURA C.2: Distribución genuino-impostor por el método de muestreo aleatorio con reposición para una base de datos de tamaño 75 %.

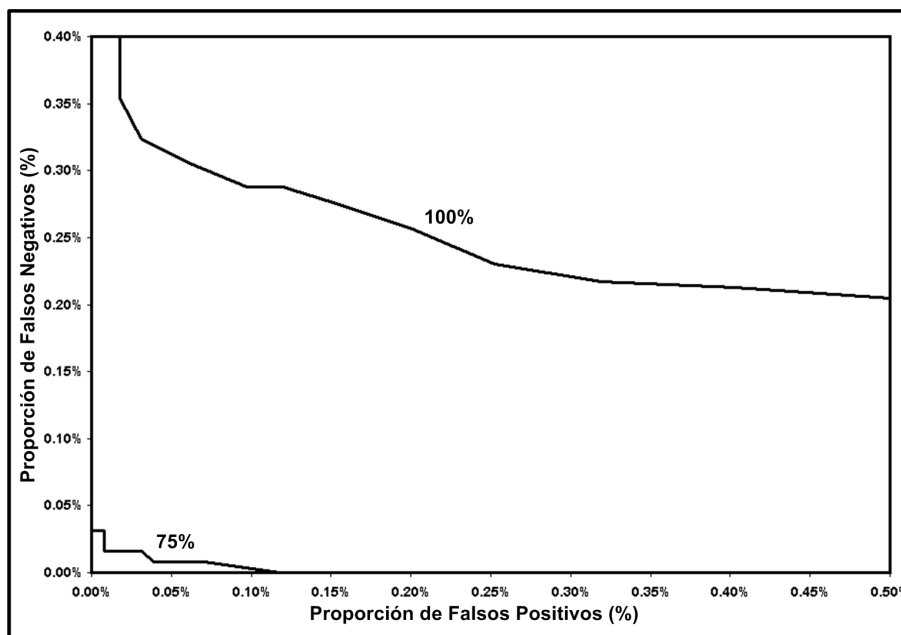


FIGURA C.3: Curva ROC por el método de muestreo aleatorio con reposición.

### C.2. Método de muestreo aleatorio sin reposición

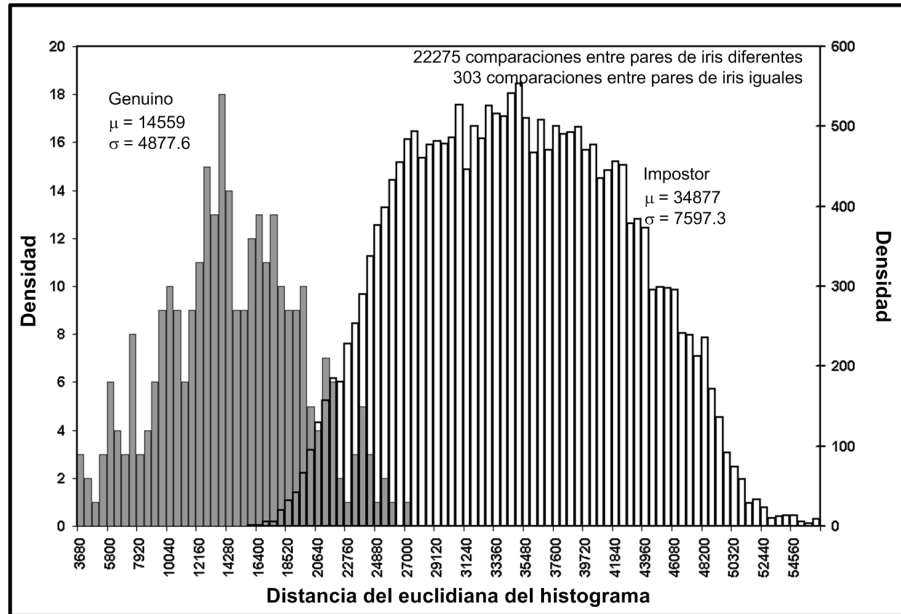


FIGURA C.4: Distribución genuino-impostor por el método de muestreo aleatorio sin reposición para una base de datos de tamaño 100 %.

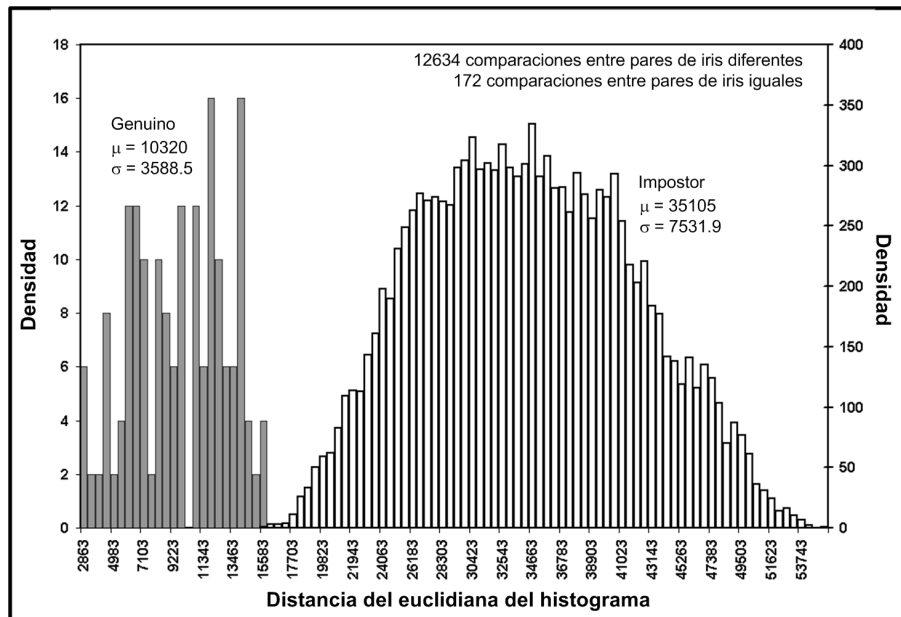


FIGURA C.5: Distribución genuino-impostor por el método de muestreo aleatorio sin reposición para una base de datos de tamaño 75 %.

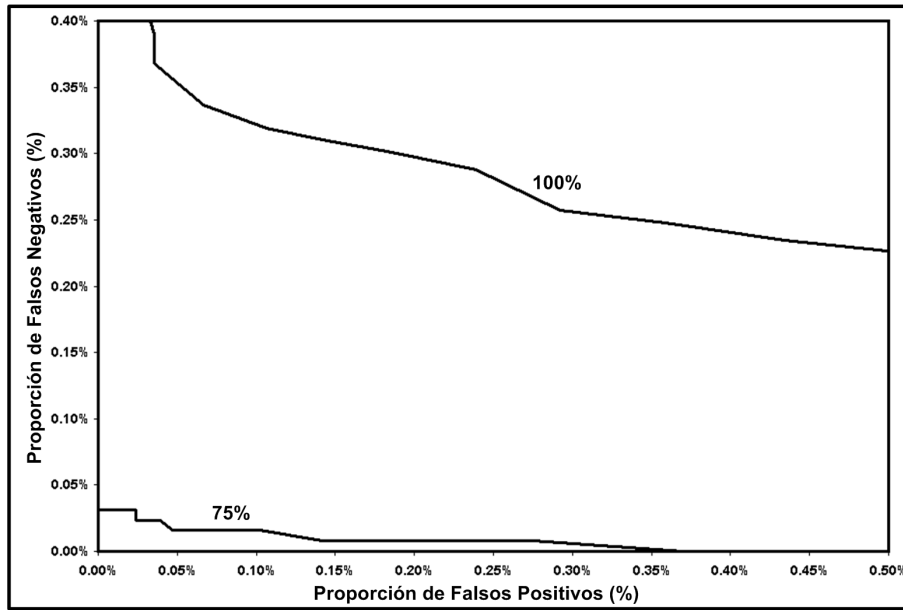


FIGURA C.6: Curva ROC por el método de muestreo aleatorio sin reposición.

### C.3. Método de muestreo Hammerlsey

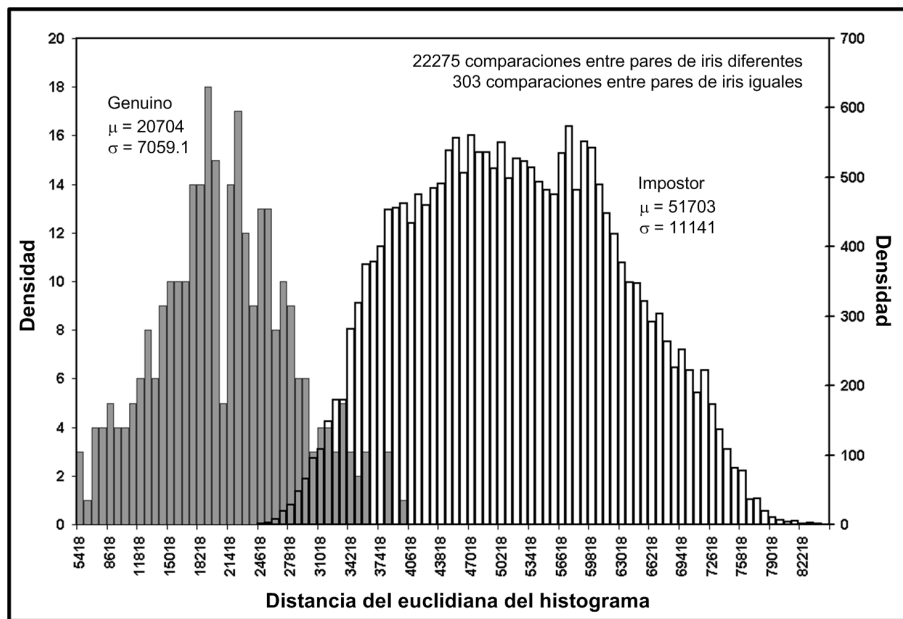


FIGURA C.7: Distribución genuino-impostor por el método de muestreo hammersley para una base de datos de tamaño 100 %.

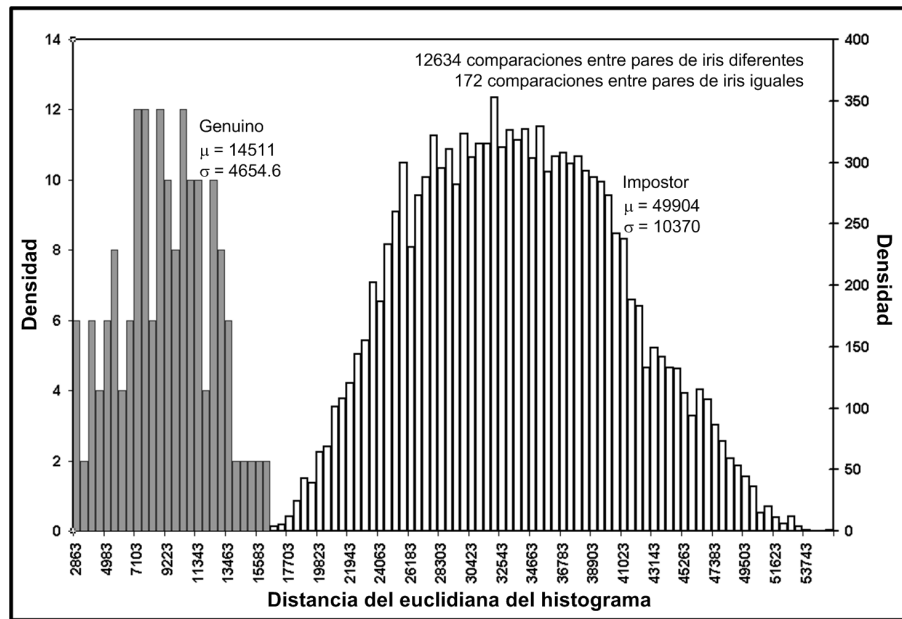


FIGURA C.8: Distribución genuino-impostor por el método de muestreo hammersley para una base de datos de tamaño 75 %.

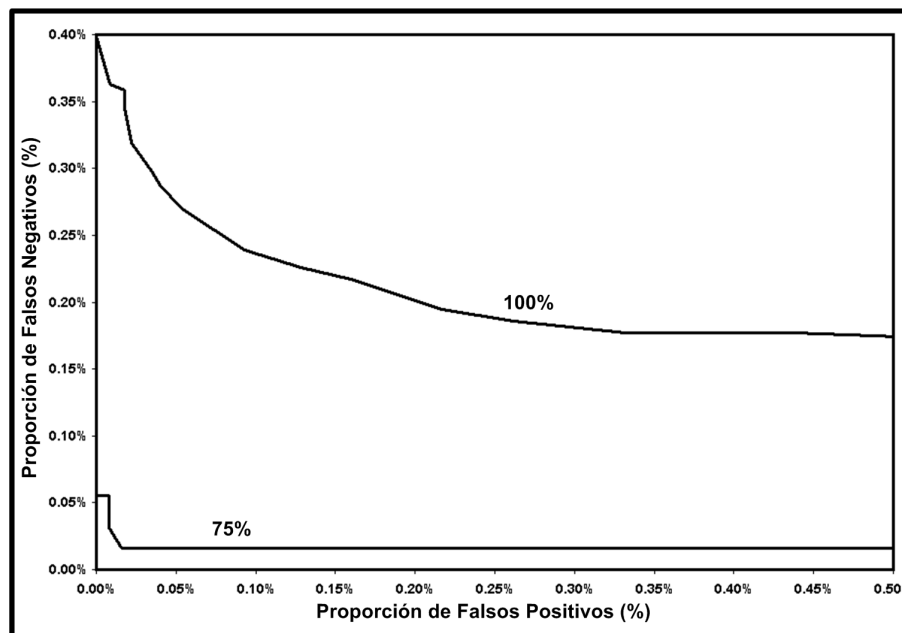


FIGURA C.9: Curva ROC por el método de muestreo hammersley.

## C.4. Método ARX

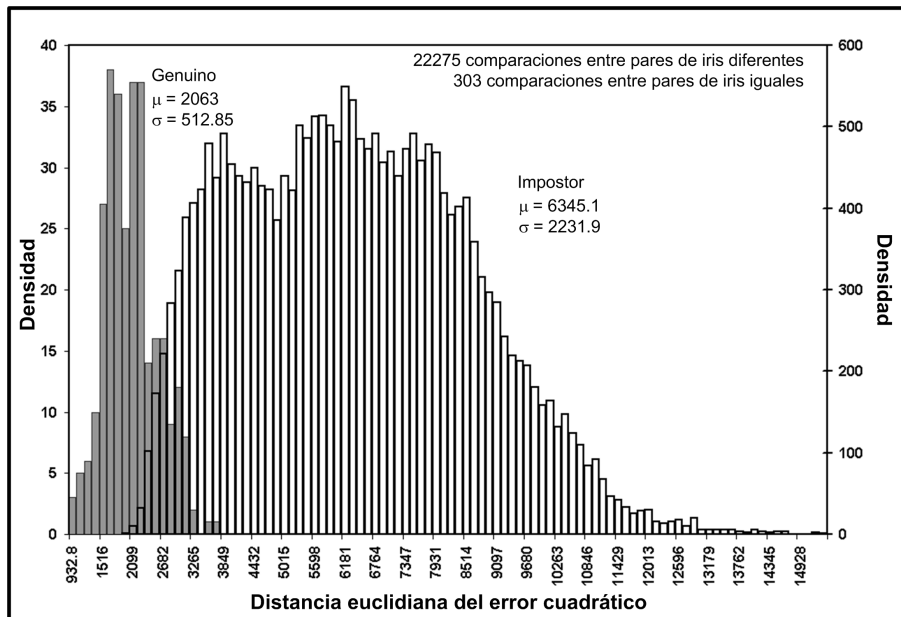


FIGURA C.10: Distribución genuino-impostor por el método de ARX para una base de datos de tamaño 100 %.

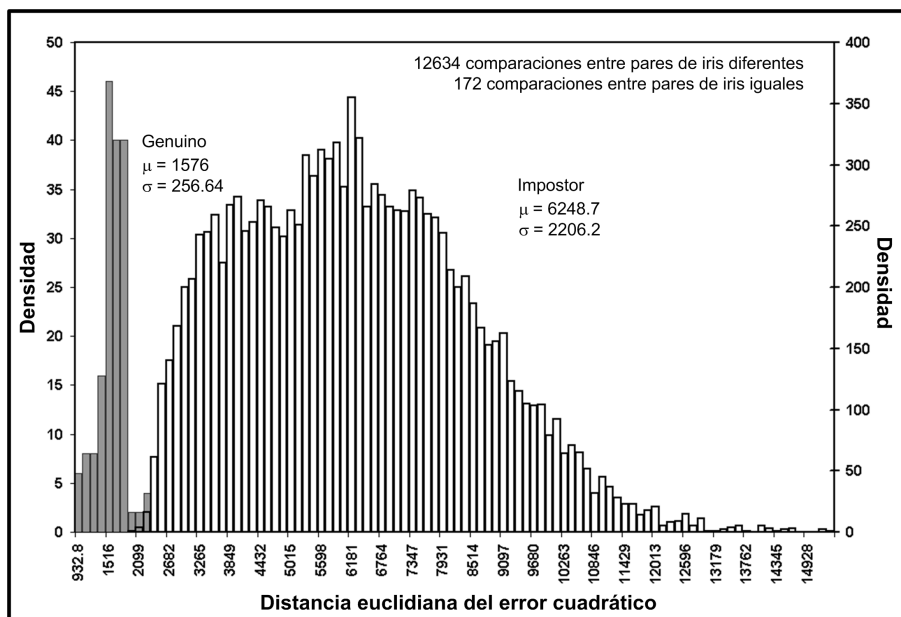


FIGURA C.11: Distribución genuino-impostor por el método de ARX para una base de datos de tamaño 75 %.

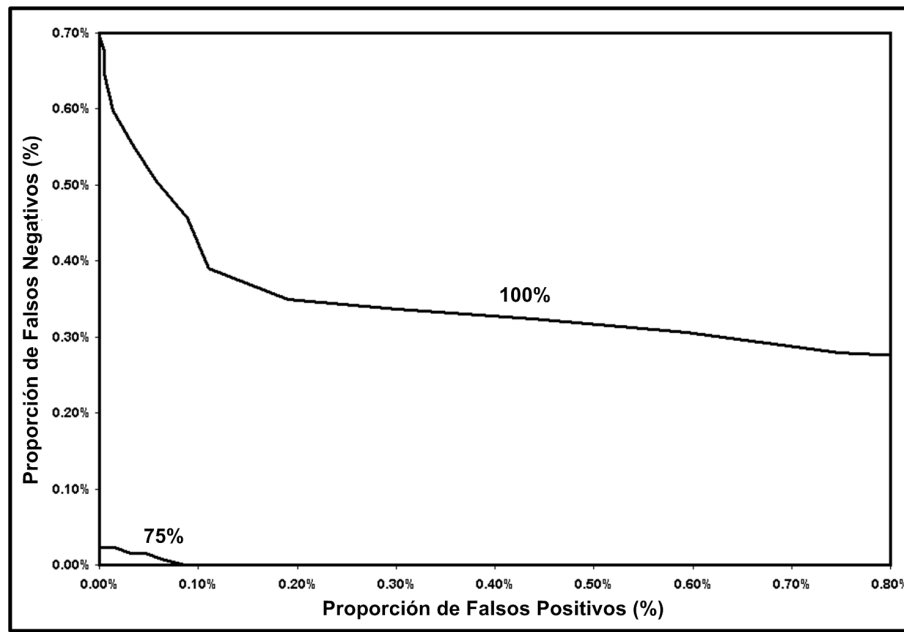


FIGURA C.12: Curva ROC por el método de ARX.





# Resultados de la base de datos UBIRIS (Verificación)

Para los métodos de muestreo se utilizó el mejor resultado que se obtuvo en modo identificación el cual fue de 115 parches de tamaño 48x13 píxeles. Para el ARX se realizaron los experimentos con unos valores de  $na = 5$  y  $nb = 5$ .

## D.1. Método de muestreo aleatorio con reposición

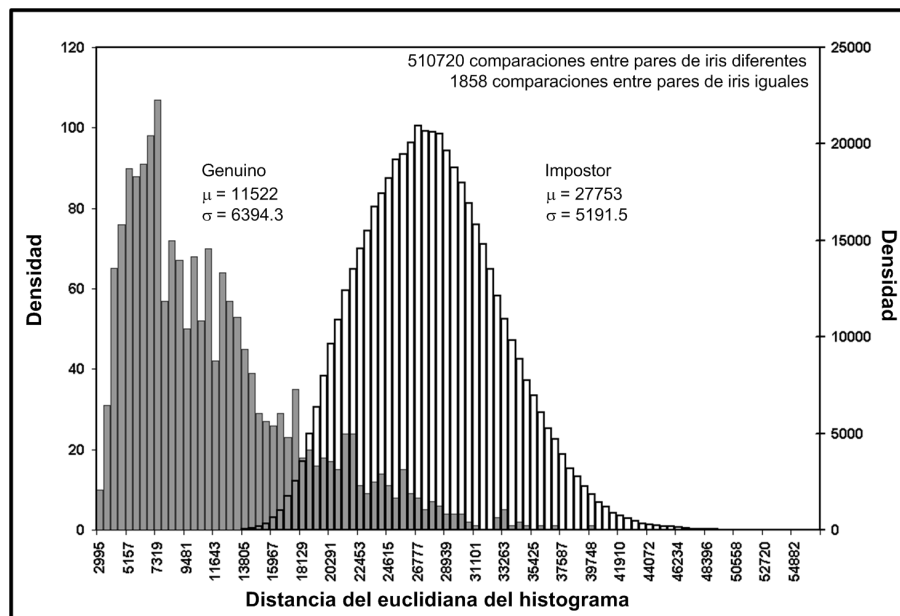


FIGURA D.1: Distribución genuino-impostor por el método de muestreo aleatorio con reposición para una base de datos de tamaño 100 %.

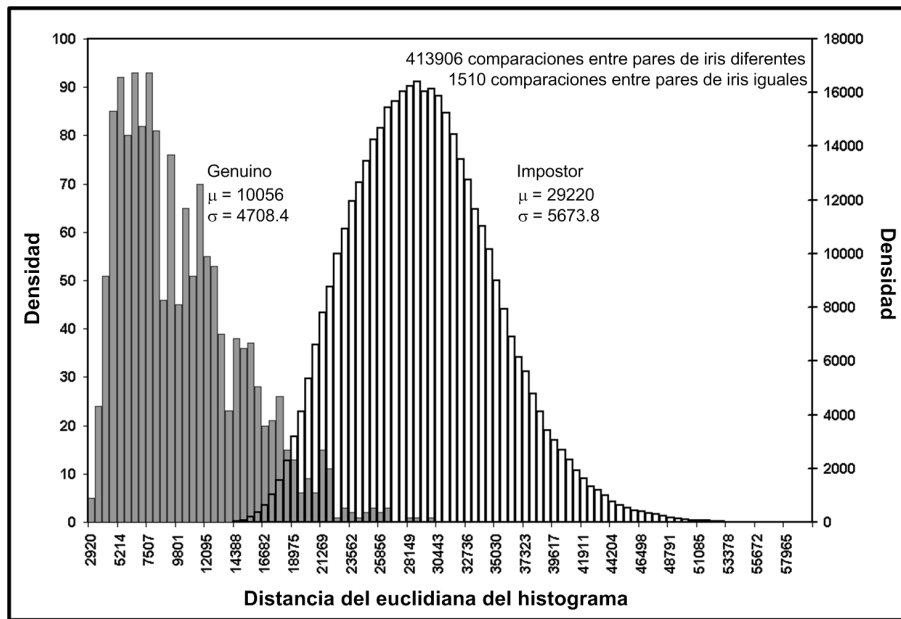


FIGURA D.2: Distribución genuino-impostor por el método de muestreo aleatorio con reposición para una base de datos de tamaño 90 %.

