UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 5

Título: Módulo de autenticación biométrica por reconocimiento de patrones del iris para el módulo de seguridad del CEDIN.

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS INFORMÁTICAS.

Autor

Lidises Beatriz Villavicencio Murguia.

Tutores

Dra. Deysi Fraga Cedré.

Ing. Ariel Peláez González.

Co-Tutor

Ing. Alejandro González Abascal.

La Habana, Mayo de 2013
"Año 55 de la Revolución"

Declaración de autoría.

Declaro ser autora de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

_____

Firma del Autor: Lidises Beatriz Villavicencio Murguia.

_____ _____

Firma del Tutor: Ariel Peláez González. Firma del Tutor: Deysi Fraga Cedré

_____

Firma del Co-Tutor: Alejandro González Abascal.

Datos de contacto.

Tutor: Ing. Ariel Peláez González.

Edad: 25 años.

Ciudadanía: cubano.

Institución: Universidad de Ciencias Informáticas (UCI).

Título: Ingeniero en Ciencias Informáticas.

Categoría Docente:

E-mail: apelaezg@uci.cu

Tutor: Deysi Fraga Cedré.

Edad: 48 años.

Ciudadanía: cubano.

Institución: Universidad de Ciencias Pedagógicas Enrique José Varona.

Título: Doctora en Ciencias Pedagógicas.

Categoría Docente: Profesor auxiliar.

E-mail: deysifc@ucpejv.rimed.cu

Co-Tutor: Ing. Alejandro González Abascal.

Edad: 26 años.

Ciudadanía: cubano.

Institución: Universidad de Ciencias Informáticas (UCI).

Título: Ingeniero en Ciencias Informáticas.

Categoría Docente:

E-mail: agabascal@uci.cu

Dedicatoria

A mis padres,

Porque sin ellos hoy esto no fuera posible.

Agradecimientos

A mis padres por apoyarme y guiarme para convertirme en la persona que soy hoy.
A mi familia por estar siempre ahí cuando la necesito, en especial a mi abuela Romelia.
A Ariel por hacer mi vida parte de la suya.
A Sheyla, Yaimarelis y Yaumara por ser mis compañeras y amigas durante estos 5 años.
A César, Alex, Julio César y Andy por brindarme su amistad y su apoyo desde que nos conocimos.
A Richel por ser mi médico de la UCI como yo le digo.
A Liván, Luis Manuel, Yoandys y todos los que fueron mis compañeros de la FEU.
A todos los que fueron mis compañeros de grupo a lo largo de estos 5 años.
A todos los profesores que tuve durante la carrera por ofrecerme sus conocimientos y su amistad.
A los profesores del tribunal por sus concejos para hacer de este un buen trabajo.
Y a todos los que dieron su granito para que hoy este sueño fuera posible.

Resumen

Los sistemas de seguridad en el mundo actual, se encuentran en un desarrollo creciente producto al gran número de ataques a los que están sometidos de manera constante los sistemas informáticos modernos. Existen varias formas de fortalecer estos sistemas y una de las más eficientes es contar con diversas vías de autenticación, entre las que se destacan aquellas que utilizan patrones biométricos. Con el fin de incrementar estas técnicas en nuestro país para evitar posibles vulnerabilidades se realiza el siguiente trabajo que tiene como objetivo desarrollar un módulo de autenticación biométrica por reconocimiento de patrones del iris.

Se investiga acerca de las principales tecnologías de desarrollo y algoritmos empleados en el mundo para el reconocimiento del iris, con el objetivo de confeccionar una propuesta de los que serán utilizados. Además, se muestra la modelación del sistema a partir de diferentes diagramas y se analiza la implementación del mismo. Por último, se muestran las diferentes pruebas realizadas al módulo para determinar su efectividad.

Palabras clave: Autenticación, biometría, seguridad, sistema, módulo.

Abstract

Security systems in the world are in a growing product development to the large number of attacks that are constantly undergoing modern computer systems. There are several ways to strengthen these systems and one of the most effective is to have various ways of authentication, including highlights those using biometric patterns. In order to develop these techniques in our country to avoid potential vulnerabilities is performed the following work aims to develop a biometric authentication module for iris pattern recognition.

It investigates the key technologies and development employees' worldwide algorithms for iris recognition, with the aim of preparing a proposal that will be used. Also shown is the modeling of the system from different diagrams and discusses the implementation. Finally, shows the various tests conducted to determine effectiveness module.