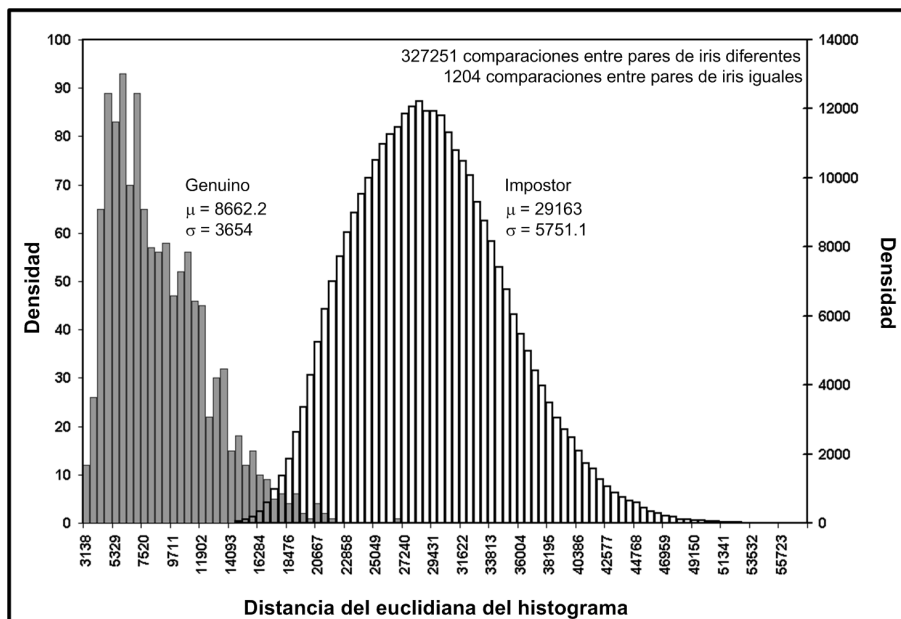


FIGURA D.3: Distribución genuino-impostor por el método de muestreo aleatorio con reposición para una base de datos de tamaño 80 %.

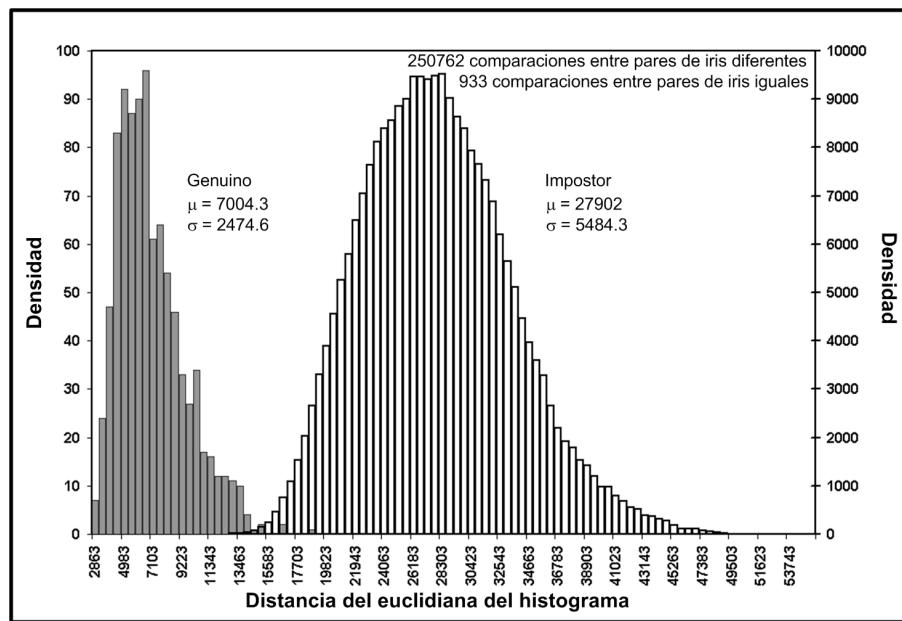


FIGURA D.4: Distribución genuino-impostor por el método de muestreo aleatorio con reposición para una base de datos de tamaño 70 %.

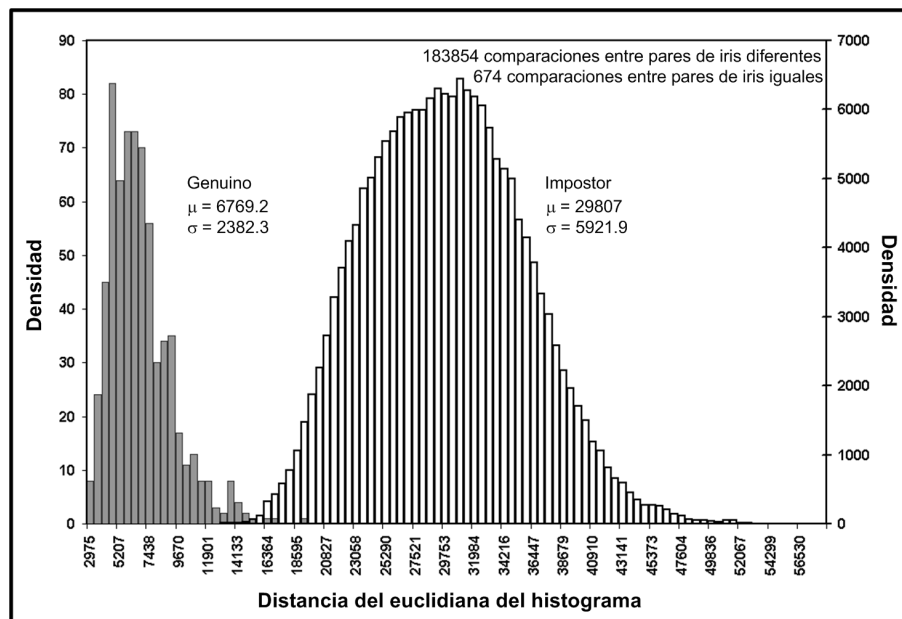


FIGURA D.5: Distribución genuino-impostor por el método de muestreo aleatorio con reposición para una base de datos de tamaño 60 %.

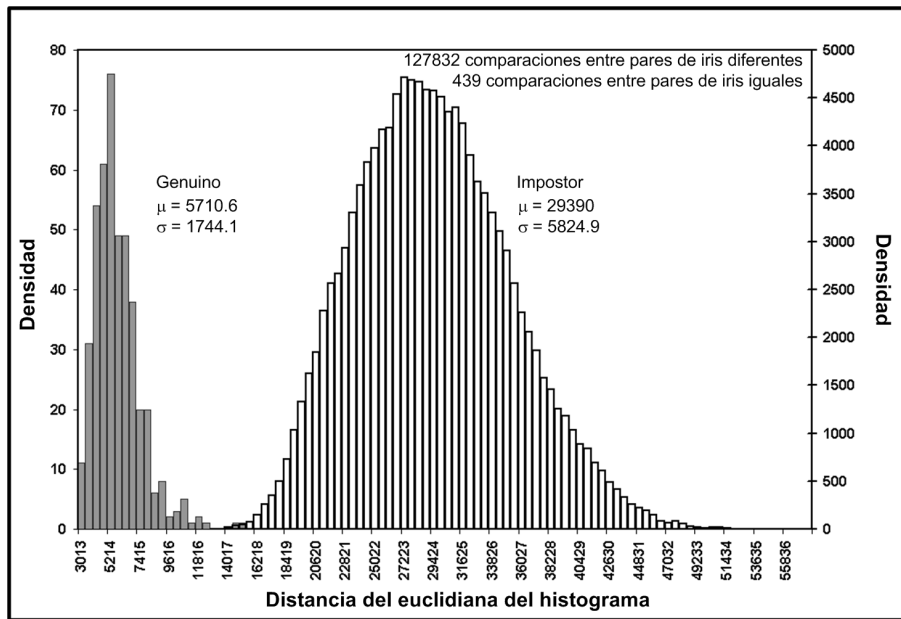


FIGURA D.6: Distribución genuino-impostor por el método de muestreo aleatorio con reposición para una base de datos de tamaño 50%.

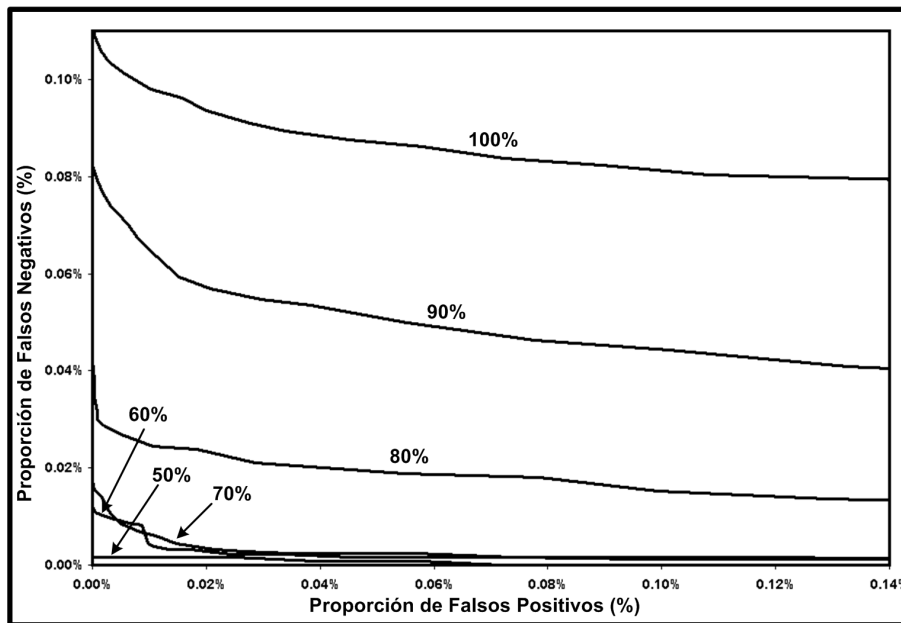


FIGURA D.7: Curva ROC por el método de muestreo aleatorio con reposición.

## D.2. Método de muestreo aleatorio sin reposición

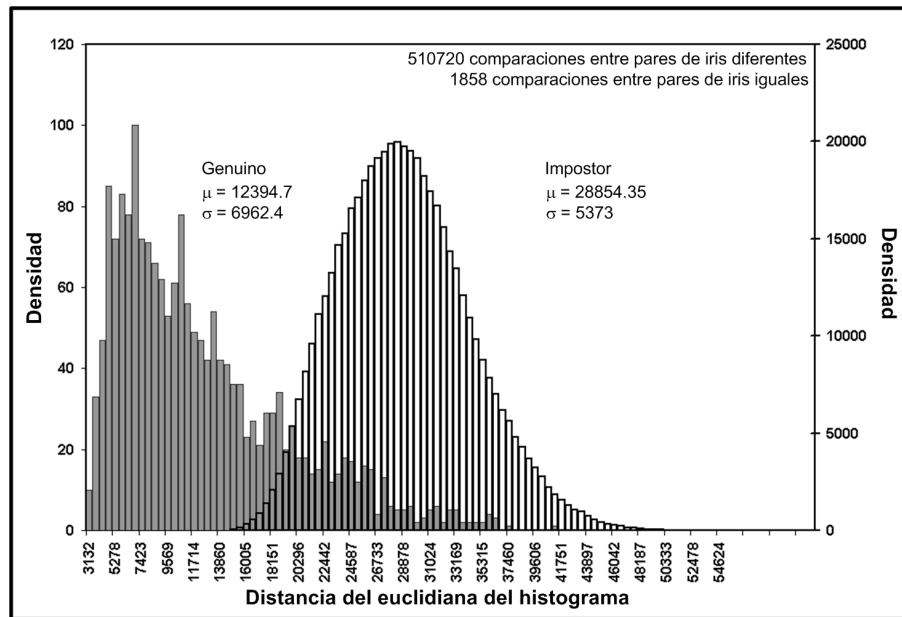


FIGURA D.8: Distribución genuino-impostor por el método de muestreo aleatorio sin reposición para una base de datos de tamaño 100 %.

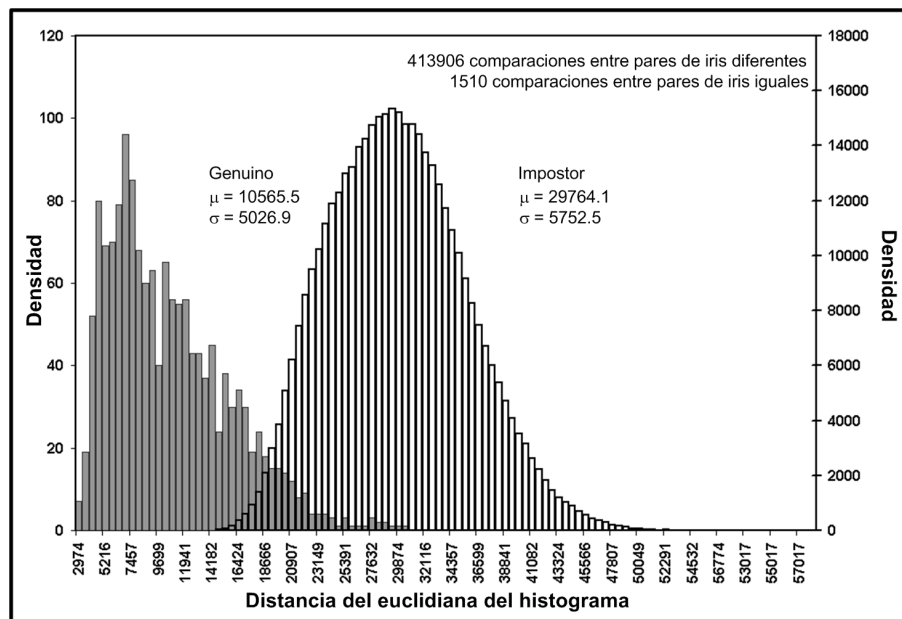


FIGURA D.9: Distribución genuino-impostor por el método de muestreo aleatorio sin reposición para una base de datos de tamaño 90 %.

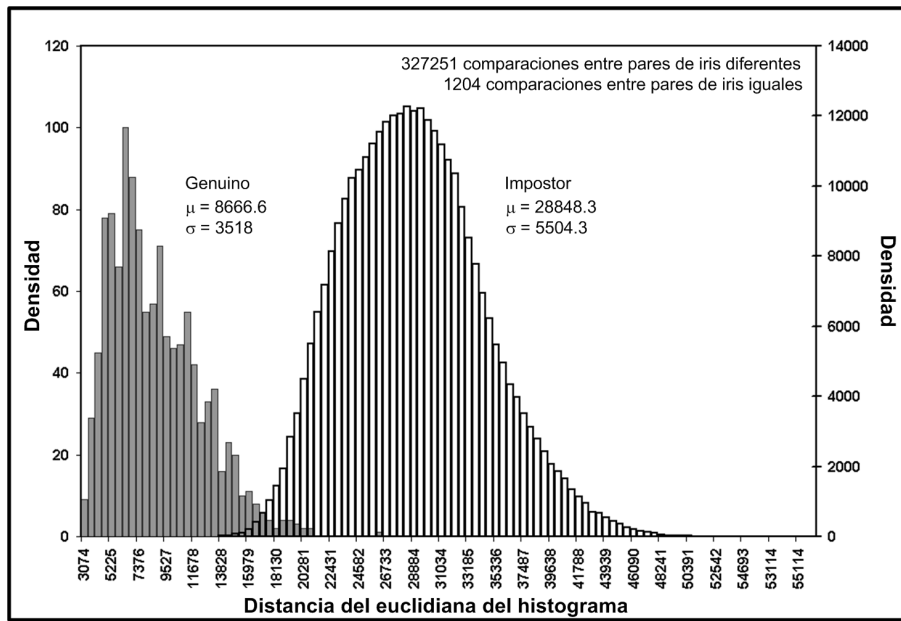


FIGURA D.10: Distribución genuino-impostor por el método de muestreo aleatorio sin reposición para una base de datos de tamaño 80 %.

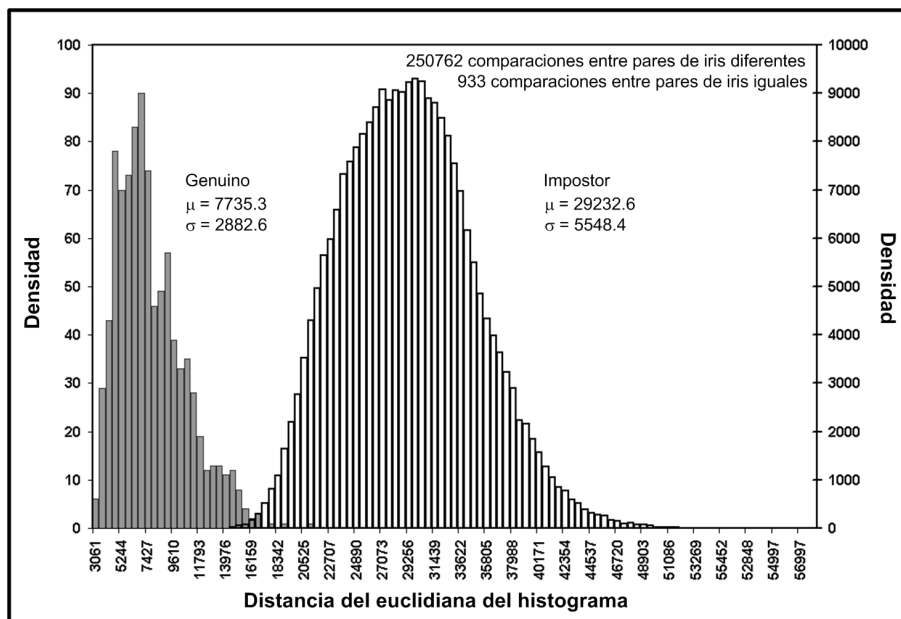


FIGURA D.11: Distribución genuino-impostor por el método de muestreo aleatorio sin reposición para una base de datos de tamaño 70 %.

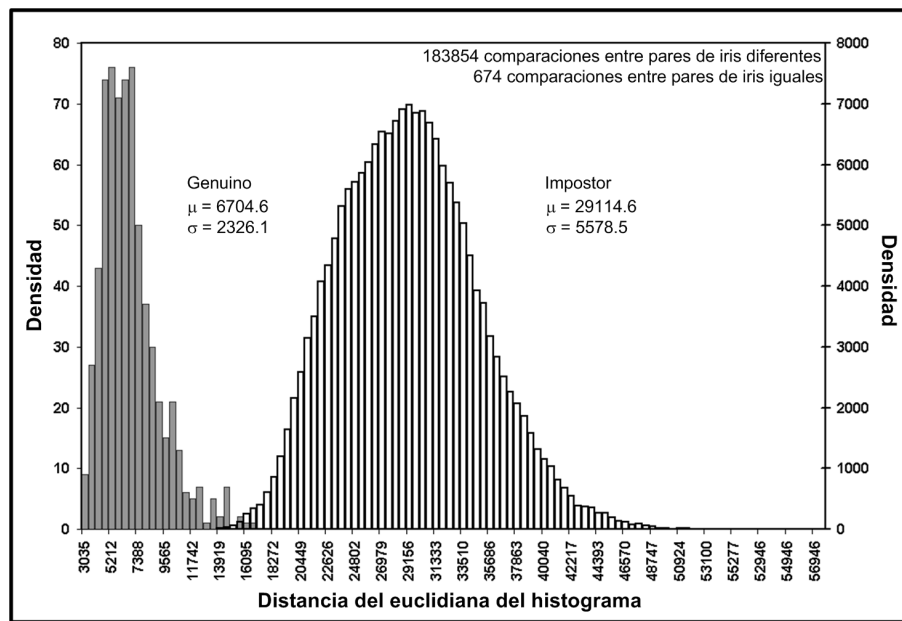


FIGURA D.12: Distribución genuino-impostor por el método de muestreo aleatorio sin reposición para una base de datos de tamaño 60 %.

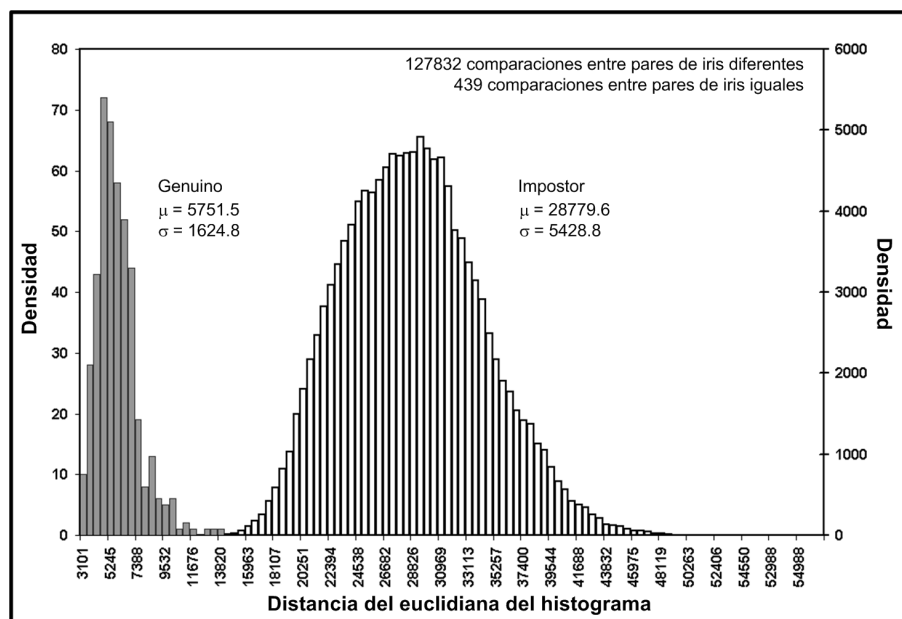


FIGURA D.13: Distribución genuino-impostor por el método de muestreo aleatorio sin reposición para una base de datos de tamaño 50 %.



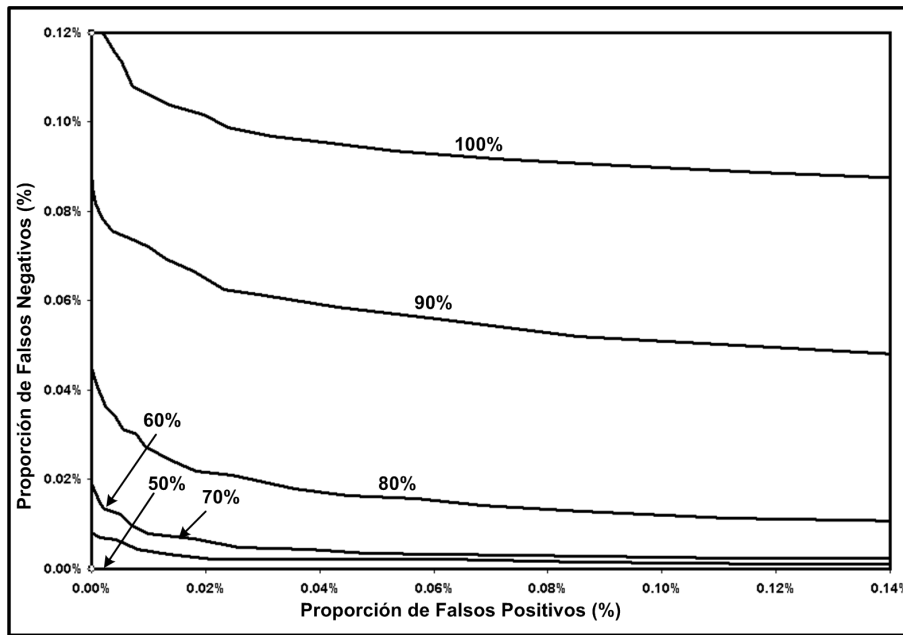


FIGURA D.14: Curva ROC por el método de muestreo aleatorio sin reposición.

### D.3. Método de muestreo Hammerlsey

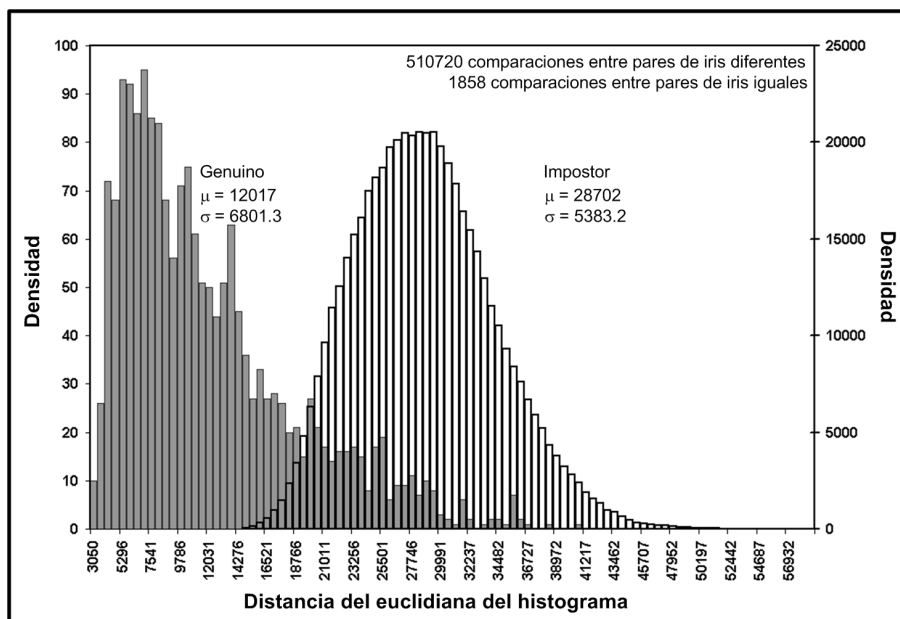


FIGURA D.15: Distribución genuino-impostor por el método de muestreo hammersley para una base de datos de tamaño 100 %.

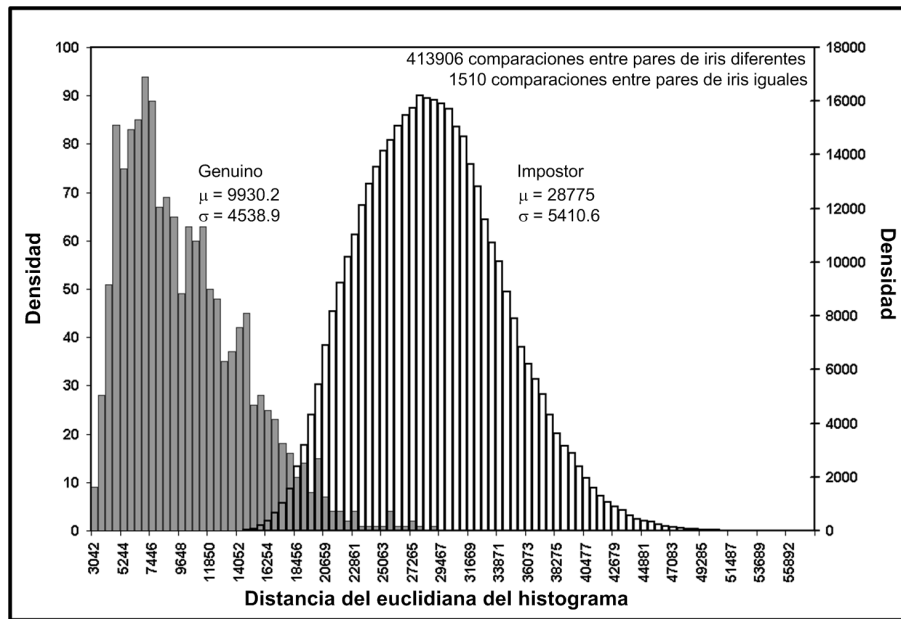


FIGURA D.16: Distribución genuino-impostor por el método de muestreo hammersley para una base de datos de tamaño 90 %.

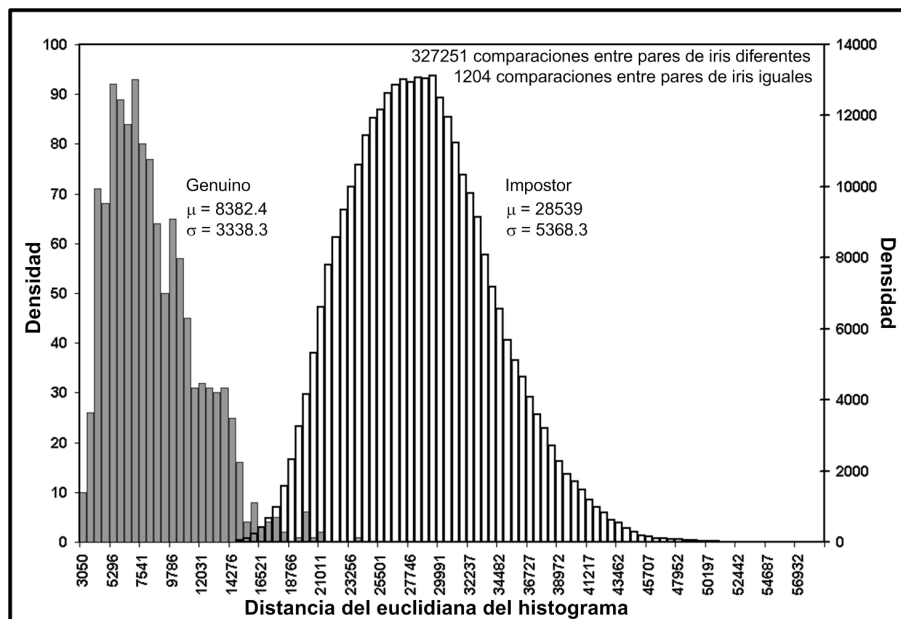


FIGURA D.17: Distribución genuino-impostor por el método de muestreo hammersley para una base de datos de tamaño 80 %.

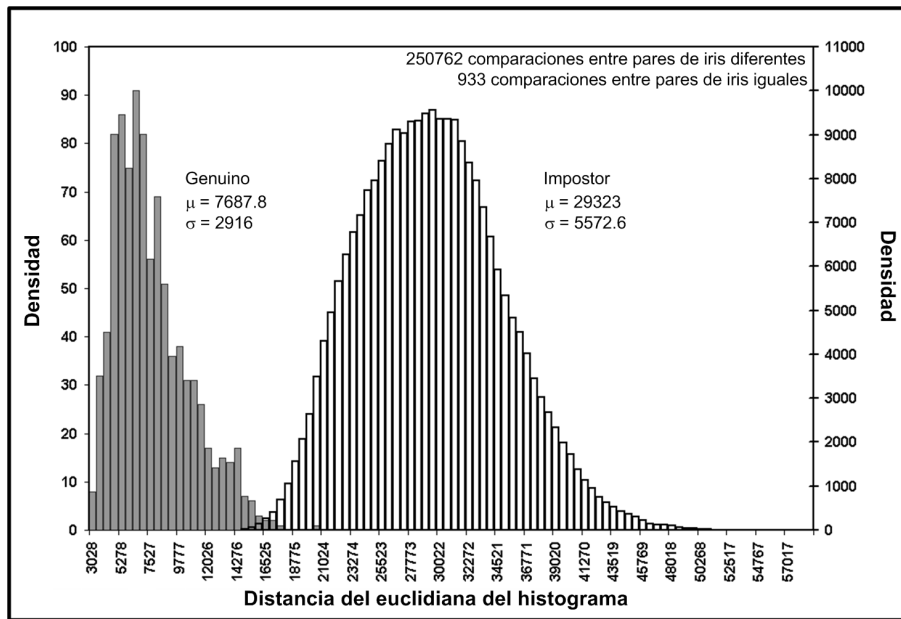


FIGURA D.18: Distribución genuino-impostor por el método de muestreo hammlerley para una base de datos de tamaño 70 %.

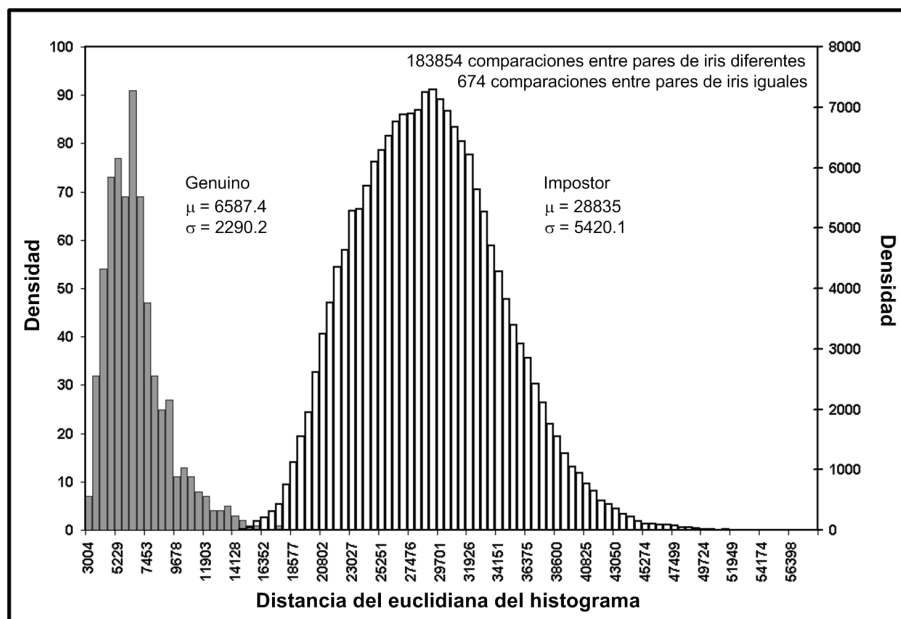


FIGURA D.19: Distribución genuino-impostor por el método de muestreo hammlerley para una base de datos de tamaño 60 %.

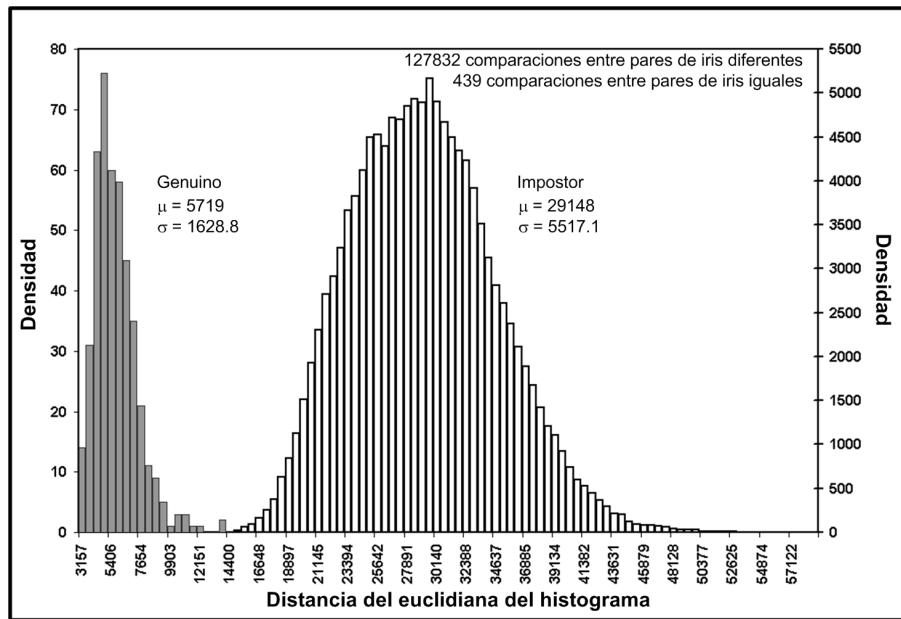


FIGURA D.20: Distribución genuino-impostor por el método de muestreo hammersley para una base de datos de tamaño 50 %.

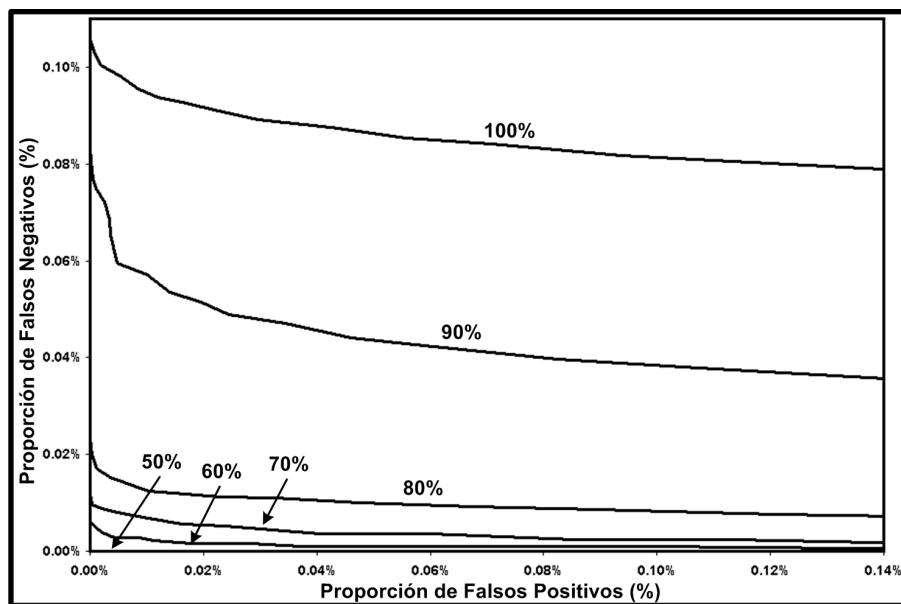


FIGURA D.21: Curva ROC por el método de muestreo hammersley.

## D.4. Método ARX

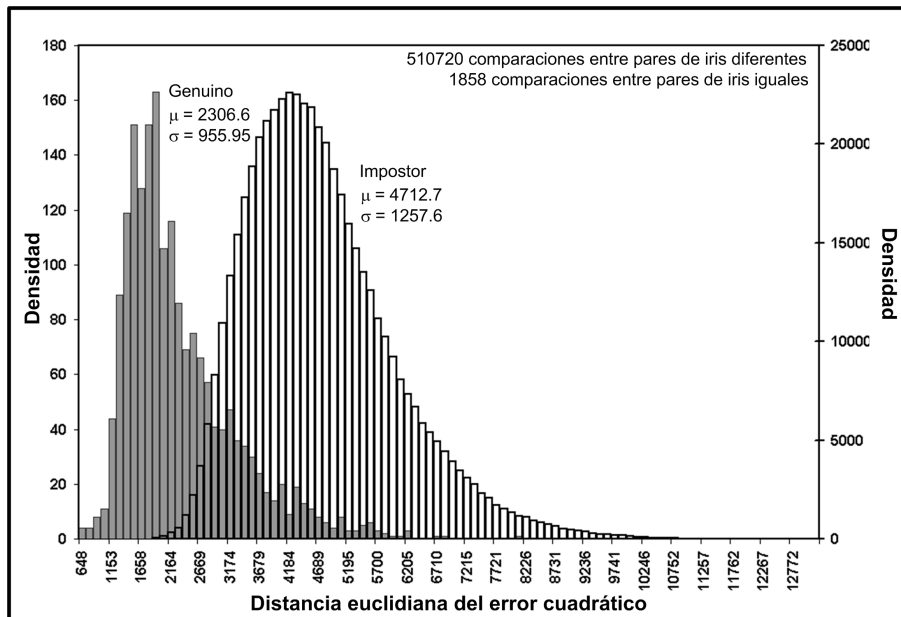


FIGURA D.22: Distribución genuino-impostor por el método de ARX para una base de datos de tamaño 100 %.

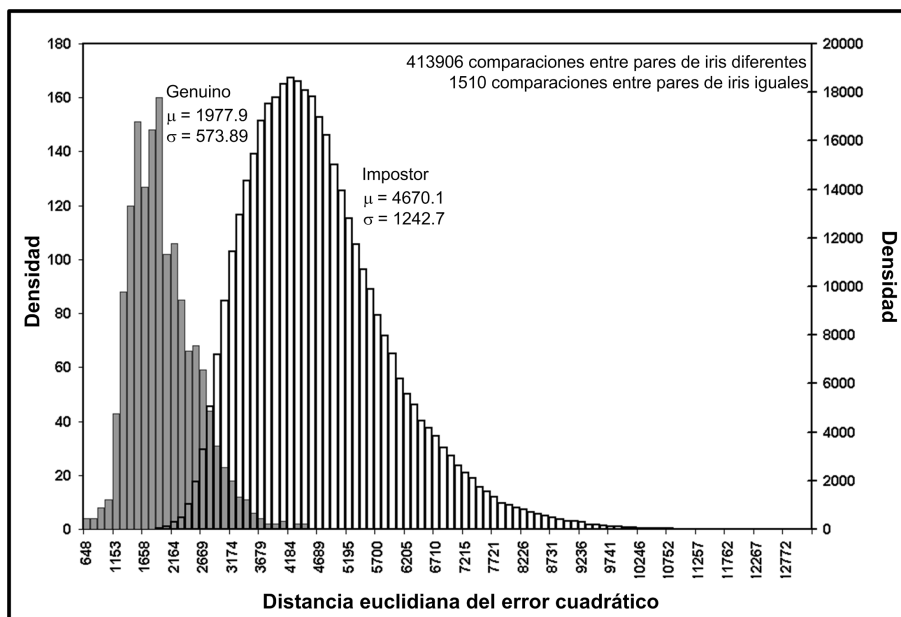


FIGURA D.23: Distribución genuino-impostor por el método de ARX para una base de datos de tamaño 90 %.

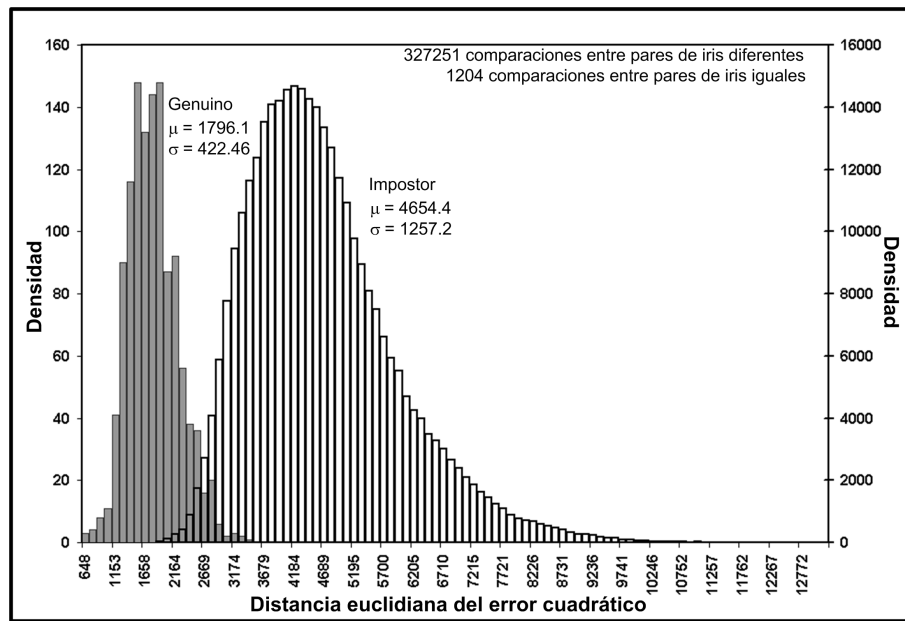


FIGURA D.24: Distribución genuino-impostor por el método de ARX para una base de datos de tamaño 80 %.

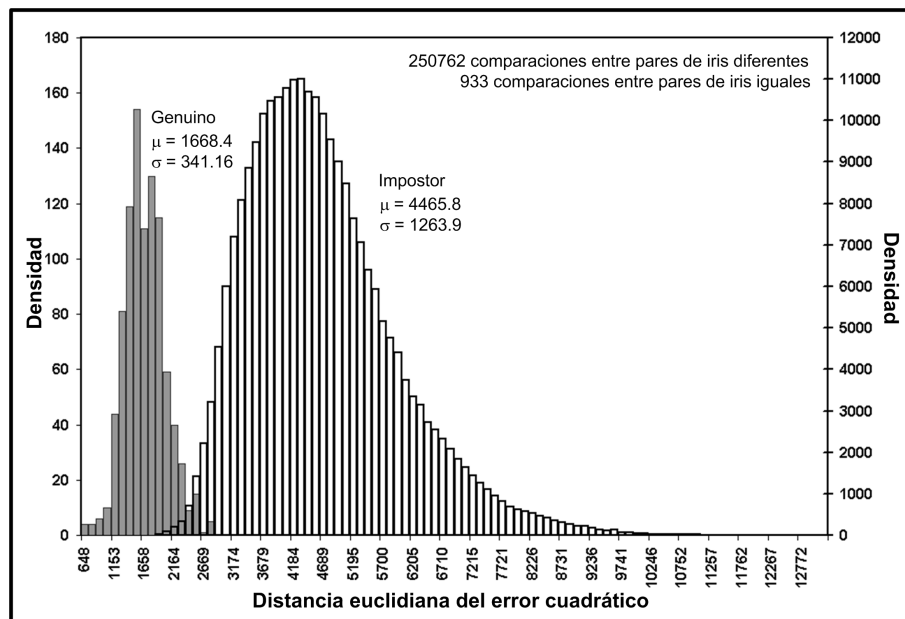


FIGURA D.25: Distribución genuino-impostor por el método de ARX para una base de datos de tamaño 70 %.

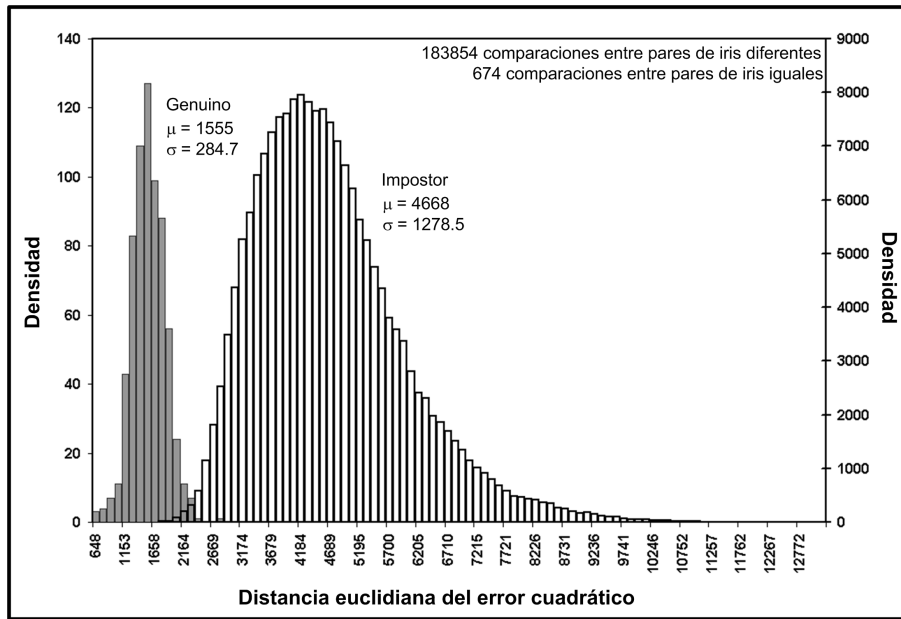


FIGURA D.26: Distribución genuino-impostor por el método de ARX para una base de datos de tamaño 60%.