***Keywords**: Authentication, biometrics, security, system, module.*

Índice

Índice de Tablas

Índice de Figuras

Introducción

La informatización de la sociedad es un proceso global y natural consecuencia de los cambios tecnológicos ocurridos en las últimas décadas. En la actualidad se construyen aplicaciones de alta complejidad con relativa facilidad, esto se ve reflejado en las transacciones bancarias, las cuales en un gran por ciento se realizan de forma digital; y en la seguridad, tanto de instalaciones como de grandes servidores de información, que expande sus horizontes de forma multidireccional. Cuba forma parte de este proceso y esto se evidencia en la Universidad de Ciencias Informáticas (UCI) con la creación de los diferentes centros productivos, destacándose el Centro de Informática Industrial (CEDIN). Entre los principales productos que se desarrollan en este centro se pueden destacar los SCADAs[1], que no son más que sistemas de control y adquisición de datos para gestionar ambientes industriales. Para garantizar la confidencialidad, integridad y disponibilidad de los datos con los que operan estos sistemas, el centro tiene un módulo de seguridad. Actualmente este módulo cuenta con un sistema de vigilancia por cámara y con un sistema de control de acceso basado en roles con dos técnicas de autenticación o verificación de la identidad de los usuarios: mediante el uso de usuario y contraseña y por reconocimiento de huellas digitales. Pero estos métodos no son 100% confiables. Para el caso del uso de usuario y contraseña se muestran vulnerabilidades como la debilidad en las contraseñas o la posible ruptura de estas por avanzados algoritmos de descifrado, mientras que para el reconocimiento de huellas digitales se registran otras como heridas y mutilaciones anatómicas o suciedad y mal estado de la piel.

Teniendo en cuenta que estas técnicas son insuficientes para garantizar la seguridad de los SCADAs se desarrolla una biblioteca de autenticación biométrica a partir de reconocimiento de patrones del iris para la identificación de usuarios, pero esta no muestra resultados efectivos en cada una de sus etapas y a pesar de que aproximadamente el 70% de las pruebas realizadas arrojan resultados satisfactorios, no alcanza los niveles de efectividad requeridos para una biblioteca de este tipo, lo que trae consigo que se pudiera denegar el acceso a aquellos que están autorizados o por el contrario permitirlo

[1]Siglas en Inglés: Supervisory Control And Data Acquisition (Control Supervisor y Adquisición de Datos).

a otros no autorizados, lo que puede provocar daños irreversibles. Por estos motivos no se recomienda el uso de la biblioteca y la misma no se encuentra en funcionamiento.

Ante esta situación se puede definir el siguiente **problema científico**: ¿Cómo lograr que el proceso de reconocimiento de patrones del iris para el módulo de seguridad del CEDIN se realice de manera efectiva?

Para ello, el **objeto de estudio** de esta investigación, se traduce en el proceso de autenticación de usuarios con técnicas biométricas.

Para dar respuesta al problema de la investigación se definió que el **objetivo general** consistiera en: Desarrollar un módulo de autenticación biométrica por reconocimiento de patrones del iris.

Para cumplir con el objetivo la investigación tendrá como **campo de acción**: los sistemas de reconocimiento de patrones del iris.

Conforme a lo planteado como objetivo general se derivan las siguientes **tareas investigativas**:

- Elaboración del marco teórico a partir del estudio del estado del arte que sirva de base para encontrar una solución al problema en cuestión.
- Identificación de las funcionalidades principales en términos de requisitos a partir del estudio de sistemas similares.
- Realización del diseño correspondiente.
- Implementación de algoritmos partiendo de los elementos del diseño.
- Definición de los casos y procesos de prueba.
- Comparación de resultados obtenidos con resultados esperados.

Métodos científicos utilizados.

En el transcurso de la investigación para alcanzar las tareas propuestas se identificaron los siguientes métodos:

Métodos Teóricos:

- **Método Histórico Lógico:** para constatar teóricamente cómo ha evolucionado el desarrollo de los sistemas de autenticación biométrica mediante el análisis de patrones oculares, específicamente mediante reconocimiento del iris, a través de los años.
- **Método Analítico – Sintético:** con el objetivo de analizar la información y la documentación relevante para el desarrollo del software enfatizando en los elementos más importantes que se relacionan con el objeto de estudio.
- **Método Comparativo:** las comparaciones han sido utilizadas de forma exhaustiva en la investigación para lograr tener una visión clara de la complejidad real del problema pues un análisis superficial podría traer serias consecuencias en un ambiente real.

Métodos Empíricos:

- **Consulta de fuentes de información:** consulta de material bibliográfico para conformar el marco teórico.
- **Consulta de especialistas:** para obtener información especializada de las características del iris humano.
- **Realización de pruebas:** para verificar la validez y confiabilidad del resultado.