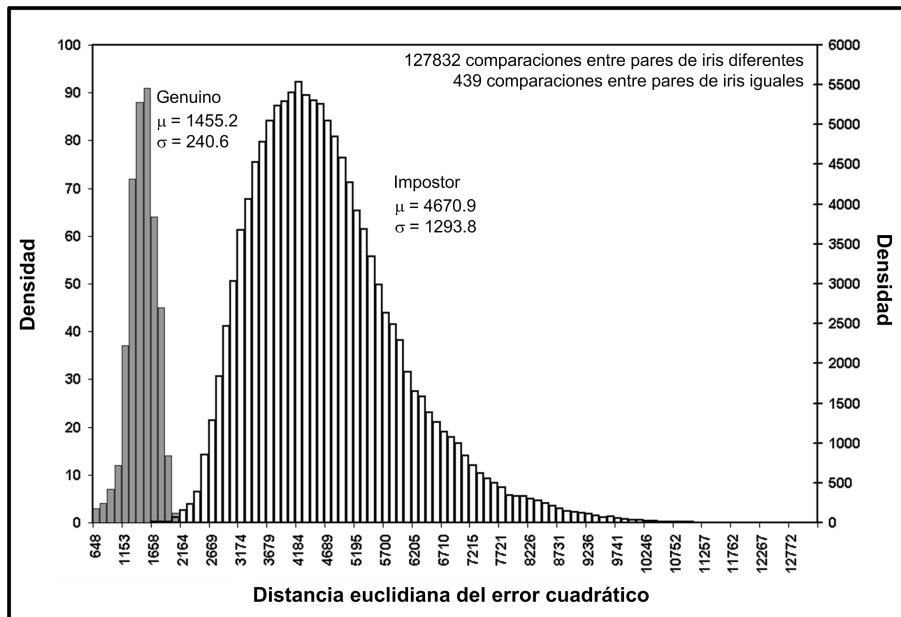


FIGURA D.27: Distribución genuino-impostor por el método de ARX para una base de datos de tamaño 50%.

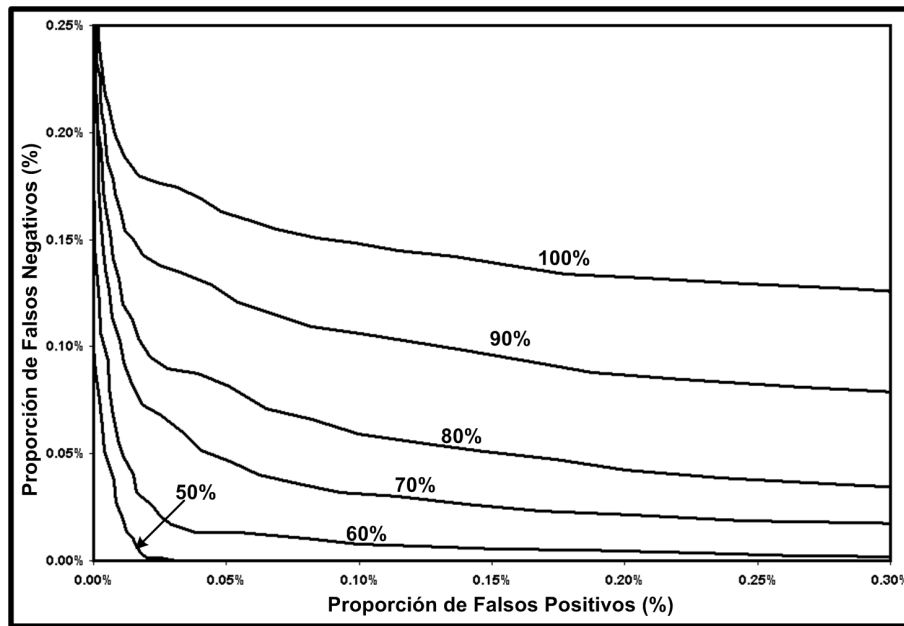


FIGURA D.28: Curva ROC por el método de ARX.





# Código fuente de los programas implementados en MATLAB®

---

## E.1. Localización de los límites límbico y pupilar del iris y normalización

```
function localnorm(irismax,sesion,sample)
%localnorm: Localiza los limites limbico y pupilar del iris de la imagen
%del iris. Una vez segmentado el iris, se obtiene la normalizacion de cada
%muestra.
%
%localnorm(irismax,sesion,sample)
%
%
%Argumentos:
%  irismax: numero maximo de iris de los que se desea pasar por la etapa
%          de localizacion y normalizacion.
%  sesion: la base de datos fue tomada en dos sesiones, solo se indica la
%          sesion de la base de datos con que se desea trabajar
%  sample: numero maximo de muestras de cada iris
%
%Recursos necesarios:
%  Base de datos: UBIRIS
%  Carpetas:
%  1. Iris (bin): Contiene las imagenes binarias de los iris de acuerdo al
%                formato de la base de datos. (Tipo de archivos: *.bin)
%  2. Irish (bin): Se generan las imagenes de los iris despues de pasar
%                  por la fase de ecualizacion por histogramas. (Tipo de archivos
%                  *.bin)
%  3. Irish (jpg): Se generan las imagenes de los iris despues de pasar
%                  por la fase de ecualizacion por histogramas. (Tipo de archivos
%                  *.jpg)
%  4. Cintas (bin): Se generan las imagenes normalizadas de los iris.
%                  (Tipo de archivos: *.bin)
%  5. Cintas (jpg): Se generan las imagenes normalizadas de los iris.
%                  (Tipo de archivos: *.jpg)
%  6. Cintash (bin): Se generan las imagenes normalizadas de los iris
%                    despues de pasar por la fase de ecualizacion por histogramas.
%                    (Tipo de archivos: *.bin)
%  6. Cintash (jpg): Se generan las imagenes normalizadas de los iris
%                    despues de pasar por la fase de ecualizacion por histogramas.
%                    (Tipo de archivos: *.jpg)
%
```

```

% Autor:
% Saul Montes de Oca Armeaga
% saul_montesdeoca@yahoo.com
% Mestria con Especialidad en Automatizacion
% Instituto Tecnologico y de Estudios Superiores de Monterrey
% Diciembre de 2005

% Todas las imagenes de la carpeta de "Iris (bin)" son pasadas por los
% procesos de ecualizacion de la imagen, localizacion de los limites
% limbico y pupilar, normalizacion y ecualizacion de la imagen normalizada.
% En las fases necesarias se obtienes los archivos *.jpg para poder
% apreciar los resultados de cada fase.
%Tamaño de las imagenes del iris:
col=800;
ren=600;

%Inician los ciclos para generar los archivos
for k=1:sesion
    for i=1:irismax
        for j=1:sample
            %Nombres de los archivos donde seran guardados los archivos de
            %cada fase.
            file_image = strcat('Iris (bin)\Img_',int2str(i),'_',int2str(k),'_',...
                int2str(j),'.bin');
            file_histo = strcat('Irish (bin)\Imgh_',int2str(i),'_',int2str(k),'_',...
                int2str(j),'.bin');
            cinta_image = strcat('Cintas (jpg)\cinta_',int2str(i),'_',int2str(k),...
                '_',int2str(j),'.jpg');
            cintah_image = strcat('Cintash (jpg)\cintah_',int2str(i),'_',...
                int2str(k),'_',int2str(j),'.jpg');
            cinta_file = strcat('Cintas (bin)\cinta_',int2str(i),'_',int2str(k),...
                '_',int2str(j),'.bin');
            cinta_histo = strcat('Cintash (bin)\cintah_',int2str(i),'_',...
                int2str(k),'_',int2str(j),'.bin');
            histo_image = strcat('Irish (jpg)\Imgh_',int2str(i),'_',...
                int2str(k),'_',int2str(j),'.jpg');
            %Abre el archivo de la imagen binaria del iris y verifica que
            %la haya leído
            fid=fopen(file_image,'r');
            if (fid>0)
                %Lee la imagen binaria y la pasa a la variable Ima
                [Ima,count]=fread(fid,[col,ren],'uchar');
                fclose(fid);
                Ima=Ima';
                %Ecualiza por histogramas la imagen del iris
                Ima=modhist(Ima);
                fidh = fopen(file_histo,'wb');
                fwrite(fidh,Ima','uchar');
                fclose(fidh);
                imagesc(Ima);colormap(gray);
                saveas(gcf,histo_image,'jpg');
                %Fase de localizacion del limite limbico
                [xl,yl,rl]=intdif(Ima,col,ren,80,200,5,1,0,0,0);
                %Fase de localizacion del limite pupilar
                [xp,yp,rp]=intdifp(Ima,col,ren,28,75,1,1,0,0.05,xl,yl,rl);
                %Fase de normalizacion de la imagen
                cinta=cartes(Ima,xl,yl,rl,xp,yp,rp,720,200);
                %Dibuja las coordenadas obtenidas de las fases de
                %localizacion
                circ=drawlim(Ima,yl,xl,rl,yp,xp,rp);
            end
        end
    end
end

```

```

        imagesc(circ);colormap(gray);
        saveas(gcf,histo_image,'jpg');
        fidh = fopen(cinta_file,'wb');
        fwrite(fidh,cinta','uchar');
        fclose(fidh);
        imagesc(cinta);colormap(gray);
        saveas(gcf,cinta_image,'jpg');
        %Ecuáliza por histogramas la imagen normalizada
        imm=modhist(cinta);
        fidh = fopen(cinta_histo,'wb');
        fwrite(fidh,imm','uchar');
        fclose(fidh);
        imagesc(imm);colormap(gray);
        saveas(gcf,cintah_image,'jpg');
    end
end
end
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DECLARACION DE FUNCIONES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Declaracion de la funcion para localizar el limite limbico del iris
function [xl,yl,rl]=intdif(im,col,ren,Rmin, Rmax,deltar,deltatheta,decimar,...
    x00,y00,r00)
%[xl,yl,rl]=intdif(im,col,ren,Rmin, Rmax,deltar,deltatheta,decimar,fac,x00,
% y00,r00)
%
%
%Argumentos:
% im: Imagen que contiene al iris.
% col: Numero de columnas de la imagen
% ren: Numero de renglones de la imagen
% Rmin: Radio minimo de la iteracion del operador integrodiferencial
% Rmax: Radio maximo de la iteracion del operador integrodiferencial
% deltar: Incremento del radio en el barrido de la imagen
% deltatheta: Incremento del angulo en el barrido de la imagen
% decimar: Como es un proceso iterativo, este valor se iguala a cero para
% asignar los valores sobre los cuales se hara el barrido de la imagen
% x00: Las coordenadas iniciales del centro del radio del limite limbico
% en el eje X
% y00: Las coordenadas iniciales del centro del radio del limite limbico
% en el eje Y
% r00: Valor inicial del radio
%
%Salida:
% xl: Coordenada sobre el eje X del centro del radio del limite limbico
% yl: Coordenada sobre el eje Y del centro del radio del limite limbico
% rl: Valor del radio del limite limbico
if decimar==0
    xmin=Rmax;
    xmax=col-Rmax;
    ymin=Rmax;
    ymax=ren-Rmax;
end
mmax=360/deltatheta;
idx=0;
maxval=0;
for xx=xmin+deltar:deltar:xmax-deltar

```

```

for yy=ymin+deltar:deltar:ymax-deltar
    x0=xx;
    y0=yy;
    for n=Rmin:deltar:Rmax
        idx=idx+1;
        sum1=0;
        sum2=0;
        for m=0:deltatheta:360
            if (m>=0&&m<=45) || (m>=135&&m<=225) || (m>=315&&m<=360)
                x1=floor((n-deltar/2)*cos(m*pi/180)+x0);
                y1=floor((n-deltar/2)*sin(m*pi/180)+y0);
                x2=floor((n+deltar/2)*cos(m*pi/180)+x0);
                y2=floor((n+deltar/2)*sin(m*pi/180)+y0);
                sum1=sum1+im(y1,x1);
                sum2=sum2+im(y2,x2);
            end
        end
        grad=sum2-sum1;
        if grad>maxval
            maxval=grad;
            x1=xx;
            y1=yy;
            r1=n;
        end
    end
end
end
end

%Declaracion de la funcion para localizar el limite pupilar del iris
function [xp,yp,rp]=intdifp(im,col,ren,Rmin, Rmax,deltar,deltatheta,decimar,...
    fac,x00,y00,r00)
[xp,yp,rp]=intdifp(im,col,ren,Rmin, Rmax,deltar,deltatheta,decimar,fac,
%   x00,y00,r00)
%
%Argumentos:
%   im: Imagen que contiene al iris.
%   col: Numero de columnas de la imagen
%   ren: Numero de renglones de la imagen
%   Rmin: Radio minimo de la iteracion del operador integrodiferencial
%   Rmax: Radio maxino de la iteracion del operador integrodiferencial
%   deltar: Incremento del radio en el barrido de la imagen
%   deltatheta: Incremento del angulo en el barrido de la imagen
%   decimar: Como es un proceso iterativo, este valor se iguala a cero para
%   asignar los valores sobre los cuales se hara el barrido de la imagen
%   fac: Factor para crear los limites de busqueda apartir del centro de la
%   localizacion del limite limbico
%   x00: Las coordenadas iniciales del centro del radio del limite pupilar
%   en el eje X
%   y00: Las coordenadas iniciales del centro del radio del limite pupilar
%   en el eje Y
%   r00: Valor inicial del radio
%
%Salida:
%   xp: Coordenada sobre el eje X del centro del radio del limite pupilar
%   yp: Coordenada sobre el eje Y del centro del radio del limite pupilar
%   rp: Valor del radio del limite pupilar
if decimar==0
    xmin=x00-fac*r00;
    xmax=x00+fac*r00;
    ymin=y00-fac*r00;

```

```

        ymax=y00+fac*r00;
    end
    mmax=360/deltatheta;
    idx=0;
    maxval=+inf;
    for xx=xmin+deltar:deltar:xmax-deltar
        for yy=ymin+deltar:deltar:ymax-deltar
            x0=floor(xx);
            y0=floor(yy);
            for n=Rmin:deltar:Rmax
                idx=idx+1;
                sum1=0;
                sum2=0;
                for m=0:deltatheta:360
                    x1=floor((n-deltar/2)*cos(m*pi/180)+x0);
                    y1=floor((n-deltar/2)*sin(m*pi/180)+y0);
                    x2=floor((n+deltar/2)*cos(m*pi/180)+x0);
                    y2=floor((n+deltar/2)*sin(m*pi/180)+y0);
                    sum1=sum1+im(y1,x1);
                    sum2=sum2+im(y2,x2);
                end
                grad=sum1+sum2;
                if grad<maxval
                    maxval=grad;
                    xp=floor(xx);
                    yp=floor(yy);
                    rp=n;
                end
            end
        end
    end
end

%Declaracion de la funcion para normalizar la imagen del iris
function cinta=cartes(im,xl,yl,rl,xp,yp,rp,m,n)
%cinta=cartes(im,xl,yl,rl,xp,yp,rp,m,n)
%
%Argumentos:
% im: Imagen que contiene al iris
% xl: Coordenada sobre el eje X del centro del radio del limite limbico
% yl: Coordenada sobre el eje Y del centro del radio del limite limbico
% rl: Valor del radio del limite limbico
% xp: Coordenada sobre el eje X del centro del radio del limite pupilar
% yp: Coordenada sobre el eje Y del centro del radio del limite pupilar
% rp: Valor del radio del limite pupilar
% m: Numero de columnas de la imagen de salida
% n: Numero de renglones de la imagen de salida
%
%Salida:
% cinta: Imagen normalizada del iris
cinta=zeros(n,m);
for x=1:m
    theta=(x*2*pi)/(m);
    Xlp=xp+rp*cos(theta);
    Ylp=yp+rp*sin(theta);
    Xll=xl+rl*cos(theta);
    Yll=yl+rl*sin(theta);
    for y=1:n
        r=rp+y*(rl-rp)/(n);
        rcero=(r-rp)/(rl-rp);
        Xrt=floor(Xll-rcero*(Xll-Xlp));
    end
end

```

```

        Yrt=floor(Yll-rcero*(Yll-Ylp));
        if y==1||y==n
            [Yrt,Xrt];
        end
        cinta(y,x)=im(Yrt,Xrt);
    end
end

%Declaracion de la funcion para producir la imagen con los limites limbico
%y pupilar
function I = drawlim(II, al, bl, rl, ap, bp, rp)
%I = drawlim(II, al, bl, rl, ap, bp, rp)
%
%Argumentos:
% II: Imagen que contiene al iris
% al: Coordenada del eje X del centro del limite limbico
% bl: Coordenada del eje Y del centro del limite limbico
% rl: Valor del radio del limite limbico
% ap: Coordenada del eje X del centro del limite pupilar
% bp: Coordenada del eje Y del centro del limite pupilar
% rl: Valor del radio del limite pupilar
%
%Salida:
% II: Imagen que contiene al iris con la localizacion de los limites
% limbico y pupilar
[m n] = size(II);
I = II ;
    for i=0.01:0.01:360
        xl = round(al + rl * cos(i*(2*pi/360)));
        yl = round(bl + rl * sin(i*(2*pi/360)));
        xp = round(ap + rp * cos(i*(2*pi/360)));
        yp = round(bp + rp * sin(i*(2*pi/360)));
        if (xl>=1) & (xl<=m)
            if (yl>=1) & (yl<=n)
                I(xl,yl) = 1;
            end
        end
        if (xp>=1) & (xp<=m)
            if (yp>=1) & (yp<=n)
                I(xp,yp) = 255;
            end
        end
    end
end

%Declaracion de la funcion para llevar a cabo la ecualizacion por
%histogramas de una imagen
function imm=modhist(im)
%imm=modhist(im)
%
%Argumentos:
% im: Imagen original
%
%Salida:
% im: Imagen ecualizada
valmin=min(min(im));
[fil,col]=size(im);
imm=[];
histograma=zeros(256,1);
acum=zeros(256,1);
for i=1:fil

```

```

    for j=1:col
        histograma(ceil(im(i,j))+1,1)=histograma(ceil(im(i,j))+1,1)+1;
    end
end
acum(1,1)=histograma(1,1);
for i=2:256
    acum(i,1)=histograma(i,1)+acum(i-1,1);
end
for i=1:fil
    for j=1:col
        imm(i,j)=ceil((256/(fil*col)*acum(im(i,j)+1,1)-1));
    end
end
end

```

## E.2. Extracción de características, comparación y decisión

### E.2.1. Método ARX

```

function [umbrales,pruebas,nocinta,tiempo]=arx(maxi,grid,factor,na,nb)
%arx: Realiza la extraccion de caracteristicas, comparacion y decision
%de la base de datos
%
% [umbrales,pruebas,nocinta,tiempo]=arx(maxi,grid,factor,na,nb)
%
%
%Argumentos:
% maxi: Numero maximo de iris de los que se desea pasar por la etapa
%       de extraccion de caracteristicas, comparacion y decision
% grid: Numero de pixeles promedio que se toma para el modelo ARX
% factor: Numero de base que se va a usar para la prueba, depende de la
%         base de datos y el nivel de ruido.
% na: Numero a's que representan al modelo ARX
% nb: Numero b's que representan al modelo ARX
%
%Salida:
% umbrales: Matriz con el resultado de los umbrales
% pruebas: Matriz con el resultado de las pruebas
% nocinta: Numero de iris usados en la base de datos ficticia
% tiempo: Tiempo total en que fueron llevadas a cabo las pruebas de todas
%         las muestras
%
%Recursos necesarios:
% Base de datos: UBIRIS
% Archivos:
% 1. seleccion100.bin: Base de datos de tamaño 100%
% 2. seleccion90.bin: Base de datos de tamaño 90%
% 3. seleccion80.bin: Base de datos de tamaño 80%
% 4. seleccion70.bin: Base de datos de tamaño 70%
% 5. seleccion60.bin: Base de datos de tamaño 60%
% 6. seleccion50.bin: Base de datos de tamaño 50%
% Carpetas:
% 1. Cintash (datos): Archivos que tienen los datos de cada imagen
%    con el grid necesario para ser analizado
% 2. Cintash (template): Sirve para guardar los datos del modelo de
%    los iris analizados
%
% Autor:
% Saul Montes de Oca Armeaga

```



```

% saul_montesdeoca@yahoo.com
% Mestria con Especialidad en Automatizacion
% Instituto Tecnologico y de Estudios Superiores de Monterrey
% Diciembre de 2005

% Numero maximo de muestras de iris y numero de iris diferentes
mm=6;nn=241;
%Seleccion de las plantillas
nocinta=templates(mm,nn,maxi,grid,na,nb,factor);

%Seleccion de los umbrales
umbrales=mcumbral(mm,nn,maxi,grid,na,nb,factor);

%Pruebas finales
[pruebas,tiempo]=mcprueba(mm,nn,maxi,grid,na,nb,factor);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DECLARACION DE FUNCIONES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Declaracion de la funcion para crear las plantillas de todas las pruebas
function nocinta=templates(mm,nn,max,grid,na,nb,factor)
% nocinta=templates(mm,nn,max,grid,na,nb,factor)
%
%
%Argumentos:
% mm: Numero maximo de muestras del iris
% nn: Numero de iris diferentes
% max: Numero maximo de iris de los que se desea pasar por la etapa
%     de extraccion de caracteristicas, comparacion y decision
% grid: Numero de pixeles promedio que se toma para el modelo ARX
% na: Numero a's que representan al modelo ARX
% nb: Numero b's que representan al modelo ARX
% factor: Numero de base que se va a usar para la prueba, depende de la
%        base de datos y el nivel de ruido.
%
%Salida:
% nocinta: Numero de iris usados en la base de datos ficticia
sesion=1;iris=max;sample=6;m=0;gamma=1;
%Tamaño de la imagen normalizada
col=720;ren=200;
%Tiempo muerto
nk=0;
switch factor,
    case 0,
        sim_file = 'seleccion0.bin';
    case 100,
        sim_file = 'seleccion100.bin';
    case 90,
        sim_file = 'seleccion90.bin';
    case 80,
        sim_file = 'seleccion80.bin';
    case 70,
        sim_file = 'seleccion70.bin';
    case 60,
        sim_file = 'seleccion60.bin';
    case 50,
        sim_file = 'seleccion50.bin';
end

```

```

sim=readbin(sim_file,nn,mm,'char');
maxiris=sum(sum(sim(1,1:max)>0));
mascara=zeros(mm,nn);
while sum(mascara(1,:))<ceil(0.75*sum(sum(sim(1,1:max)>0))),
    randumb=ceil(rand*(max));
    if mascara(1,randumb)==0 && sim(1,randumb)~=0
        mascara(1,randumb)=1;
    end
end
for k=1:sesion
    for i=1:iris
        if mascara(1,i)~=0 && sim(1,i)~=0
            for j=1:sample
                if sim(1,i)==j
                    m=m+1;
                    datos_file = strcat('Cintash (datos)\datos',int2str(grid),'_',...
                        int2str(i),'_',int2str(k),'_',int2str(j),'.bin');
                    theta_file = strcat('Cintash (template)\theta_',int2str(m),'.bin');
                    yn_file = strcat('Cintash (template)\yn_',int2str(m),'.bin');
                    data=readbin(datos_file,2,144000/(grid*grid),'char');
                    [psi,yn,theta,Jn]=pmc(data,na,nb,nk,gamma);
                    nocinta(m,:)=[i,k,j];
                    fidt = fopen(theta_file,'wb');
                    fwrite(fidt,theta','float');
                    fclose(fidt);
                    fidy = fopen(yn_file,'wb');
                    fwrite(fidy,yn','float');
                    fclose(fidy);
                end
            end
        end
    end
end
end
end
fidn = fopen('nocinta.bin','wb');
fwrite(fidn,nocinta','char');
fclose(fidn);
fid = fopen('mascara.bin','wb');
fwrite(fid,mascara','char');
fclose(fid);

```

```

%Declaracion de la funcion para crear los umbrales de todas las pruebas
function umbrales=mcumbral(mm,nn,maximo,grid,na,nb,factor)
% umbrales=mcumbral(mm,nn,maximo,grid,na,nb,factor)
%
%
%Argumentos:
% mm: Numero maximo de muestras del iris
% nn: Numero de iris diferentes
% max: Numero maximo de iris de los que se desea pasar por la etapa
% de extraccion de caracteristicas, comparacion y decision
% grid: Numero de pixeles promedio que se toma para el modelo ARX
% na: Numero a's que representan al modelo ARX
% nb: Numero b's que representan al modelo ARX
% factor: Numero de base que se va a usar para la prueba, depende de la
% base de datos y el nivel de ruido.
%
%Salida:
% umbrales: Matriz con el resultado de los umbrales
gamma=1;mn=max(na,nb);
%Tamaño de la imagen normalizada

```

```

col=720;ren=200;
%Tiempo muerto
nk=0;
switch factor,
    case 0,
        sim_file = 'seleccion0.bin';
    case 100,
        sim_file = 'seleccion100.bin';
    case 90,
        sim_file = 'seleccion90.bin';
    case 80,
        sim_file = 'seleccion80.bin';
    case 70,
        sim_file = 'seleccion70.bin';
    case 60,
        sim_file = 'seleccion60.bin';
    case 50,
        sim_file = 'seleccion50.bin';
end
sim=readbin(sim_file,nn,mm,'char');
maxiris=sum(sum(sim(1,1:maximo)>0));
mascara=readbin('mascara.bin',nn,mm,'char');
sesion=1;iris=maximo;sample=6;mu=0;
m=ceil(0.05*sum(sum(sim(:,1:maxiris)>0))/2);
nocinta=readbin('nocinta.bin',3,ceil(0.75*sum(sum(sim(1,1:maxiris)>0))),'char');
while sum(mascara(2,:))<m,
    randumb=ceil(rand*(maximo));
    if mascara(2,randumb)==0 && mascara(1,randumb)==1...
        && randumb~=0 && sim(2,randumb)~=0
        k=sesion;
        i=randumb;
        j=sim(2,randumb);
        datos_file = strcat('Cintash (datos)\datos',int2str(grid),'_',...
            int2str(i),'_',int2str(k),'_',int2str(j),'.bin');
        data=readbin(datos_file,2,144000/(grid*grid),'char');
        mu=mu+1;
        jnmin(mu,1)=+inf;
        baseorig_u(mu,1)=0;
        for l=1:ceil(0.75*sum(sum(sim(1,1:maxiris)>0)))
            if nocinta(l,1)==i
                baseorig_u(mu,1)=l;
            end
            theta_file = strcat('Cintash (template)\theta_',int2str(l),'.bin');
            yn_file = strcat('Cintash (template)\yn_',int2str(l),'.bin');
            theta=readbin(theta_file,1,na+nb,'float');
            yn=readbin(yn_file,1,144000/(grid*grid)-mn,'float');
            [psi,yn,jn]=errormc(theta,yn,data,na,nb,gamma);
            if(jnmin(mu,1)>jn)
                jnmin(mu,1)=jn;
                noiris_u(mu,:)=[i,j];
                nobase_u(mu,1)=l;
            end
        end
        mascara(2,randumb)=1;
    end
end
while sum(mascara(2,:))<2*m,
    randumb=ceil(rand*(maximo));
    if mascara(2,randumb)==0 && mascara(1,randumb)==0...
        && randumb~=0 && sim(2,randumb)~=0
        k=sesion;

```

```

        i=randumb;
        j=sim(2,randumb);
        datos_file = strcat('Cintash (datos)\datos',int2str(grid),...
            '_' ,int2str(i),'_' ,int2str(k),'_' ,int2str(j),'.bin');
        data=readbin(datos_file,2,144000/(grid*grid),'char');
        mu=mu+1;
        jnmin(mu,1)=+inf;
        baseorig_u(mu,1)=0;
        for l=1:ceil(0.75*sum(sum(sim(1,1:maxiris)>0)))
            if nocinta(1,1)==i
                baseorig_u(mu,1)=1;
            end
            theta_file = strcat('Cintash (template)\theta_',int2str(l),'.bin');
            yn_file = strcat('Cintash (template)\yn_',int2str(l),'.bin');
            theta=readbin(theta_file,1,na+nb,'float');
            yn=readbin(yn_file,1,144000/(grid*grid)-mn,'float');
            [psi,yn,jn]=errormc(theta,yn,data,na,nb,gamma);
            if (jnmin(mu,1)>jn)
                jnmin(mu,1)=jn;
                noiris_u(mu,:)=[i,j];
                nobase_u(mu,1)=1;
            end
        end
        mascara(2,randumb)=1;
    end
end
umbrales=[baseorig_u,noiris_u,nobase_u,jnmin_u];
fid = fopen('mascara.bin','wb');
fwrite(fid,mascara','char');
fclose(fid);

%Declaracion de la funcion que simula las pruebas
function [pruebas,tiempo]=mcprueba(mm,nn,maximo,grid,na,nb,factor)
% [pruebas,tiempo]=mcprueba(mm,nn,maximo,grid,na,nb,factor)
%
%
%Argumentos:
% mm: Numero maximo de muestras del iris
% nn: Numero de iris diferentes
% max: Numero maximo de iris de los que se desea pasar por la etapa
% de extraccion de caracteristicas, comparacion y decision
% grid: Numero de pixeles promedio que se toma para el modelo ARX
% na: Numero a's que representan al modelo ARX
% nb: Numero b's que representan al modelo ARX
% factor: Numero de base que se va a usar para la prueba, depende de la
% base de datos y el nivel de ruido.
%
%Salida:
% pruebas: Matriz con el resultado de las pruebas
% tiempo: Tiempo total en que fueron llevadas a cabo las pruebas de todas
% las muestras
gamma=1;mn=max(na,nb);
%Tamaño de la imagen normalizada
col=720;ren=200;
%Tiempo muerto
nk=0;
switch factor,
    case 0,
        sim_file = 'seleccion0.bin';
    case 100,
        sim_file = 'seleccion100.bin';
end

```

```

case 90,
    sim_file = 'seleccion90.bin';
case 80,
    sim_file = 'seleccion80.bin';
case 70,
    sim_file = 'seleccion70.bin';
case 60,
    sim_file = 'seleccion60.bin';
case 50,
    sim_file = 'seleccion50.bin';
end
sim=readbin(sim_file,nn,mm,'char');
maxiris=sum(sum(sim(1,1:maximo)>0));
mascara=readbin('mascara.bin',nn,mm,'char');
nocinta=readbin('nocinta.bin',3,ceil(.75*maximo),'char');
sesion=1;iris=maximo;sample=6;
m=ceil(0.75*sum(sum(sim(1,1:maximo)>0)));
[mm,nn]=size(sim);mu=0;ver=1;k=sesion;
tic;
for q=1:mm
    for i=1:maximo
        if sim(q,i)~=0 && mascara(q,i)==0
            j=sim(q,i);
            datos_file = strcat('Cintash (datos)\datos',int2str(grid),...
                '_ ',int2str(i),'_ ',int2str(k),'_ ',int2str(j),'.bin');
            data=readbin(datos_file,2,144000/(grid*grid),'char');
            mu=mu+1;
            jnmin_p(mu,1)=+inf;
            baseorig_p(mu,1)=0;
            for l=1:m
                if nocinta(l,1)==i
                    baseorig_p(mu,1)=1;
                end
                theta_file = strcat('Cintash (template)\theta_',int2str(l),'.bin');
                yn_file = strcat('Cintash (template)\yn_',int2str(l),'.bin');
                theta=readbin(theta_file,1,na+nb,'float');
                yn=readbin(yn_file,1,144000/(grid*grid)-mn,'float');
                [psi,yn,jn]=errormc(theta,yn,data,na,nb,gamma);
                if(jnmin_p(mu,1)>jn)
                    jnmin_p(mu,1)=jn;
                    noiris_p(mu,:)=[i,j];
                    nobase_p(mu,1)=1;
                end
            end
        end
    end
end
tiempo=toc;
pruebas=[baseorig_p,noiris_p,nobase_p,jnmin_p];

%Declaracion de la funcion para obtener el modelo ARX
function [psi,yn,theta,Jn]=pmc(data,na,nb,nk,gamma)
% [psi,yn,theta,Jn]=pmc(data,na,nb,nk,gamma)
%
%
%Argumentos:
% data: Datos que tienen las entradas y salidas del modelo ARX
% na: Numero a's que representan al modelo ARX
% nb: Numero b's que representan al modelo ARX
% nk: Tiempo muerto del modelo ARX
% factor: Numero de base que se va a usar para la prueba, depende de la

```

```

%     base de datos y el nivel de ruido.
%     gamma: Factor de olvido
%
%Salida:
%     Para definir el predictor del modelo se emplea la siguiente ecuacion:
%     error=yn-psi*theta
%     Jn=error'*error
data(:,1)=data(:,1)-mean(data(:,1));
data(:,2)=data(:,2)-mean(data(:,2));
mn=max(na,nb+nk);
[m,n]=size(data);
yn=data(mn+1:m,2);
psi=data(mn:m-1,2);
for i=1:na-1
    psi=[psi,data(mn-i:m-1-i,2)];
end
for i=0:nb-1
    psi=[psi,data(mn-i-nk:m-1-i-nk,1)];
end
[mpsi,npsi]=size(psi);
psit=psi;
for i=1:mpsi
    psit(i,:)=psi(i,:)*gamma^(mpsi-i);
end
theta=(psit'*psi)^(-1)*psit'*yn;
error=yn-psi*theta;
errorr=error;
for i=1:mpsi
    errorr(i,:)=error(i,:)*gamma^(mpsi-i);
end
Jn=sum(errorr'*error)/m;

%Declaracion de la funcion para obtener el error de prediccion del modelo
function [psi,yn,Jn]=errormc(theta_,yn,data,na,nb,gamma)
% [psi,yn,Jn]=errormc(theta_,yn,data,na,nb,gamma)
%
%
%Argumentos:
%     theta_: Parametros del modelo del iris plantilla
%     yn: Salida del modelo del iris plantilla
%     data: Datos que tienen las entradas y salidas del modelo ARX
%     na: Numero a's que representan al modelo ARX
%     nb: Numero b's que representan al modelo ARX
%     nk: Tiempo muerto del modelo ARX
%     gamma: Factor de olvido
%
%Salida:
%     Para definir el predictor del modelo se emplea la siguiente ecuacion:
%     error=yn-psi*theta
%     Jn=error'*error
data(:,1)=data(:,1)-mean(data(:,1));
data(:,2)=data(:,2)-mean(data(:,2));
mn=max(na,nb);
[m,n]=size(data);
psi=data(mn:m-1,2);
for i=1:na-1
    psi=[psi,data(mn-i:m-1-i,2)];
end
for i=0:nb-1
    psi=[psi,data(mn-i:m-1-i,1)];
end

```

```

end
[mpsi,npsi]=size(psi);
error=yn-psi*theta_;
errort=error;
for i=1:mpsi
    errort(i,:)=error(i,:)*gamma^(mpsi-i);
end
Jn=sum(errort'*error)/m;

```

## E.2.2. Método de muestreo

```

function [umbrales,pruebas,tiempo]=sampling(maxiris,noparche,sizeparchealto,...
    sizeparchelargo,tamanobase,tamanoumbbral,base,repertir,method)
%sampling: Realiza la extraccion de caracteristicas, comparacion y decision
%de la base de datos
%
% [umbrales,pruebas,tiempo]=sampling(maxiris,noparche,sizeparchealto,
%   sizeparchelargo,tamanobase,tamanoumbbral,base,repertir,method)
%
%
%Argumentos:
% maxiris: Numero maximo de iris de los que se desea pasar por la etapa
%   de extraccion de caracteristicas, comparacion y decision
% noparche: Numero de subimagenes que se desea muestrear de cada iris
%   para caracterizar al iris
% sizeparchealto: Altura de cada subimagen en pixeles
% sizeparchelargo: Ancho de cada subimagen en pixeles
% tamaño de la base: Porcentaje del numero de iris diferentes de la base
%   de datos que se van a designar para crear la base de datos ficticia, el
%   numero es de 0 a 1
% tamanoumbbral: Porcentaje del numero de total de las muestras de los
%   iris de la base de datos que se van a designar el numero de muestras
%   usadas para crear los umbrales de decision.
% base: Numero de base que se va a usar para la prueba, depende de la
%   base de datos y el nivel de ruido.
% repertir: Numero con el que se va a guardar el archivo con los
%   resultados
% method: El tipo de muestreo que se va aplicar
%   'mac'         - Metodo de muestreo con reposicion
%   'mas'         - Metodo de muestreo sin reposicion
%   'hammer'     - Metodo de muestreo Hammersley
%
%Salida:
% umbrales: Matriz con el resultado de los umbrales
% pruebas: Matriz con el resultado de las pruebas
% tiempo: Tiempo total en que fueron llevadas a cabo las pruebas de todas
%   las muestras
%
%Recursos necesarios:
% Base de datos: UBIRIS
% Archivos:
%   1. seleccion100.bin: Base de datos de tamaño 100%
%   2. seleccion90.bin: Base de datos de tamaño 90%
%   3. seleccion80.bin: Base de datos de tamaño 80%
%   4. seleccion70.bin: Base de datos de tamaño 70%
%   5. seleccion60.bin: Base de datos de tamaño 60%
%   6. seleccion50.bin: Base de datos de tamaño 50%
% Carpetas:
%   1. Cintas (bin): Se generan las imagenes normalizadas de los iris
%   despues de pasar por la fase de ecualizacion por histogramas.

```

```

%           (Tipo de archivos: *.bin)
%           2. Resultados (sac): Carpeta donde se guardan los archivos de los
%              resultados del muestreo aleatorio con reposicion
%           3. Resultados (sas): Carpeta donde se guardan los archivos de los
%              resultados del muestreo aleatorio sin reposicion
%           4. Resultados (sh): Carpeta donde se guardan los archivos de los
%              resultados del muestreo Hammersley
%
% Autor:
% Saul Montes de Oca Armeaga
% saul_montesdeoca@yahoo.com
% Mestria con Especialidad en Automatizacion
% Instituto Tecnologico y de Estudios Superiores de Monterrey
% Diciembre de 2005

% Todas las imagenes normalizadas de los iris deben estar en la carpeta de
% Cintas(bin) aunque solo seran utilizadas las imagenes de la base de datos
% seleccionada. Los archivos son generados en la carpeta correspondiente al
% tipo de muestreo seleccionado.
% Tamaño de las imagenes normalizadas:
col=720;ren=200;
% Numero de iris diferentes y numero maximo de muestras por iris de la base
% de datos UBIRIS
iris=241;sample=6;

% Creacion de las coordenadas con muestreo seleccionado
switch method,
    case 'mac',
        coord=macon(col,ren,noparche,sizeparchealto,sizeparchelargo);
    case 'mas',
        coord=masin(col,ren,noparche,sizeparchealto,sizeparchelargo);
    case 'hammer',
        coord=hammersley(col,ren,noparche,sizeparchealto,sizeparchelargo,2);
end
save datossa coord

% Creacion de las plantillas para todas las pruebas
template(coord,col,ren,sample,iris,maxiris,noparche,sizeparchealto,...
    sizeparchelargo,tamanobase,base);

% Creacion de los umbrales
umbrales=umbral(coord,col,ren,sample,iris,maxiris,noparche,sizeparchealto,...
    sizeparchelargo,tamanobase,tamanoumbrales);

% Simulacion de las pruebas y generacion de los archivos con los resultados
load datossa mascara
[pruebas,tiempo]=prueba(mascara,coord,col,ren,sample,iris,maxiris,...
    noparche,sizeparchealto,sizeparchelargo,tamanobase);
load datossa
switch method,
    case 'mac',
        result_file=strcat('Resultados (sac)\resultado_',num2str(noparche),...
            '_',num2str(sizeparchealto),'_',num2str(sizeparchelargo),...
            '_',num2str(tamanobase),'_',num2str(tamanoumbrales),'_',num2str(base),...
            '_',num2str(repetir),'.mat');
    case 'mas',
        result_file=strcat('Resultados (sas)\resultado_',num2str(noparche),...
            '_',num2str(sizeparchealto),'_',num2str(sizeparchelargo),...
            '_',num2str(tamanobase),'_',num2str(tamanoumbrales),'_',num2str(base),...
            '_',num2str(repetir),'.mat');
    case 'hammer',

```



```

        result_file=strcat('Resultados (sh)\resultado_',num2str(noparche),...
            '_ ',num2str(sizeparchealto),'_ ',num2str(sizeparchelargo),...
            '_ ',num2str(tamanobase),'_ ',num2str(tamanoumbrales),'_ ',num2str(base),...
            '_ ',num2str(repetir),'.mat');
end
save(result_file,'nobase','pruebas','tiempo','umbrales');
delete('datossa.mat');delete('datosumb.mat')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DECLARACION DE FUNCIONES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Declaracion de la funcion para crear las plantillas de todas las pruebas
function template(coord,col,ren,sample,iris,maxiris,noparches,sizeparchealto,...
    sizeparchelargo,tamanobase,base)
%template(coord,col,ren,sample,iris,maxiris,noparches,sizeparchealto,
% sizeparchelargo,tamanobase,base)
%
%
%Argumentos:
% coord: Coordenadas de las subimagenes muestreadas
% col: Numero de columnas de la imagen normalizada
% ren: Numero de renglones de la imagen normalizada
% sample: Numero maximo de muestras que tiene cada iris
% iris: Numero de iris diferentes que tiene la base de datos
% maxiris: Numero maximo de iris de los que se desea pasar por la etapa
% de extraccion de caracteristicas, comparacion y decision
% noparche: Numero de subimagenes que se desea muestrear de cada iris
% para caracterizar al iris
% sizeparchealto: Altura de cada subimagen en pixeles
% sizeparchelargo: Ancho de cada subimagen en pixeles
% tamaño de la base: Porcentaje del numero de iris diferentes de la base
% de datos que se van a designar para crear la base de datos ficticia, el
% numero es de 0 a 1
% base: Numero de base que se va a usar para la prueba, depende de la
% base de datos y el nivel de ruido.
sesion=1;m=0;
%Base de datos que va a ocupar, es necesario tener los archivos de
%seleccion para poder llevar a cabo este paso
switch base,
    case 100,
        sim_file = 'seleccion100.bin';
    case 90,
        sim_file = 'seleccion90.bin';
    case 80,
        sim_file = 'seleccion80.bin';
    case 70,
        sim_file = 'seleccion70.bin';
    case 60,
        sim_file = 'seleccion60.bin';
    case 50,
        sim_file = 'seleccion50.bin';
end
sim=readbin(sim_file,iris,sample);
%El objetivo de mascara es saber que iris han sido empleados en cada etapa,
%donde cero significa que no ha sido usado en ninguna etapa y uno que ya
%fue empleado.
[mm,nn]=size(sim);mascara=zeros(sample,iris);
while sum(mascara(1,:))<ceil(tamanobase*sum(sum(sim(1,1:maxiris)>0))),
    randumb=ceil(rand*(maxiris));

```

```

    if mascara(1,randumb)==0 && sim(1,randumb)~=0
        mascara(1,randumb)=1;
    end
end
for i=1:maxiris
    if mascara(1,i)~=0 && sim(1,i)~=0
        for j=1:sample
            if sim(1,i)==j
                cinta_file = strcat('Cintash (bin)\cintah_',int2str(i),...
                    '_ ',int2str(sesion),'_',int2str(j),'.bin');
                m=m+1;
                acumulado(:,m,:)=parches(cinta_file,noparches,...
                    sizeparchealto,sizeparchelargo,coord,col,ren);
                nobase(m,:)=[i,j];
            end
        end
    end
end
mascaraorig=mascara;
%Las variables las va a guardar en el archivo datossa.mat
save datossa nobase mascara mascaraorig sim -APPEND
save datosumb acumulado

%Declaracion de la funcion para crear los umbrales de todas las pruebas
function umbrales=umbral(coord,col,ren,sample,iris,maxiris,noparches,...
    sizeparchealto,sizeparchelargo,tamanobase,tamanoumbrales)
%umbral(coord,col,ren,sample,iris,maxiris,noparches,sizeparchealto,
% sizeparchelargo,tamanobase,tamanoumbrales)
%
%
%Argumentos:
% coord: Coordenadas de las subimagenes muestreadas
% col: Numero de columnas de la imagen normalizada
% ren: Numero de renglones de la imagen normalizada
% sample: Numero maximo de muestras que tiene cada iris
% iris: Numero de iris diferentes que tiene la base de datos
% maxiris: Numero maximo de iris de los que se desea pasar por la etapa
% de extraccion de caracteristicas, comparacion y decision
% noparche: Numero de subimagenes que se desea muestrear de cada iris
% para caracterizar al iris
% sizeparchealto: Altura de cada subimagen en pixeles
% sizeparchelargo: Ancho de cada subimagen en pixeles
% tamaño de la base: Porcentaje del numero de iris diferentes de la base
% de datos que se van a designar para crear la base de datos ficticia,
% el numero es de 0 a 1
% tamanoumbrales: Porcentaje del numero de total de las muestras de los
% iris de la base de datos que se van a designar el numero de muestras
% usadas para crear los umbrales de decision
%
%Salida:
% umbrales: Matriz con el resultado de los umbrales
load datossa nobase mascara sim
load datosumb
mu=0;sesion=1;m=ceil(tamanoumbrales*sum(sum(sim(:,1:maxiris)>0))/2.0);
while sum(mascara(2,:))+sum(mascara(3,:))<m
    randumb=ceil(rand*(maxiris));
    if ((mascara(2,randumb)==0&&sim(2,randumb)~=0)|| (mascara(3,...
        randumb)==0&&sim(3,randumb)~=0)) && mascara(1,...
        randumb)==1 && randumb~=0
        i=randumb;j=sim(2,randumb);mu=mu+1;
        if mascara(2,randumb)==1
            j=sim(3,randumb);
        end
    end
end

```

```

end
cinta_file = strcat('Cintash (bin)\cintah_',int2str(i),'_',...
    int2str(sesion),'_',int2str(j)'.bin');
acumulado_u(:,1,:)=parches(cinta_file,noparches,sizeparchealto,...
    sizeparchelargo,coord,col,ren);
disteuclmin_u(mu,1)=+inf;
baseorig_u(mu,1)=0;
for l=1:ceil(tamanobase*sum(sum(sim(1,1:maxiris)>0)))
    if nobase(1,1)==i
        baseorig_u(mu,1)=1;
    end
    disteucl=sqrt(sum(sum((acumulado(:,1,:)-acumulado_u(:,1,:)).^2)));
    if(disteuclmin_u(mu,1)>disteucl)
        disteuclmin_u(mu,1)=disteucl;
        noiris_u(mu,:)=[i,j];
        nobase_u(mu,1)=1;
    end
end
if mascara(2,randumb)==1
    mascara(3,randumb)=1;
else
    mascara(2,randumb)=1;
end
end
end
while sum(mascara(2,:))+sum(mascara(3,:))<2*m
    randumb=ceil(rand*(maxiris));
    if ((mascara(2,randumb)==0&&sim(2,randumb)~=0) || (mascara(3,...
        randumb)==0&&sim(3,randumb)~=0)) && mascara(1,randumb)...
        ==0 && randumb~=0
        k=sesion;i=randumb;j=sim(2,randumb);mu=mu+1;
        if mascara(2,randumb)==1
            j=sim(3,randumb);
        end
        cinta_file = strcat('Cintash (bin)\cintah_',int2str(i),...
            '__',int2str(sesion),'_',int2str(j)'.bin');
        acumulado_u(:,1,:)=parches(cinta_file,noparches,sizeparchealto,...
            sizeparchelargo,coord,col,ren);
        disteuclmin_u(mu,1)=+inf;
        baseorig_u(mu,1)=0;
        for l=1:ceil(tamanobase*sum(sum(sim(1,1:maxiris)>0)))
            if nobase(1,1)==i
                baseorig_u(mu,1)=1;
            end
            disteucl=sqrt(sum(sum((acumulado(:,1,:)-acumulado_u(:,1,:)).^2)));
            if(disteuclmin_u(mu,1)>disteucl)
                disteuclmin_u(mu,1)=disteucl;
                noiris_u(mu,:)=[i,j];
                nobase_u(mu,1)=1;
            end
        end
        if mascara(2,randumb)==1
            mascara(3,randumb)=1;
        else
            mascara(2,randumb)=1;
        end
    end
end
end
umbrales=[baseorig_u,noiris_u,nobase_u,disteuclmin_u];
save datossa mascara umbrales -APPEND