Método estadístico:

- **Mediante la estadística descriptiva:** con el empleo de tablas, gráficas para valorar los resultados obtenidos con los resultados esperados a fin de determinar la validez y confiabilidad del resultado.

Resultados esperados al concluir el trabajo de Diploma:

Módulo de autenticación biométrica por reconocimiento de patrones del iris para el módulo de seguridad del CEDIN.

La investigación estará estructurada de la siguiente forma:

Capítulo 1: Se sintetiza sobre cómo se realiza el proceso de autenticación de usuarios con técnicas biométricas, específicamente mediante el reconocimiento del iris. Se describe el estado del arte de los sistemas de autenticación de usuarios por patrones del iris y se explican algunas bibliotecas utilizadas en el procesamiento de imágenes digitales.

Capítulo 2: Se describe la solución propuesta en cuanto a metodologías, tecnologías, herramientas y algoritmos para cada una de las etapas del proceso de reconocimiento del iris. Se realiza la modelación del sistema a partir de diagramas de casos de uso, diagramas de secuencia y diagramas de clases. Se plantean los diferentes patrones utilizados para enriquecer la arquitectura del sistema.

Capítulo 3: Se muestran detalles de la implementación a partir de diagramas, así como los casos y resultados de prueba para los casos de uso críticos. Se especifica el estándar de codificación.

Capítulo 1: Fundamentación Teórica.

1.1 Introducción al capítulo

En el presente capítulo se exponen algunas características del iris humano y las fases con que cuenta un sistema de autenticación de usuarios por reconocimiento del iris, así como algunos de los algoritmos que se emplean para dar solución a cada una de estas fases. También, se explican algunas bibliotecas empleadas en el procesamiento de imágenes.

1.2 El iris humano

El iris es un órgano que se encuentra en el ojo humano, compuesto de un tejido fibroso llamado estroma, que conecta músculos encargados de contraer y dilatar la pupila en respuesta a los cambios de iluminación. A pesar de ser un órgano interno, es a la vez visible desde el exterior. (1)

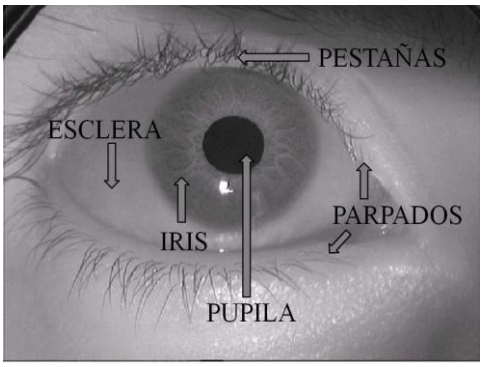

Figura 1: Diagrama del ojo humano

Como se aprecia en la Figura 1, el iris es la parte del ojo comprendida entre la pupila y una capa blanca denominada esclera o esclerótica. Existen muchas razones por las cuales el iris es utilizado para la identificación de individuos, por ejemplo, el iris humano es generado en los primeros meses de gestación y no varía durante toda la vida, no existe evidencia de la existencia de dos iris iguales, incluso en el caso de los iris de dos gemelos o en el ojo izquierdo y derecho de una misma persona. El iris es una característica biométrica muy rica en cuanto a información que permite identificar un individuo, pues posee más de cuatro veces la cantidad de características distinguibles que posee una huella dactilar. Además, su textura pueda ser capturada solo con una simple fotografía y es imposible modificarlo por medios no-quirúrgico, incluso en operaciones como las de cataratas se mantiene constante la textura del iris.

1.3 Etapas de los sistemas de reconocimiento del iris

La mayoría de los sistemas de reconocimiento del iris disponibles comercialmente a la fecha utilizan los algoritmos desarrollados por el físico-matemático alemán John Daugman, quien captura la imagen del ojo de la persona y, a partir de la textura del iris, genera un código del iris que luego es almacenado en una base de datos y utilizado para la identificación de la persona. Este proceso consta de varias etapas como se puede apreciar en la siguiente figura.

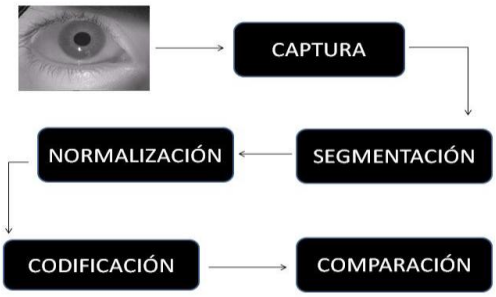

Figura 2: Diagrama en bloque del sistema de reconocimiento del iris.