```

```

%Declaracion de la funcion que simula las pruebas
function [pruebas,tiempo]=prueba(mascara,coord,col,ren,sample,iris,maxiris,...
    noparches,sizeparchealto,sizeparchelargo,tamanobase)
%[pruebas,tiempo]=prueba(mascara,coord,col,ren,sample,iris,maxiris,
%    noparches,sizeparchealto,sizeparchelargo,tamanobase)
%
%
%Argumentos:
%    mascara: Archivo que ayuda a saber que muestras fueron usadas en las
%    etapas anteriores
%    coord: Coordenadas de las subimagenes muestreadas
%    col: Numero de columnas de la imagen normalizada
%    ren: Numero de renglones de la imagen normalizada
%    sample: Numero maximo de muestras que tiene cada iris
%    iris: Numero de iris diferentes que tiene la base de datos
%    maxiris: Numero maximo de iris de los que se desea pasar por la etapa
%    de extraccion de características, comparacion y decision
%    noparche: Numero de subimagenes que se desea muestrear de cada iris
%    para caracterizar al iris
%    sizeparchealto: Altura de cada subimagen en pixeles
%    sizeparchelargo: Ancho de cada subimagen en pixeles
%    tamaño de la base: Porcentaje del numero de iris diferentes de la base
%    de datos que se van a designar para crear la base de datos ficticia, el
%    numero es de 0 a 1
%
%Salida:
%    pruebas: Matriz con el resultado de las pruebas
%    tiempo: Tiempo total en que fueron llevadas a cabo las pruebas de todas
%    las muestras
load datossa sim nobase
load datosumb
[mm,nn]=size(sim);sesion=1;m=ceil(tamanobase*sum(sum(sim(1,1:...
    maxiris)>0)));mu=0;
tic;
for q=1:mm
    for i=1:maxiris
        if sim(q,i)~=0 && mascara(q,i)==0
            j=sim(q,i);
            cinta_file = strcat('Cintash (bin)\cintah_',int2str(i),'_',...
                int2str(sesion),'_',int2str(j),'.bin');
            mu=mu+1;
            acumulado_p(:,1,:)=parches(cinta_file,noparches,sizeparchealto,...
                sizeparchelargo,coord,col,ren);
            disteuclmin_p(mu,1)=+inf;
            baseorig_p(mu,1)=0;
            for l=1:m
                if nobase(l,1)==i
                    baseorig_p(mu,1)=l;
                end
                disteucl=sqrt(sum(sum((acumulado(:,1,:)-acumulado_p(:,1,:)).^2)));
                if(disteuclmin_p(mu,1)>disteucl)
                    disteuclmin_p(mu,1)=disteucl;
                    noiris_p(mu,:)=[i,j];
                    nobase_p(mu,1)=l;
                end
            end
        end
    end
end
end
tiempo=toc;
pruebas=[baseorig_p,noiris_p,nobase_p,disteuclmin_p];

```

```

%Declaracion de la funcion para obtener los histogramas acumulados de las
%subimagenes muestreadas
function acumulado=parches(cinta_file,noparches,sizeparchealto,...
    sizeparchelargo,coord,col,ren)
%acumulado=parches(cinta_file,noparches,sizeparchealto,sizeparchelargo,
% coord,col,ren)
%
%Argumentos:
% cinta_file: Archivo de la imagen normalizada de la cual se van a
% obtener las subimagenes muestreadas
% noparche: Numero de subimagenes que se desea muestrear de cada iris
% para caracterizar al iris
% sizeparchealto: Altura de cada subimagen en pixeles
% sizeparchelargo: Ancho de cada subimagen en pixeles
% coord: Coordenadas de las subimagenes muestreadas
% col: Numero de columnas de la imagen normalizada
% ren: Numero de renglones de la imagen normalizada
%
%Salida:
% acumulado: Matriz con los histogramas acumulados de las subimagenes
% muestreadas
fid=fopen(cinta_file,'r');
im=fread(fid,[col,ren]);
fclose(fid);
im=im';
for k=1:noparches
    %Lectura del parche en la cinta y obtencion de su histograma acumulado
    parche=im(coord(k,1):coord(k,1)+sizeparchealto-1,coord(k,2):...
        coord(k,2)+sizeparchelargo-1);
    acumulado(:,1,k)=histoimage(parche);
end

%Declaracion de la funcion para crear el histograma acumulado de la imagen
function acumulado=histoimage(im)
%acumulado=histoimage(im)
%
%Argumentos:
% im: Imagen original
%
%Salida:
% acumulado: Vector con el histograma acumulado de la imagen
[fil,col]=size(im);
histograma=zeros(256,1);
acum=zeros(256,1);
for i=1:fil
    for j=1:col
        histograma(ceil(im(i,j))+1,1)=histograma(ceil(im(i,j))+1,1)+1;
    end
end
acumulado(1,1)=histograma(1,1);
for i=2:256
    acumulado(i,1)=histograma(i,1)+acumulado(i-1,1);
end

%Declaracion de la funcion para leer archivos *.bin
function im=readbin(image_file,col,ren)
%im=readbin(image_file,col,ren)
%
```

```

%Argumentos:
% image_file: Imagen que va a leer (*.bin)
% col: numero de columnas de la imagen
% ren: numero de renglones de la imagen
%
%Salida:
% im: Matriz de la imagen
fid=fopen(image_file,'r');
if (fid>0)
    im=fread(fid,[col,ren]);
    fclose(fid);
    im=im';
else
    im=0;
end

%Declaracion de la funcion para crear las coordenadas de las subimagenes
%muestreadas por el metodo de muestreo aleatorio con reposicion
function coord=macon(col,ren,noparche,sizeparchealto,sizeparchelargo)
%coord=macon(col,ren,noparche,sizeparchealto,sizeparchelargo)
%
%Argumentos:
% noparche: Numero de subimagenes que se desea muestrear de cada iris
% para caracterizar al iris
% sizeparchealto: Altura de cada subimagen en pixeles
% sizeparchelargo: Ancho de cada subimagen en pixeles
% col: Numero de columnas de la imagen normalizada
% ren: Numero de renglones de la imagen normalizada
%
%Salida:
% coord: Coordenadas de las subimagenes muestreadas
for i=1:noparche
    coord(i,1)=ceil(rand*(ren-sizeparchealto));
    coord(i,2)=ceil(rand*(col-sizeparchelargo));
end

%Declaracion de la funcion para crear las coordenadas de las subimagenes
%muestreadas por el metodo de muestreo aleatorio sin reposicion
function coord=masin(col,ren,noparche,sizeparchealto,sizeparchelargo)
%coord=masin(col,ren,noparche,sizeparchealto,sizeparchelargo)
%
%Argumentos:
% noparche: Numero de subimagenes que se desea muestrear de cada iris
% para caracterizar al iris
% sizeparchealto: Altura de cada subimagen en pixeles
% sizeparchelargo: Ancho de cada subimagen en pixeles
% col: Numero de columnas de la imagen normalizada
% ren: Numero de renglones de la imagen normalizada
%
%Salida:
% coord: Coordenadas de las subimagenes muestreadas
z = 1;
cinta=zeros(ren,col);ncinta=zeros(ren,col);
while z<=noparche
    x(z) = ceil(rand*(ren-sizeparchealto));
    y(z) = ceil(rand*(col-sizeparchelargo));
    ncinta(x(z):x(z)+sizeparchealto-1,y(z):y(z)+sizeparchelargo-1)=...
        ones(sizeparchealto,sizeparchelargo);
    if max(max(ncinta+cinta))<2
        cinta(x(z):x(z)+sizeparchealto-1,y(z):y(z)+sizeparchelargo-1)=...

```

```

        ones(sizeparchealto,sizeparchelargo);
        z=z+1;
    end
    ncinta=zeros(ren,col);
end
coord = [x' y'];

%Declaracion de la funcion para crear las coordenadas de las subimagenes
%muestreadas por el metodo de muestreo hammersley
function datos=hammersley(col,ren,noparche,sizeparchealto,...
    sizeparchelargo,fact)
%datos=hammersley(col,ren,noparche,sizeparchealto,sizeparchelargo,fact)
%
%Argumentos:
%   noparche: Numero de subimagenes que se desea muestrear de cada iris
%             para caracterizar al iris
%   sizeparchealto: Altura de cada subimagen en pixeles
%   sizeparchelargo: Ancho de cada subimagen en pixeles
%   col: Numero de columnas de la imagen normalizada
%   ren: Numero de renglones de la imagen normalizada
%   fact: Numero de coordenadas que van a ser creadas, con el fin de que no
%         se empalmen en caso de que esto llegara a ocurrir
%
%Salida:
%   datos: Coordenadas de las subimagenes muestreadas
primo=ceil(rand*2);
r = i_to_hammersley_sequence (2,noparche*fact,rand*10000,...
    [ceil(rand)*rand('seed')...
    ceil(rand)*rand('seed')],[1 1],[-noparche prime(primo)]);r=r';
r=[ceil(r(:,1)*(ren-sizeparchealto)),ceil(r(:,2)*(col-sizeparchelargo))];
xx=r(:,2);yy=r(:,1);zz=zeros(ceil(noparche*fact),1);
Lx = col-sizeparchelargo;
Ly = ren-sizeparchealto;
z = 1;
cinta=zeros(Lx+sizeparchealto,Ly+sizeparchelargo);
ncinta=zeros(Lx+sizeparchealto,Ly+sizeparchelargo);
cont=0;
while z<=noparche
    cont=cont+1;
    if cont>=noparche*fact
        break
    end
    x(z) = xx(cont,1);
    y(z) = yy(cont,1);
    ncinta(x(z):x(z)+sizeparchealto-1,y(z):y(z)+sizeparchelargo-1)=...
        ones(sizeparchealto,sizeparchelargo);
    if max(max(ncinta+cinta))<2
        cinta(x(z):x(z)+sizeparchealto-1,y(z):y(z)+sizeparchelargo-1)=...
            ones(sizeparchealto,sizeparchelargo);
        datos(z,:)=r(cont,:);
        zz(cont,1)=1;
        z=z+1;
    end
    ncinta=zeros(Lx+sizeparchealto,Ly+sizeparchelargo);
end
for i=1:noparche*fact
    if zz(i)==0
        datos(z,:)=r(i,:);
        z=z+1;
    end
    if z==noparche+1

```

```

        break
    end
end

function r = i_to_hammersley_sequence ( dim_num, n, step, seed, leap, base )

% I_TO_HAMMERSLEY_SEQUENCE: next N elements of an DIM_NUM-dimensional
% Hammersley sequence.
%
% r = i_to_hammersley_sequence ( ndim, n, step, seed, leap, base )
% The program requests input values from the user:
%
% * NDIM, the spatial dimension,
% * N, the number of points to generate,
% * STEP, the index of the first subsequence element to be computed.
% * SEED(1:NDIM), the Hammersley sequence index corresponding to STEP = 0.
% * LEAP(1:NDIM), the successive jumps in the Hammersley sequence.
% * BASE(1:NDIM), the bases (usually distinct primes or -N).
%
% Discussion:
%
% The DIM_NUM-dimensional Hammersley sequence is really DIM_NUM separate
% sequences, each generated by a particular base. If the base is
% greater than 1, a standard 1-dimensional
% van der Corput sequence is generated. But if the base is
% negative, this is a signal that the much simpler sequence  $J/(-BASE)$ 
% is to be generated. For the standard Hammersley sequence, the
% first spatial coordinate uses a base of  $(-N)$ , and subsequent
% coordinates use bases of successive primes (2, 3, 5, 7, 11, ...).
% This program allows the user to specify any combination of bases,
% included nonprimes and repeated values.
%
% This routine selects elements of a "leaped" subsequence of the
% Hammersley sequence. The subsequence elements are indexed by a
% quantity called STEP, which starts at 0. The STEP-th subsequence
% element is simply element
%
%     SEED(1:DIM_NUM) + STEP * LEAP(1:DIM_NUM)
%
% of the original Hammersley sequence.
%
% Modified:
%
%     30 September 2004
%
% Author:
%
%     John Burkardt
%
% Reference:
%
%     J M Hammersley,
%     Monte Carlo methods for solving multivariable problems,
%     Proceedings of the New York Academy of Science,
%     Volume 86, 1960, pages 844-874.
%
%     Ladislav Kocis and William Whiten,
%     Computational Investigations of Low-Discrepancy Sequences,
%     ACM Transactions on Mathematical Software,

```



```

% Volume 23, Number 2, 1997, pages 266-294.
%
% Parameters:
%
% Input, integer DIM_NUM, the spatial dimension.
% 1 <= DIM_NUM is required.
%
% Input, integer N, the number of elements of the sequence.
%
% Input, integer STEP, the index of the subsequence element.
% 0 <= STEP is required.
%
% Input, integer SEED(DIM_NUM), the sequence index corresponding
% to STEP = 0.
%
% Input, integer LEAP(DIM_NUM), the successive jumps in the sequence.
%
% Input, integer BASE(DIM_NUM), the bases.
%
% Output, double precision R(DIM_NUM,N), the next N elements of the
% leaped subsequence, beginning with element STEP.
%
fiddle = 1.0;
dim_num = floor ( dim_num );
n = floor ( n );
step = floor ( step );
seed(1:dim_num) = floor ( seed(1:dim_num) );
leap(1:dim_num) = floor ( leap(1:dim_num) );
base(1:dim_num) = floor ( base(1:dim_num) );
%
% Check the input.
%
if ( ~halham_dim_num_check ( dim_num ) )
    error ( 'I_TO_HAMMERSLEY_SEQUENCE - Fatal error!' );
end

if ( ~halham_n_check ( n ) )
    error ( 'I_TO_HAMMERSLEY_SEQUENCE - Fatal error!' );
end

if ( ~halham_step_check ( step ) )
    error ( 'I_TO_HAMMERSLEY_SEQUENCE - Fatal error!' );
end

if ( ~halham_seed_check ( dim_num, seed ) )
    error ( 'I_TO_HAMMERSLEY_SEQUENCE - Fatal error!' );
end

if ( ~halham_leap_check ( dim_num, leap ) )
    error ( 'I_TO_HAMMERSLEY_SEQUENCE - Fatal error!' );
end

if ( ~hammersley_base_check ( dim_num, base ) )
    error ( 'I_TO_HAMMERSLEY_SEQUENCE - Fatal error!' );
end
%
% Calculate the data.
%
r(1:dim_num,1:n) = 0.0;

for i = 1: dim_num

```

```

if ( 1 < base(i) )

    seed2(1:n) = seed(i) + step * leap(i) : leap(i) : ...
                seed(i) + ( step + n - 1 ) * leap(i);

    base_inv = 1.0 / base(i);

    while ( any ( seed2 ~= 0 ) )
        digit(1:n) = mod ( seed2(1:n), base(i) );
        r(i,1:n) = r(i,1:n) + digit(1:n) * base_inv;
        base_inv = base_inv / base(i);
        seed2(1:n) = floor ( seed2(1:n) / base(i) );
    end

%
% In the following computation, the value of FIDDLE can be:
%
% 0,   for the sequence 0/N, 1/N, ..., N-1/N
% 1,   for the sequence 1/N, 2/N, ..., N/N
% 1/2, for the sequence 1/(2N), 3/(2N), ..., (2*N-1)/(2N)
%
else

    temp(1:n) = seed(i) + step * leap(i) : leap(i) : seed(i) + ...
                ( step + n - 1 ) * leap(i);

    r(i,1:n) = ( mod ( temp(1:n), -base(i) ) + fiddle ) / ( -base(i) );

end

end

function p = prime ( n )

%% PRIME returns returns any of the first PRIME_MAX prime numbers.
%
% Discussion:
%
% PRIME_MAX is 1600, and the largest prime stored is 13499.
%
% Thanks to Bart Vandewoestyne for pointing out a typo, 18 February 2005.
%
% Modified:
%
% 18 February 2005
%
% Author:
%
% John Burkardt
%
% Reference:
%
% Milton Abramowitz and Irene Stegun,
% Handbook of Mathematical Functions,
% US Department of Commerce, 1964, pages 870-873.
%
% Daniel Zwillinger,
% CRC Standard Mathematical Tables and Formulae,
% 30th Edition,
% CRC Press, 1996, pages 95-98.
%
%

```

```

% Parameters:
%
% Input, integer N, the index of the desired prime number.
% In general, is should be true that 0 <= N <= PRIME_MAX.
% N = -1 returns PRIME_MAX, the index of the largest prime available.
% N = 0 is legal, returning PRIME = 1.
%
% Output, integer P, the N-th prime. If N is out of range, P
% is returned as -1.
%
prime_max = 1600;

prime_vector(1:1600) = [
    2, 3, 5, 7, 11, 13, 17, 19, 23, 29, ...
    31, 37, 41, 43, 47, 53, 59, 61, 67, 71, ...
    73, 79, 83, 89, 97, 101, 103, 107, 109, 113, ...
    127, 131, 137, 139, 149, 151, 157, 163, 167, 173, ...
    179, 181, 191, 193, 197, 199, 211, 223, 227, 229, ...
    233, 239, 241, 251, 257, 263, 269, 271, 277, 281, ...
    283, 293, 307, 311, 313, 317, 331, 337, 347, 349, ...
    353, 359, 367, 373, 379, 383, 389, 397, 401, 409, ...
    419, 421, 431, 433, 439, 443, 449, 457, 461, 463, ...
    467, 479, 487, 491, 499, 503, 509, 521, 523, 541, ...
    547, 557, 563, 569, 571, 577, 587, 593, 599, 601, ...
    607, 613, 617, 619, 631, 641, 643, 647, 653, 659, ...
    661, 673, 677, 683, 691, 701, 709, 719, 727, 733, ...
    739, 743, 751, 757, 761, 769, 773, 787, 797, 809, ...
    811, 821, 823, 827, 829, 839, 853, 857, 859, 863, ...
    877, 881, 883, 887, 907, 911, 919, 929, 937, 941, ...
    947, 953, 967, 971, 977, 983, 991, 997, 1009, 1013, ...
    1019, 1021, 1031, 1033, 1039, 1049, 1051, 1061, 1063, 1069, ...
    1087, 1091, 1093, 1097, 1103, 1109, 1117, 1123, 1129, 1151, ...
    1153, 1163, 1171, 1181, 1187, 1193, 1201, 1213, 1217, 1223, ...
    1229, 1231, 1237, 1249, 1259, 1277, 1279, 1283, 1289, 1291, ...
    1297, 1301, 1303, 1307, 1319, 1321, 1327, 1361, 1367, 1373, ...
    1381, 1399, 1409, 1423, 1427, 1429, 1433, 1439, 1447, 1451, ...
    1453, 1459, 1471, 1481, 1483, 1487, 1489, 1493, 1499, 1511, ...
    1523, 1531, 1543, 1549, 1553, 1559, 1567, 1571, 1579, 1583, ...
    1597, 1601, 1607, 1609, 1613, 1619, 1621, 1627, 1637, 1657, ...
    1663, 1667, 1669, 1693, 1697, 1699, 1709, 1721, 1723, 1733, ...
    1741, 1747, 1753, 1759, 1777, 1783, 1787, 1789, 1801, 1811, ...
    1823, 1831, 1847, 1861, 1867, 1871, 1873, 1877, 1879, 1889, ...
    1901, 1907, 1913, 1931, 1933, 1949, 1951, 1973, 1979, 1987, ...
    1993, 1997, 1999, 2003, 2011, 2017, 2027, 2029, 2039, 2053, ...
    2063, 2069, 2081, 2083, 2087, 2089, 2099, 2111, 2113, 2129, ...
    2131, 2137, 2141, 2143, 2153, 2161, 2179, 2203, 2207, 2213, ...
    2221, 2237, 2239, 2243, 2251, 2267, 2269, 2273, 2281, 2287, ...
    2293, 2297, 2309, 2311, 2333, 2339, 2341, 2347, 2351, 2357, ...
    2371, 2377, 2381, 2383, 2389, 2393, 2399, 2411, 2417, 2423, ...
    2437, 2441, 2447, 2459, 2467, 2473, 2477, 2503, 2521, 2531, ...
    2539, 2543, 2549, 2551, 2557, 2579, 2591, 2593, 2609, 2617, ...
    2621, 2633, 2647, 2657, 2659, 2663, 2671, 2677, 2683, 2687, ...
    2689, 2693, 2699, 2707, 2711, 2713, 2719, 2729, 2731, 2741, ...
    2749, 2753, 2767, 2777, 2789, 2791, 2797, 2801, 2803, 2819, ...
    2833, 2837, 2843, 2851, 2857, 2861, 2879, 2887, 2897, 2903, ...
    2909, 2917, 2927, 2939, 2953, 2957, 2963, 2969, 2971, 2999, ...
    3001, 3011, 3019, 3023, 3037, 3041, 3049, 3061, 3067, 3079, ...
    3083, 3089, 3109, 3119, 3121, 3137, 3163, 3167, 3169, 3181, ...
    3187, 3191, 3203, 3209, 3217, 3221, 3229, 3251, 3253, 3257, ...
    3259, 3271, 3299, 3301, 3307, 3313, 3319, 3323, 3329, 3331, ...
    3343, 3347, 3359, 3361, 3371, 3373, 3389, 3391, 3407, 3413, ...

```

3433, 3449, 3457, 3461, 3463, 3467, 3469, 3491, 3499, 3511, ...  
3517, 3527, 3529, 3533, 3539, 3541, 3547, 3557, 3559, 3571, ...  
3581, 3583, 3593, 3607, 3613, 3617, 3623, 3631, 3637, 3643, ...  
3659, 3671, 3673, 3677, 3691, 3697, 3701, 3709, 3719, 3727, ...  
3733, 3739, 3761, 3767, 3769, 3779, 3793, 3797, 3803, 3821, ...  
3823, 3833, 3847, 3851, 3853, 3863, 3877, 3881, 3889, 3907, ...  
3911, 3917, 3919, 3923, 3929, 3931, 3943, 3947, 3967, 3989, ...  
4001, 4003, 4007, 4013, 4019, 4021, 4027, 4049, 4051, 4057, ...  
4073, 4079, 4091, 4093, 4099, 4111, 4127, 4129, 4133, 4139, ...  
4153, 4157, 4159, 4177, 4201, 4211, 4217, 4219, 4229, 4231, ...  
4241, 4243, 4253, 4259, 4261, 4271, 4273, 4283, 4289, 4297, ...  
4327, 4337, 4339, 4349, 4357, 4363, 4373, 4391, 4397, 4409, ...  
4421, 4423, 4441, 4447, 4451, 4457, 4463, 4481, 4483, 4493, ...  
4507, 4513, 4517, 4519, 4523, 4547, 4549, 4561, 4567, 4583, ...  
4591, 4597, 4603, 4621, 4637, 4639, 4643, 4649, 4651, 4657, ...  
4663, 4673, 4679, 4691, 4703, 4721, 4723, 4729, 4733, 4751, ...  
4759, 4783, 4787, 4789, 4793, 4799, 4801, 4813, 4817, 4831, ...  
4861, 4871, 4877, 4889, 4903, 4909, 4919, 4931, 4933, 4937, ...  
4943, 4951, 4957, 4967, 4969, 4973, 4987, 4993, 4999, 5003, ...  
5009, 5011, 5021, 5023, 5039, 5051, 5059, 5077, 5081, 5087, ...  
5099, 5101, 5107, 5113, 5119, 5147, 5153, 5167, 5171, 5179, ...  
5189, 5197, 5209, 5227, 5231, 5233, 5237, 5261, 5273, 5279, ...  
5281, 5297, 5303, 5309, 5323, 5333, 5347, 5351, 5381, 5387, ...  
5393, 5399, 5407, 5413, 5417, 5419, 5431, 5437, 5441, 5443, ...  
5449, 5471, 5477, 5479, 5483, 5501, 5503, 5507, 5519, 5521, ...  
5527, 5531, 5557, 5563, 5569, 5573, 5581, 5591, 5623, 5639, ...  
5641, 5647, 5651, 5653, 5657, 5659, 5669, 5683, 5689, 5693, ...  
5701, 5711, 5717, 5737, 5741, 5743, 5749, 5779, 5783, 5791, ...  
5801, 5807, 5813, 5821, 5827, 5839, 5843, 5849, 5851, 5857, ...  
5861, 5867, 5869, 5879, 5881, 5897, 5903, 5923, 5927, 5939, ...  
5953, 5981, 5987, 6007, 6011, 6029, 6037, 6043, 6047, 6053, ...  
6067, 6073, 6079, 6089, 6091, 6101, 6113, 6121, 6131, 6133, ...  
6143, 6151, 6163, 6173, 6197, 6199, 6203, 6211, 6217, 6221, ...  
6229, 6247, 6257, 6263, 6269, 6271, 6277, 6287, 6299, 6301, ...  
6311, 6317, 6323, 6329, 6337, 6343, 6353, 6359, 6361, 6367, ...  
6373, 6379, 6389, 6397, 6421, 6427, 6449, 6451, 6469, 6473, ...  
6481, 6491, 6521, 6529, 6547, 6551, 6553, 6563, 6569, 6571, ...  
6577, 6581, 6599, 6607, 6619, 6637, 6653, 6659, 6661, 6673, ...  
6679, 6689, 6691, 6701, 6703, 6709, 6719, 6733, 6737, 6761, ...  
6763, 6779, 6781, 6791, 6793, 6803, 6823, 6827, 6829, 6833, ...  
6841, 6857, 6863, 6869, 6871, 6883, 6899, 6907, 6911, 6917, ...  
6947, 6949, 6959, 6961, 6967, 6971, 6977, 6983, 6991, 6997, ...  
7001, 7013, 7019, 7027, 7039, 7043, 7057, 7069, 7079, 7103, ...  
7109, 7121, 7127, 7129, 7151, 7159, 7177, 7187, 7193, 7207, ...  
7211, 7213, 7219, 7229, 7237, 7243, 7247, 7253, 7283, 7297, ...  
7307, 7309, 7321, 7331, 7333, 7349, 7351, 7369, 7393, 7411, ...  
7417, 7433, 7451, 7457, 7459, 7477, 7481, 7487, 7489, 7499, ...  
7507, 7517, 7523, 7529, 7537, 7541, 7547, 7549, 7559, 7561, ...  
7573, 7577, 7583, 7589, 7591, 7603, 7607, 7621, 7639, 7643, ...  
7649, 7669, 7673, 7681, 7687, 7691, 7699, 7703, 7717, 7723, ...  
7727, 7741, 7753, 7757, 7759, 7789, 7793, 7817, 7823, 7829, ...  
7841, 7853, 7867, 7873, 7877, 7879, 7883, 7901, 7907, 7919, ...  
7927, 7933, 7937, 7949, 7951, 7963, 7993, 8009, 8011, 8017, ...  
8039, 8053, 8059, 8069, 8081, 8087, 8089, 8093, 8101, 8111, ...  
8117, 8123, 8147, 8161, 8167, 8171, 8179, 8191, 8209, 8219, ...  
8221, 8231, 8233, 8237, 8243, 8263, 8269, 8273, 8287, 8291, ...  
8293, 8297, 8311, 8317, 8329, 8353, 8363, 8369, 8377, 8387, ...  
8389, 8419, 8423, 8429, 8431, 8443, 8447, 8461, 8467, 8501, ...  
8513, 8521, 8527, 8537, 8539, 8543, 8563, 8573, 8581, 8597, ...  
8599, 8609, 8623, 8627, 8629, 8641, 8647, 8663, 8669, 8677, ...  
8681, 8689, 8693, 8699, 8707, 8713, 8719, 8731, 8737, 8741, ...

```

8747, 8753, 8761, 8779, 8783, 8803, 8807, 8819, 8821, 8831, ...
8837, 8839, 8849, 8861, 8863, 8867, 8887, 8893, 8923, 8929, ...
8933, 8941, 8951, 8963, 8969, 8971, 8999, 9001, 9007, 9011, ...
9013, 9029, 9041, 9043, 9049, 9059, 9067, 9091, 9103, 9109, ...
9127, 9133, 9137, 9151, 9157, 9161, 9173, 9181, 9187, 9199, ...
9203, 9209, 9221, 9227, 9239, 9241, 9257, 9277, 9281, 9283, ...
9293, 9311, 9319, 9323, 9337, 9341, 9343, 9349, 9371, 9377, ...
9391, 9397, 9403, 9413, 9419, 9421, 9431, 9433, 9437, 9439, ...
9461, 9463, 9467, 9473, 9479, 9491, 9497, 9511, 9521, 9533, ...
9539, 9547, 9551, 9587, 9601, 9613, 9619, 9623, 9629, 9631, ...
9643, 9649, 9661, 9677, 9679, 9689, 9697, 9719, 9721, 9733, ...
9739, 9743, 9749, 9767, 9769, 9781, 9787, 9791, 9803, 9811, ...
9817, 9829, 9833, 9839, 9851, 9857, 9859, 9871, 9883, 9887, ...
9901, 9907, 9923, 9929, 9931, 9941, 9949, 9967, 9973, 10007, ...
10009, 10037, 10039, 10061, 10067, 10069, 10079, 10091, 10093, 10099, ...
10103, 10111, 10133, 10139, 10141, 10151, 10159, 10163, 10169, 10177, ...
10181, 10193, 10211, 10223, 10243, 10247, 10253, 10259, 10267, 10271, ...
10273, 10289, 10301, 10303, 10313, 10321, 10331, 10333, 10337, 10343, ...
10357, 10369, 10391, 10399, 10427, 10429, 10433, 10453, 10457, 10459, ...
10463, 10477, 10487, 10499, 10501, 10513, 10529, 10531, 10559, 10567, ...
10589, 10597, 10601, 10607, 10613, 10627, 10631, 10639, 10651, 10657, ...
10663, 10667, 10687, 10691, 10709, 10711, 10723, 10729, 10733, 10739, ...
10753, 10771, 10781, 10789, 10799, 10831, 10837, 10847, 10853, 10859, ...
10861, 10867, 10883, 10889, 10891, 10903, 10909, 10937, 10939, 10949, ...
10957, 10973, 10979, 10987, 10993, 11003, 11027, 11047, 11057, 11059, ...
11069, 11071, 11083, 11087, 11093, 11113, 11117, 11119, 11131, 11149, ...
11159, 11161, 11171, 11173, 11177, 11197, 11213, 11239, 11243, 11251, ...
11257, 11261, 11273, 11279, 11287, 11299, 11311, 11317, 11321, 11329, ...
11351, 11353, 11369, 11383, 11393, 11399, 11411, 11423, 11437, 11443, ...
11447, 11467, 11471, 11483, 11489, 11491, 11497, 11503, 11519, 11527, ...
11549, 11551, 11579, 11587, 11593, 11597, 11617, 11621, 11633, 11657, ...
11677, 11681, 11689, 11699, 11701, 11717, 11719, 11731, 11743, 11777, ...
11779, 11783, 11789, 11801, 11807, 11813, 11821, 11827, 11831, 11833, ...
11839, 11863, 11867, 11887, 11897, 11903, 11909, 11923, 11927, 11933, ...
11939, 11941, 11953, 11959, 11969, 11971, 11981, 11987, 12007, 12011, ...
12037, 12041, 12043, 12049, 12071, 12073, 12097, 12101, 12107, 12109, ...
12113, 12119, 12143, 12149, 12157, 12161, 12163, 12197, 12203, 12211, ...
12227, 12239, 12241, 12251, 12253, 12263, 12269, 12277, 12281, 12289, ...
12301, 12323, 12329, 12343, 12347, 12373, 12377, 12379, 12391, 12401, ...
12409, 12413, 12421, 12433, 12437, 12451, 12457, 12473, 12479, 12487, ...
12491, 12497, 12503, 12511, 12517, 12527, 12539, 12541, 12547, 12553, ...
12569, 12577, 12583, 12589, 12601, 12611, 12613, 12619, 12637, 12641, ...
12647, 12653, 12659, 12671, 12689, 12697, 12703, 12713, 12721, 12739, ...
12743, 12757, 12763, 12781, 12791, 12799, 12809, 12821, 12823, 12829, ...
12841, 12853, 12889, 12893, 12899, 12907, 12911, 12917, 12919, 12923, ...
12941, 12953, 12959, 12967, 12973, 12979, 12983, 13001, 13003, 13007, ...
13009, 13033, 13037, 13043, 13049, 13063, 13093, 13099, 13103, 13109, ...
13121, 13127, 13147, 13151, 13159, 13163, 13171, 13177, 13183, 13187, ...
13217, 13219, 13229, 13241, 13249, 13259, 13267, 13291, 13297, 13309, ...
13313, 13327, 13331, 13337, 13339, 13367, 13381, 13397, 13399, 13411, ...
13417, 13421, 13441, 13451, 13457, 13463, 13469, 13477, 13487, 13499 ];

if ( n == -1 )
    p = prime_max;
elseif ( n == 0 )
    p = 1;
elseif ( n <= prime_max )
    p = prime_vector(n);
else
    p = -1;
end

```

```
function value = halham_dim_num_check ( dim_num )

%% HALHAM_DIM_NUM_CHECK checks DIM_NUM for a Halton or Hammersley sequence.
%%
%% Modified:
%%
%% 04 August 2005
%%
%% Author:
%%
%% John Burkardt
%%
%% Parameters:
%%
%% Input, integer DIM_NUM, the new value for the dimension.
%%
%% Output, logical VALUE, is true if DIM_NUM is legal.
%%
if ( dim_num < 1 )
    fprintf ( 1, '\n' );
    fprintf ( 1, 'HALHAM_DIM_NUM_SET - Fatal error!\n' );
    fprintf ( 1, ' Input value of DIM_NUM < 1!\n' );
    fprintf ( 1, ' DIM_NUM = %d\n', dim_num );
    value = 0;
else
    value = 1;
end

function value = halham_n_check ( n )

%% HALHAM_N_CHECK checks N for a Halton or Hammersley sequence.
%%
%% Modified:
%%
%% 21 September 2004
%%
%% Author:
%%
%% John Burkardt
%%
%% Parameters:
%%
%% Input, integer N, the number of points in the subsequence.
%%
%% Output, logical VALUE, is true if N is legal.
%%
if ( n < 1 )
    fprintf ( 1, '\n' );
    fprintf ( 1, 'HALHAM_N_CHECK - Fatal error!\n' );
    fprintf ( 1, ' Input value of N < 1!\n' );
    fprintf ( 1, ' N = %d\n', n );
    value = 0;
else
    value = 1;
end

function value = halham_step_check ( step )
```

```

%% HALHAM_STEP_CHECK checks STEP for a Halton or Hammersley sequence.
%
% Modified:
%
%   21 September 2004
%
% Author:
%
%   John Burkardt
%
% Parameters:
%
%   Input, integer STEP, the step of the leaped subsequence.
%
%   Output, logical VALUE, is true if STEP is legal.
%
if ( step < 0 )
    fprintf ( 1, '\n' );
    fprintf ( 1, 'HALHAM_STEP_CHECK - Fatal error!\n' );
    fprintf ( 1, '   Input value of STEP < 0!\n' );
    fprintf ( 1, '   STEP = %d\n', step );
    value = 0;
else
    value = 1;
end

function value = halham_seed_check ( dim_num, seed )

%% HALHAM_SEED_SET checks SEED for a Halton or Hammersley sequence.
%
% Modified:
%
%   21 September 2004
%
% Author:
%
%   John Burkardt
%
% Parameters:
%
%   Input, integer DIM_NUM, the spatial dimension.
%
%   Input, integer SEED(1:DIM_NUM), the seed of the leaped subsequence.
%   Each base should be greater than 1.
%
%   Output, logical VALUE, is true if SEED is legal.
%
if ( any ( seed(1:dim_num) < 0 ) )
    fprintf ( 1, '\n' );
    fprintf ( 1, 'HALHAM_SEED_CHECK - Fatal error!\n' );
    fprintf ( 1, '   At least one of the input seeds is < 0!\n' );
    value = 0;
else
    value = 1;
end

function value = halham_leap_check ( dim_num, leap )

%% HALHAM_LEAP_CHECK checks LEAP for a Halton or Hammersley sequence.
%
% Modified:

```

```

%
%   21 September 2004
%
% Author:
%
%   John Burkardt
%
% Parameters:
%
%   Input, integer LEAP(1:DIM_NUM), the leap vector.
%   Each entry should be 1 or greater. Only the integer
%   part of a base is used.
%
%   Output, logical VALUE, is true if LEAP is legal.
%
if ( any ( leap(1:dim_num) < 1 ) )
    fprintf ( 1, '\n' );
    fprintf ( 1, 'HALHAM_LEAP_CHECK - Fatal error!\n' );
    fprintf ( 1, ' At least one of the input leap entries is <= 1!\n' );
    value = 0;
else
    value = 1;
end

function value = hammersley_base_check ( dim_num, base )

%% HAMMERSLEY_BASE_CHECK checks BASE for a Hammersley sequence.
%
% Modified:
%
%   21 September 2004
%
% Author:
%
%   John Burkardt
%
% Parameters:
%
%   Input, integer BASE(1:DIM_NUM), the bases.
%   Bases should not be 0 or 1.
%
%   Output, logical VALUE, is true if BASE is legal.
%
if ( any ( base(1:dim_num) == 0 ) | any ( base(1:dim_num) == 1 ) )
    fprintf ( 1, '\n' );
    fprintf ( 1, 'HAMMERSLEY_BASE_CHECK - Fatal error!\n' );
    fprintf ( 1, ' BASE(I) = 0 or 1 for some I!\n' );
    value = 0;
else
    value = 1;
end

```

### E.2.3. Optimización del muestreo Hammersley por recocido simulado

```

function optimo=recocidosimulado(maximo,base,N,ft,delta,no,sil,si2)
%optimo: Encuentra los parametros optimos del metodo de muestreo Hammersley
% para la maximizacion del desempeño
%
%optimo=recocidosimulado(maximo,base,N,ft,delta,no,sil,si2)
%

```



```

%
%Argumentos:
%   maximo: Numero maximo de iris de los que se desea para obtener los
%           parametros optimos por recocido simulado
%   base: Numero de base que se va a usar para la prueba, depende de la
%         base de datos y el nivel de ruido.
%   N: Numero de iteraciones
%   ft: Factor de decrecimiento de la temperatura
%   delta: Amplitud de cambio de cada parametro
%   no: Numero inicial de subimagenes que se desea muestrear de cada iris
%       para caracterizar al iris
%   sizeparchealto: Altura inicial de cada subimagen en pixeles
%   sizeparchelargo: Ancho inicial de cada subimagen en pixeles
%
%Salida:
%   optimo: Matriz con los resultados obtenidos de las iteraciones de
%           recocido simulado, se guardan los valores de desempeño y sus
%           parametros de cada iteracion
%
%Recursos necesarios:
%   Base de datos: UBIRIS
%   Archivos:
%       1. seleccion100.bin: Base de datos de tamaño 100%
%       2. seleccion90.bin: Base de datos de tamaño 90%
%       3. seleccion80.bin: Base de datos de tamaño 80%
%       4. seleccion70.bin: Base de datos de tamaño 70%
%       5. seleccion60.bin: Base de datos de tamaño 60%
%       6. seleccion50.bin: Base de datos de tamaño 50%
%   Carpetas:
%       1. Cintas (bin): Se generan las imagenes normalizadas de los iris
%           despues de pasar por la fase de ecualizacion por histogramas.
%           (Tipo de archivos: *.bin)
%
%Nota: Las funciones que no se agregan son similares a las del programa
%   sampling
%
% Autor:
% Saul Montes de Oca Armeaga
% saul_montesdeoca@yahoo.com
% Mestria con Especialidad en Automatizacion
% Instituto Tecnologico y de Estudios Superiores de Monterrey
% Diciembre de 2005

% Todas las imagenes normalizadas de los iris deben estar en la carpeta de
% Cintas(bin) aunque solo seran utilizadas las imagenes de la base de datos
% seleccionada. Los archivos son generados en la carpeta correspondiente al
% tipo de muestreo seleccionado.
% Numero de iris y muestras de cada iris
nn=241;mm=5;
switch base,
    case 100,
        sim_file = 'seleccion100.bin';
    case 90,
        sim_file = 'seleccion90.bin';
    case 80,
        sim_file = 'seleccion80.bin';
    case 70,
        sim_file = 'seleccion70.bin';
    case 60,
        sim_file = 'seleccion60.bin';
    case 50,

```

```

        sim_file = 'seleccion50.bin';
    end
    sim=readbin(sim_file,nn,mm,'char');
    mascara=zeros(mm,nn);
    while sum(mascara(1,:))<ceil(.75*maximo),
        randumb=ceil(rand*(maximo));
        if mascara(1,randumb)==0 && sim(1,randumb)~=0
            mascara(1,randumb)=1;
        end
    end
    optimo=zeros(N,6);
    optimo(1,4)=no;
    optimo(1,5)=si1;
    optimo(1,6)=si2;
    [optimo(1,1),optimo(1,2)]=muestreo(maximo,optimo(1,4),optimo(1,5),...
        optimo(1,6),base,sim,mascara);
    T_i = 1.0; % Temperatura inicial

% Iteraciones de recocido simulado
for i=2:N
    u=rand;
    estimado(1,4)=min(max(10,ceil(optimo(1,4)+delta*4*randn)),k);
    estimado(1,5)=min(max(5,ceil(optimo(1,5)+delta*randn)),m);
    estimado(1,6)=min(max(5,ceil(optimo(1,6)+delta*3.6*randn)),n);
    [estimado(1,1),estimado(1,2)]=muestreo(maximo,estimado(1,4),...
        estimado(1,5),estimado(1,6),base,sim,mascara);
    optimodos=hist(i,1);
    alpha=(estimado(1,1)/optimo(i-1,1))^(1/T_i);
    if u<alpha
        optimo(i,:)=estimado(1,:);
    else
        optimo(i,:)=optimo(i-1,:);
    end
    T_i = T_i*ft; % Ajuste de la temperatura
    save sa
    [i,optimo(i,:)]
end

%Declaracion de la funcion que realiza la extraccion de caracteristicas,
%comparacion y decision de la base de datos
function [efectividad,tiempo]=muestreo(maximo,noparche,sizeparchealto,...
    sizeparchelargo,base,sim,mascara)
%[efectividad,tiempo]=muestreo(maximo,noparche,sizeparchealto,
% sizeparchelargo,base,sim,mascara)
%
%Argumentos:
% maximo: Numero maximo de iris de los que se desea para obtener los
% parametros optimos por recocido simulado
% noparche: Numero de subimagenes que se desea muestrear de cada iris
% para caracterizar al iris
% sizeparchealto: Altura de cada subimagen en pixeles
% sizeparchelargo: Ancho de cada subimagen en pixeles
% base: Numero de base que se va a usar para la prueba, depende de la
% base de datos y el nivel de ruido
% sim: Matriz que contiene los numeros de iris que pertenecen a la base
% de datos
% mascara: Matriz que define el numero de muestras utilizadas en la fase
% de creacion de las plantillas
%
%Salida:

```

```

% efectividad: Desempeño del la simulacion de las pruebas
% tiempo: Tiempo total en que fueron llevadas a cabo las pruebas de todas
% las muestras
mm=5;nn=241;
% Creacion de las coordenadas por el metodo de muestreo Hammersley
coord=hammersley(noparche,sizeparchealto,sizeparchelargo,2);
% Creacion de las plantillas para todas las pruebas
[acumulado,nocinta]=templates(coord,mm,nn,maximo,noparche,...
    sizeparchealto,sizeparchelargo,sim,mascara);
% Creacion de los umbrales
[nobase_u,noiris_u,baseorig_u,disteuclmin_u]=umbral(mm,nn,maximo,...
    acumulado,coord,noparche,sizeparchealto,sizeparchelargo,base);
% Simulacion de las pruebas
[nobase_p,noiris_p,baseorig_p,disteuclmin_p,tiempo]=prueba(mm,nn,...
    maximo,acumulado,coord,noparche,sizeparchealto,sizeparchelargo,base);
% Analisis de los resultados
tprueba=maxvalid([nobase_u,noiris_u,baseorig_u,disteuclmin_u],...
    [baseorig_p,noiris_p,nobase_p,disteuclmin_p]);
efectividad=max(tprueba(11,:));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DECLARACION DE FUNCIONES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Declaracion de la funcion para crear las plantillas de todas las pruebas
funcion [acumulado,nocinta]=templates(coord,mm,nn,max,noparches,...
    sizeparchealto,sizeparchelargo,sim,mascara)
%[acumulado,nocinta]=templates(coord,mm,nn,max,noparches,sizeparchealto,
% sizeparchelargo,sim,mascara)
%
%Argumentos:
% coord: Coordenadas de las subimagenes muestreadas
% mm: Numero de muestras de cada iris
% nn: Numero de iris diferentes de la base de datos
% max: Numero maximo de iris del que se desea crear las plantillas
% noparche: Numero de subimagenes que se desea muestrear de cada iris
% para caracterizar al iris
% sizeparchealto: Altura de cada subimagen en pixeles
% sizeparchelargo: Ancho de cada subimagen en pixeles
% sim: Matriz que contiene los numeros de iris que pertenecen a la base
% de datos
% mascara: Matriz que define el numero de muestras utilizadas en la fase
% de creacion de las plantillas
%
%Salida:
% acumulado: Matriz que contiene los histogramas acumulados de las
% subimagenes de todas las plantillas.
% nocinta: Numero de iris usados en la base de datos ficticia
col=720;ren=200;sesion=1;iris=max;sample=6;m=0;
for k=1:sesion
    for i=1:iris
        if mascara(1,i)~=0 && sim(1,i)~=0
            for j=1:sample
                if sim(1,i)==j
                    cinta_file = strcat('Cintash (bin)\cintah_',int2str(i),'_',...
                        int2str(k),'_',int2str(j),'.bin');
                    fid=fopen(cinta_file,'r');
                    if (fid>0)
                        m=m+1;
                end
            end
        end
    end
end

```

```

                [acumulado(:,m,:)] = parches(cinta_file, noparches, ...
                sizeparchealto, sizeparchelargo, coord, col, ren);
                nocinta(m,:) = [i, k, j];
            end
        end
    end
end
end
end
end
end
fidn = fopen('nocinta.bin', 'wb');
fwrite(fidn, nocinta, 'char');
fclose(fidn);
fid = fopen('mascara.bin', 'wb');
fwrite(fid, mascara, 'char');
fclose(fid);

%Declaracion para la funcion de que obtiene la efectividad barriendo un
%determinado umbral
function tprueba = maxvalid(umbrales, pruebas)
tprueba = maxvalid(umbrales, pruebas)
%
%Argumentos:
%  umbrales: Matriz con el resultado de los umbrales
%  pruebas: Matriz con el resultado de las pruebas
%
%Salida:
%  tprueba: Vector con los resultados de efectividad maxima de un
%           intervalo de umbral
cont = 0;
for i = .3:.01:.9
    cont = cont + 1;
    tprueba(:, cont) = validacion(umbrales, pruebas, i);
end

%Declaracion para la funcion que obtiene la efectividad dado un umbral
function tabla = validacion(umbrales, pruebas, test)
tabla = validacion(umbrales, pruebas, test)
%
%Argumentos:
%  umbrales: Matriz con el resultado de los umbrales
%  pruebas: Matriz con el resultado de las pruebas
%  test: Umbral del cual se obtendra la efectividad
%
%Salida:
%  tabla: Vector con los resultados de efectividad del umbral test
facumb = test;
[dim, x] = size(umbrales);
past = 0; fail = 0; contpast = 0; contfail = 0;
for i = 1:dim
    if umbrales(i, 1) == umbrales(i, 5)
        past = past + umbrales(i, 6);
        contpast = contpast + 1;
    else
        fail = fail + umbrales(i, 6);
        contfail = contfail + 1;
    end
end
umbr = (((fail/contfail) - (past/contpast)) * facumb) + (past/contpast);
[dim, x] = size(pruebas);
pruebas(:, 7) = pruebas(:, 6) <= umbr;
pruebas(:, 8) = and(pruebas(:, 1) == pruebas(:, 5), pruebas(:, 7) == 1);

```

```

pruebas(:,9)=and(pruebas(:,1)~=pruebas(:,5),pruebas(:,7)==0);
pruebas(:,10)=and(pruebas(:,1)~=pruebas(:,5),pruebas(:,7)==1);
pruebas(:,11)=and(pruebas(:,1)==pruebas(:,5),pruebas(:,7)==0);
tabla=[sum(pruebas(:,8));sum(pruebas(:,9));sum(pruebas(:,10));sum(pruebas(:,11))];
tabla1=tabla/dim;
tabla3=sum(tabla1(1:2,1));
tabla=[test;umbr;tabla;tabla1;tabla3];

```

### E.2.4. Muestreo Hammerley Factorial

```

function samplinghf(maxiris,noparches,sizeparchealto,sizeparchelargo,...
    noparches0,sizeparchealto0,sizeparchelargo0,tamanobase,...
    tamanoumbbral,base,repertir)
%samplinghf: Realiza la extraccion de caracteristicas, comparacion y decision
%de la base de datos por el metodo de muestreo Hammersley factorial
%
%samplinghf(maxiris,noparches,sizeparchealto,sizeparchelargo,noparches0,
% sizeparchealto0,sizeparchelargo0,tamanobase,tamanoumbbral,base,repertir)
%
%
%Argumentos:
% maxiris: Numero maximo de iris de los que se desea pasar por la etapa
% de extraccion de caracteristicas, comparacion y decision
% noparches: Numero de subimagenes que se desea muestrear de cada iris
% para caracterizar al iris del segundo nivel de muestreo
% sizeparchealto: Altura de cada subimagen en pixeles del segundo nivel
% de muestreo
% sizeparchelargo: Ancho de cada subimagen en pixeles del segundo nivel
% de muestreo
% noparches0: Numero de subimagenes que se desea muestrear de cada iris
% para caracterizar al iris del primer nivel de muestreo
% sizeparchealto0: Altura de cada subimagen en pixeles del primer nivel
% de muestreo
% sizeparchelargo0: Ancho de cada subimagen en pixeles del primer nivel
% de muestreo
% tamaño de la base: Porcentaje del numero de iris diferentes de la base
% de datos que se van a designar para crear la base de datos ficticia, el
% numero es de 0 a 1
% tamanoumbbral: Porcentaje del numero de total de las muestras de los
% iris de la base de datos que se van a designar el numero de muestras
% usadas para crear los umbrales de decision.
% base: Numero de base que se va a usar para la prueba, depende de la
% base de datos y el nivel de ruido.
% repertir: Numero con el que se va a guardar el archivo con los
% resultados
%
%Salida:
% umbrales: Matriz con el resultado de los umbrales
% pruebas: Matriz con el resultado de las pruebas
% tiempo: Tiempo total en que fueron llevadas a cabo las pruebas de todas
% las muestras
%
%Recursos necesarios:
% Base de datos: UBIRIS
% Archivos:
% 1. seleccion100.bin: Base de datos de tamaño 100%
% 2. seleccion90.bin: Base de datos de tamaño 90%
% 3. seleccion80.bin: Base de datos de tamaño 80%
% 4. seleccion70.bin: Base de datos de tamaño 70%
% 5. seleccion60.bin: Base de datos de tamaño 60%

```

```

%      6. seleccion50.bin: Base de datos de tamaño 50%
%      Carpetas:
%      1. Cintas (bin): Se generan las imagenes normalizadas de los iris
%      despues de pasar por la fase de ecualizacion por histogramas.
%      (Tipo de archivos: *.bin)
%      2. Resultados (shf): Carpeta donde se guardan los archivos de los
%      resultados del muestreo Hammersley factorial
%
%      Autor:
%      Saul Montes de Oca Armeaga
%      saul_montesdeoca@yahoo.com
%      Mestria con Especialidad en Automatizacion
%      Instituto Tecnologico y de Estudios Superiores de Monterrey
%      Diciembre de 2005

%      Todas las imagenes normalizadas de los iris deben estar en la carpeta de
%      Cintas(bin) aunque solo seran utilizadas las imagenes de la base de datos
%      seleccionada. Los archivos son generados en la carpeta de "Resultados
%      (shf)"
%      Tamaño de las imagenes normalizadas:
col=720;ren=200;
%No. de iris diferentes (iris) y no. maximo de muestras por iris (sample)
iris=241;sample=6;

%Crear las coordenadas con muestreo Hammerley para los dos niveles
coord0=hammersley(col,ren,noparches0,sizeparchealto0,sizeparchelargo0,2);
coord=hammersley(col,ren,noparches,sizeparchealto,sizeparchelargo,2);
save datossa coord0 coord

%Creacion de plantillas para todas las pruebas
template(coord0,coord,col,ren,sample,iris,maxiris,noparches0,...
    sizeparchealto0,sizeparchelargo0,noparches,sizeparchealto,...
    sizeparchelargo,tamanobase,base);

%Creacion de umbrales
umbral(coord0,coord,col,ren,sample,iris,maxiris,noparches,...
    sizeparchealto0,sizeparchelargo0,noparches,sizeparchealto,...
    sizeparchelargo,tamanobase,tamanoumbrales);

%Pruebas sin reposicion
load datossa mascara
[pruebas,tiempo]=prueba(mascara,coord0,coord,col,ren,sample,iris,...
    maxiris,noparches,sizeparchealto,sizeparchelargo,noparches0,...
    sizeparchealto0,sizeparchelargo0,tamanobase);
load datossa
result_file=strcat('Resultados (shf)\resultado_',num2str(noparches),...
    '_',num2str(sizeparchealto),'_',num2str(sizeparchelargo),...
    '_',num2str(tamanobase),'_',num2str(tamanoumbrales),'_',...
    num2str(base),'_',num2str(repetir),'_mat');
save(result_file,'nobase','pruebas','tiempo','umbrales');
delete('datossa.mat');delete('datosumb.mat')

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%DECLARACION DE FUNCIONES
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%Declaracion de la funcion para crear las plantillas de todas las pruebas
function template(coord0,coord,col,ren,sample,iris,maxiris,noparches0,...
    sizeparchealto0,sizeparchelargo0,noparches,sizeparchealto,...

```

```

    sizeparchelargo,tamanobase,base)
%template(coord0,coord,col,ren,sample,iris,maxiris,noparches0,
%   sizeparchealto0,sizeparchelargo0,noparches,sizeparchealto,
%   sizeparchelargo,tamanobase,base)
%
%Argumentos:
% coord0: Coordenadas de las subimagenes muestreadas del primer nivel
% coord: Coordenadas de las subimagenes muestreadas del segundo nivel
% col: Numero de columnas de la imagen normalizada
% ren: Numero de renglones de la imagen normalizada
% sample: Numero maximo de muestras que tiene cada iris
% iris: Numero de iris diferentes que tiene la base de datos
% maxiris: Numero maximo de iris de los que se desea pasar por la etapa
%   de extraccion de caracteristicas, comparacion y decision
% noparches0: Numero de subimagenes del muestreo del primer nivel que se
%   desea muestrear de cada iris para caracterizar al iris
% sizeparchealto0: Altura de cada subimagen en pixeles del muestre del
%   primer nivel
% sizeparchelargo0: Ancho de cada subimagen en pixeles del muestreo del
%   primer nivel
% noparches: Numero de subimagenes del muestreo del segundo nivel que se
%   desea muestrear de cada iris para caracterizar al iris
% sizeparchealto: Altura de cada subimagen en pixeles del muestre del
%   segundo nivel
% sizeparchelargo: Ancho de cada subimagen en pixeles del muestreo del
%   segundo nivel
% tamano de la base: Porcentaje del numero de iris diferentes de la base
%   de datos que se van a designar para crear la base de datos ficticia, el
%   numero es de 0 a 1
% base: Numero de base que se va a usar para la prueba, depende de la
%   base de datos y el nivel de ruido.
sesion=1;m=0;
%Base de datos que va a ocupar, es necesario tener los archivos de
%seleccion para poder llevar a cabo este paso
switch base,
    case 100,
        sim_file = 'seleccion100.bin';
    case 90,
        sim_file = 'seleccion90.bin';
    case 80,
        sim_file = 'seleccion80.bin';
    case 70,
        sim_file = 'seleccion70.bin';
    case 60,
        sim_file = 'seleccion60.bin';
    case 50,
        sim_file = 'seleccion50.bin';
end
sim=readbin(sim_file,iris,sample);
%El objetivo de mascara es saber que iris han sido empleados en cada etapa,
%donde cero significa que no ha sido usado en ninguna etapa y uno que ya
%fue empleado.
[mm,nn]=size(sim);mascara=zeros(sample,iris);
while sum(mascara(1,:))<ceil(tamanobase*sum(sum(sim(1,1:maxiris)>0))),
    randumb=ceil(rand*(maxiris));
    if mascara(1,randumb)==0 && sim(1,randumb)~=0
        mascara(1,randumb)=1;
    end
end
end

for i=1:maxiris

```

```

if mascara(1,i)~=0 && sim(1,i)~=0
    for j=1:sample
        if sim(1,i)==j
            cinta_file = strcat('Cintash (bin)\cintah_',int2str(i),'_',...
                int2str(sesion),'_',int2str(j)'.bin');
            m=m+1;
            acumulado0(:,m,:)=parches(cinta_file,noparches0,...
                sizeparchealto0,sizeparchelargo0,coord0,col,ren);
            acumulado(:,m,:)=parches(cinta_file,noparches,sizeparchealto,...
                sizeparchelargo,coord,col,ren);
            nobase(m,:)=[i,j];
        end
    end
end
end
mascaraorig=mascara;
%Las variables las va a guardar en el archivo datossa.mat
save datossa nobase mascara mascaraorig sim -APPEND
save datosumb acumulado acumulado0

%Declaracion de la funcion para crear los umbrales de todas las pruebas
function umbral(coord0,coord,col,ren,sample,iris,maxiris,noparches,...
    sizeparchealto0,sizeparchelargo0,noparches0,sizeparchealto,...
    sizeparchelargo,tamanobase,tamanoumbbral)
%umbral(coord0,coord,col,ren,sample,iris,maxiris,noparches,sizeparchealto0,
%sizeparchelargo0,noparches0,sizeparchealto,sizeparchelargo,tamanobase,
%tamanoumbbral)
%
%Argumentos:
% coord0: Coordenadas de las subimagenes muestreadas del primer nivel
% coord: Coordenadas de las subimagenes muestreadas del segundo nivel
% col: Numero de columnas de la imagen normalizada
% ren: Numero de renglones de la imagen normalizada
% sample: Numero maximo de muestras que tiene cada iris
% iris: Numero de iris diferentes que tiene la base de datos
% maxiris: Numero maximo de iris de los que se desea pasar por la etapa
% de extraccion de caracteristicas, comparacion y decision
% noparches0: Numero de subimagenes del muestreo del primer nivel que se
% desea muestrear de cada iris para caracterizar al iris
% sizeparchealto0: Altura de cada subimagen en pixeles del muestre del
% primer nivel
% sizeparchelargo0: Ancho de cada subimagen en pixeles del muestreo del
% primer nivel
% noparches: Numero de subimagenes del muestreo del segundo nivel que se
% desea muestrear de cada iris para caracterizar al iris
% sizeparchealto: Altura de cada subimagen en pixeles del muestre del
% segundo nivel
% sizeparchelargo: Ancho de cada subimagen en pixeles del muestreo del
% segundo nivel
% tamaño de la base: Porcentaje del numero de iris diferentes de la base
% de datos que se van a designar para crear la base de datos ficticia, el
% numero es de 0 a 1
% tamanoumbbral: Porcentaje del numero de total de las muestras de los
% iris de la base de datos que se van a designar el numero de muestras
% usadas para crear los umbrales de decision
load datossa nobase mascara sim
load datosumb
mu=0;sesion=1;m=ceil(tamanoumbbral*sum(sum(sim(:,1:maxiris)>0))/2.0);

while sum(mascara(2,:))+sum(mascara(3,:))<m
    randumb=ceil(rand*(maxiris));

```



```

if ((mascara(2,randumb)==0&&sim(2,randumb)~=0) || (mascara(3,...
    randumb)==0&&sim(3,randumb)~=0)) && mascara(1,randumb)...
    ==1 && randumb~=0
    i=randumb; j=sim(2,randumb); mu=mu+1;
    if mascara(2,randumb)==1
        j=sim(3,randumb);
    end
    cinta_file = strcat('Cintash (bin)\cintah_',int2str(i),'_',...
        int2str(sesion),'_',int2str(j),'.bin');
    acumulado_u(:,1,:)=parches(cinta_file,noparches,sizeparchealto,...
        sizeparchelargo,coord,col,ren);
    acumulado0_u(:,1,:)=parches(cinta_file,noparches0,sizeparchealto0,...
        sizeparchelargo0,coord0,col,ren);
    disteuclmin_u(mu,1)=+inf;
    disteuclmin0_u(mu,1)=+inf;
    baseorig_u(mu,1)=0;
    for l=1:ceil(tamanobase*sum(sum(sim(1,1:maxiris)>0)))
        if nobase(l,1)==i
            baseorig_u(mu,1)=l;
        end
        disteucl=sqrt(sum(sum((acumulado(:,1,:)-acumulado_u(:,1,:)).^2)));
        disteucl0=sqrt(sum(sum((acumulado0(:,1,:)-acumulado0_u(:,1,:)).^2)));
        if(disteuclmin_u(mu,1)>disteucl)
            disteuclmin_u(mu,1)=disteucl;
            disteuclmin0_u(mu,1)=disteucl0;
            noiris_u(mu,:)=[i,j];
            nobase_u(mu,1)=l;
        end
    end
    if mascara(2,randumb)==1
        mascara(3,randumb)=1;
    else
        mascara(2,randumb)=1;
    end
end
end

while sum(mascara(2,:))+sum(mascara(3,:))<2*m
    randumb=ceil(rand*(maxiris));
    if ((mascara(2,randumb)==0&&sim(2,randumb)~=0) || (mascara(3,...
        randumb)==0&&sim(3,randumb)~=0)) && mascara(1,randumb)...
        ==0 && randumb~=0
        k=sesion; i=randumb; j=sim(2,randumb); mu=mu+1;
        if mascara(2,randumb)==1
            j=sim(3,randumb);
        end
        cinta_file = strcat('Cintash (bin)\cintah_',int2str(i),'_',...
            int2str(sesion),'_',int2str(j),'.bin');
        acumulado0_u(:,1,:)=parches(cinta_file,noparches0,sizeparchealto0,...
            sizeparchelargo0,coord0,col,ren);
        acumulado_u(:,1,:)=parches(cinta_file,noparches,sizeparchealto,...
            sizeparchelargo,coord,col,ren);
        disteuclmin_u(mu,1)=+inf;
        disteuclmin0_u(mu,1)=+inf;
        baseorig_u(mu,1)=0;
        for l=1:ceil(tamanobase*sum(sum(sim(1,1:maxiris)>0)))
            if nobase(l,1)==i
                baseorig_u(mu,1)=l;
            end
            disteucl0=sqrt(sum(sum((acumulado0(:,1,:)-acumulado0_u(:,1,:)).^2)));
            disteucl=sqrt(sum(sum((acumulado(:,1,:)-acumulado_u(:,1,:)).^2)));

```

```

        if(disteuclmin_u(mu,1)>disteucl)
            disteuclmin_u(mu,1)=disteucl;
            disteuclmin0_u(mu,1)=disteucl0;
            noiris_u(mu,:)=[i,j];
            nobase_u(mu,1)=1;
        end
    end
end
if mascara(2,randumb)==1
    mascara(3,randumb)=1;
else
    mascara(2,randumb)=1;
end
end
end
umbral0=[baseorig_u,noiris_u,nobase_u,disteuclmin_u];
umbral0=[baseorig_u,noiris_u,nobase_u,disteuclmin0_u];
umbral0(:,6)=umbral0(:,1)==umbral0(:,4);
verdadero=sum(umbral0(:,6).*umbral0(:,5))/sum(umbral0(:,6));
umbr=verdadero
save datossa mascara umbrales umbr -APPEND

function tabla=resultado(umbrales,verdadero,falso,test)
umbr=verdadero+(falso-verdadero)*test;
umbrales(:,6)=umbrales(:,5)<=umbr;
umbrales(:,7)=and(umbrales(:,1)==umbrales(:,4),umbrales(:,6)==1);
umbrales(:,8)=and(umbrales(:,1)~=umbrales(:,4),umbrales(:,6)==0);
umbrales(:,9)=and(umbrales(:,1)~=umbrales(:,4),umbrales(:,6)==1);
umbrales(:,10)=and(umbrales(:,1)==umbrales(:,4),umbrales(:,6)==0);
tabla=[sum(umbrales(:,7));sum(umbrales(:,8));sum(umbrales(:,9));...
    sum(umbrales(:,10))];
tabla1=tabla/sum(tabla);
tabla3=sum(tabla1(1:2,1));
tabla=[test;umbr;tabla;tabla1;tabla3];

%Declaracion de la funcion que simula las pruebas
function [pruebas,tiempo]=prueba(mascara,coord0,coord,col,ren,sample...
    ,iris,maxiris,noparches,sizeparchealto,sizeparchelargo,noparches0,...
    sizeparchealto0,sizeparchelargo0,tamanobase)
[pruebas,tiempo]=prueba(mascara,coord,col,ren,sample,iris,maxiris,
%   noparches,sizeparchealto,sizeparchelargo,tamanobase)
%
%Argumentos:
%   mascara: Archivo que ayuda a saber que muestras fueron usadas en las
%   etapas anteriores
%   coord0: Coordenadas de las subimagenes muestreadas del primer nivel
%   coord: Coordenadas de las subimagenes muestreadas del segundo nivel
%   col: Numero de columnas de la imagen normalizada
%   ren: Numero de renglones de la imagen normalizada
%   sample: Numero maximo de muestras que tiene cada iris
%   iris: Numero de iris diferentes que tiene la base de datos
%   maxiris: Numero maximo de iris de los que se desea pasar por la etapa
%   de extraccion de caracteristicas, comparacion y decision
%   noparches0: Numero de subimagenes del muestreo del primer nivel que se
%   desea muestrear de cada iris para caracterizar al iris
%   sizeparchealto0: Altura de cada subimagen en pixeles del muestre del
%   primer nivel
%   sizeparchelargo0: Ancho de cada subimagen en pixeles del muestreo del
%   primer nivel
%   noparches: Numero de subimagenes del muestreo del segundo nivel que se

```

```

% desea muestrear de cada iris para caracterizar al iris
% sizeparchealto: Altura de cada subimagen en pixeles del muestre del
% segundo nivel
% sizeparchelargo: Ancho de cada subimagen en pixeles del muestreo del
% segundo nivel
% tamaño de la base: Porcentaje del numero de iris diferentes de la base
% de datos que se van a designar para crear la base de datos ficticia, el
% numero es de 0 a 1
%
%Salida:
% pruebas: Matriz con el resultado de las pruebas
% tiempo: Tiempo total en que fueron llevadas a cabo las pruebas de todas
% las muestras
load datossa sim nobase umbr
load datosumb
[mm,nn]=size(sim);sesion=1;m=ceil(tamanobase*sum(sum(sim(1,1:...
maxiris)>0)));mu=0;

tic;
for q=1:mm
    for i=1:maxiris
        if sim(q,i)~=0 && mascara(q,i)==0
            j=sim(q,i);
            cinta_file = strcat('Cintash (bin)\cintah_',int2str(i),'_',int2str(...
                sesion),'_',int2str(j),'.bin');
            mu=mu+1;
            acumulado0_p(:,1,:)=parches(cinta_file,noparches0,...
                sizeparchealto0,sizeparchelargo0,coord0,col,ren);
            acumulado_p(:,1,:)=parches(cinta_file,noparches,sizeparchealto,...
                sizeparchelargo,coord,col,ren);
            disteuclmin_p(mu,1)=+inf;
            baseorig_p(mu,1)=0;
            factorial=zeros(m,1);
            for l=1:m
                disteucl=sqrt(sum(sum((acumulado0(:,l,:)-acumulado0_p(:,1...
                    ,:)).^2)));
                if disteucl<umbr
                    factorial(l,1)=1;
                end
            end
            for l=1:m
                if nobase(l,1)==i
                    baseorig_p(mu,1)=1;
                end
                if factorial(l,1)==1 || (sum(factorial)==0&&l==m)
                    disteucl=sqrt(sum(sum((acumulado(:,l,:)-acumulado_p(:,1...
                        ,:)).^2)));
                    if(disteuclmin_p(mu,1)>disteucl)
                        disteuclmin_p(mu,1)=disteucl;
                        fact_p(mu,:)=sum(factorial);
                        noiris_p(mu,:)=[i,j];
                        nobase_p(mu,1)=1;
                    end
                end
            end
        end
    end
end
end
tiempo=toc;
pruebas=[baseorig_p,noiris_p,nobase_p,disteuclmin_p];

```

```

%Declaracion de la funcion para obtener los histogramas acumulados de las
%subimagenes muestreadas
function acumulado=parches(cinta_file,noparches,sizeparchealto,...
    sizeparchelargo,coord,col,ren)
%acumulado=parches(cinta_file,noparches,sizeparchealto,sizeparchelargo,
% coord,col,ren)
%
%
%Argumentos:
% cinta_file: Archivo de la imagen normalizada de la cual se van a
% obtener las subimagenes muestreadas
% noparche: Numero de subimagenes que se desea muestrear de cada iris
% para caracterizar al iris
% sizeparchealto: Altura de cada subimagen en pixeles
% sizeparchelargo: Ancho de cada subimagen en pixeles
% coord: Coordenadas de las subimagenes muestreadas
% col: Numero de columnas de la imagen normalizada
% ren: Numero de renglones de la imagen normalizada
%
%Salida:
% acumulado: Matriz con los histogramas acumulados de las subimagenes
% muestreadas
fid=fopen(cinta_file,'r');
im=fread(fid,[col,ren]);
fclose(fid);
im=im';
for k=1:noparches
    %Lectura del parche en la cinta y obtencion de su histograma acumulado
    parche=im(coord(k,1):coord(k,1)+sizeparchealto-1,coord(k,2):...
        coord(k,2)+sizeparchelargo-1);
    acumulado(:,1,k)=histoimage(parche);
end

%Declaracion de la funcion para crear el histograma acumulado de la imagen
function acumulado=histoimage(im)
%acumulado=histoimage(im)
%
%Argumentos:
% im: Imagen original
%
%Salida:
% acumulado: Vector con el histograma acumulado de la imagen
[fil,col]=size(im);
histograma=zeros(256,1);
acum=zeros(256,1);
for i=1:fil
    for j=1:col
        histograma(ceil(im(i,j))+1,1)=histograma(ceil(im(i,j))+1,1)+1;
    end
end
acumulado(1,1)=histograma(1,1);
for i=2:256
    acumulado(i,1)=histograma(i,1)+acumulado(i-1,1);
end

%Declaracion de la funcion para leer archivos *.bin
function im=readbin(image_file,col,ren)
%im=readbin(image_file,col,ren)
%
%Argumentos:
% image_file: Imagen que va a leer (*.bin)

```

```
% col: numero de columnas de la imagen
% ren: numero de renglones de la imagen
%
%Salida:
% im: Matriz de la imagen
fid=fopen(image_file,'r');
if (fid>0)
    im=fread(fid,[col,ren]);
    fclose(fid);
    im=im';
else
    im=0;
end
```