A continuación se hace una breve descripción del flujo de estas etapas y posteriormente se caracterizarán las mismas con más detalles.

- Primero, se hace una **captura** de una imagen del ojo de la persona.
- Una vez capturada la imagen, es necesario aislar la textura del iris y descartar el ruido producido por los párpados, las pestañas y cualquier reflejo ambiental. Esta etapa se denomina **segmentación**.
- La textura del iris pasa por un proceso de **normalización** que debe permitir invarianza[2] frente a variaciones en el proceso de captura, como por ejemplo el tamaño del iris en la imagen o el diámetro de la pupila.
- Esta textura normalizada pasa por un proceso de **codificación**, donde se obtiene una representación de la textura del iris que simplificará el proceso de identificación y verificación. Una vez generado el código del iris, se pueden realizar dos acciones:
- Almacenar el código en una base de datos para referencia futura, o bien realizar una búsqueda del código en una base de datos existente con el

[2]Propiedad de ser invariante. Invariante, algo que no cambia al aplicarle un conjunto de transformaciones.

objetivo de identificar o verificar la identidad de la persona, lo que se conoce como etapa de **comparación**. (2)

1.3.1 Segmentación

La segmentación consiste en identificar la región que contiene el iris y separarla de otros elementos de la imagen que no contienen información relevante, como la pupila, párpados y pestañas. Es importante la obtención de óptimos resultados en esta etapa, pues en caso contrario se utilizarán datos erróneos en las etapas posteriores, provocando que el código generado contenga errores y la efectividad del sistema sea muy baja.

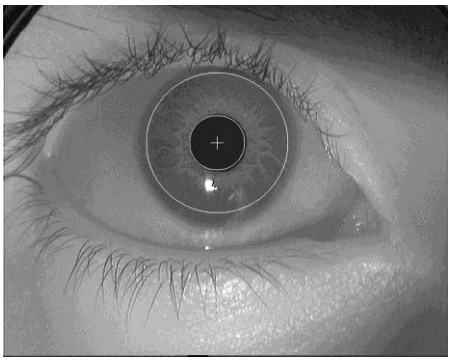

Figura 3: Imagen segmentada.

En el mundo existen diferentes técnicas y algoritmos para obtener los resultados que se muestran en la imagen anterior (Figura 3). Algunos de ellos se describen a continuación por ser muy utilizados para sistemas de este tipo.

Operador íntegro-diferencial

El Operador íntegro-diferencial es una herramienta matemática utilizada por John Daugman para su sistema de reconocimiento del iris. Para el empleo de este operador, Daugman parte de la premisa de que el iris es un anillo circular, ya que esta técnica consiste en calcular la mayor variación de gris para cada punto de la imagen en una región circular y para ello se emplea la siguiente fórmula.

$$\max(r, x_0, y_0) \left| G_\sigma(r) * \frac{\partial}{\partial r} \oint_{r, x_0, y_0} \frac{I(x,y)}{2\pi r} ds \right| \quad (3)$$

Figura 4: Fórmula del operador íntegro-diferencial.

Detección de bordes y transformada de Hough

Esta es otra técnica propuesta por Wildes y muy diferente a la utilizada por Daugman. La misma consiste en realizar un mapa de bordes binario, y sobre ese mapa aplicar una transformada de Hough para la detección de círculos. Este método es considerablemente más estable frente a perturbaciones ruidosas que el propuesto por Daugman pero también contiene la premisa de la circularidad del iris y presenta una menor precisión, aunque muy leve. (3)

Algoritmo Cruz

El algoritmo Cruz a pesar de no ser tan reconocido como los dos anteriores es simple y eficaz, aunque su propósito es resolver la detección de la pupila y no la segmentación del iris en su totalidad. Este algoritmo calcula la mayor variación de gris para cada punto de la imagen describiendo trazas en las direcciones norte, sur, este y oeste; determinando como centro del iris y la pupila el punto donde los valores de estas tazas son lo más similares posibles y como radio de la pupila el valor de la menor de estas trazas, con el objetivo de no dejar fuera de la segmentación algún píxel del iris. (4)

1.3.2 Normalización

Luego de segmentada la imagen se pasa a la etapa de normalización que tiene como objetivo obtener una imagen de dimensiones fijas logrando de esta manera que dos imágenes del mismo iris, adquiridas bajo diferentes condiciones, tengan las mismas características espaciales, haciendo posible su posterior comparación.

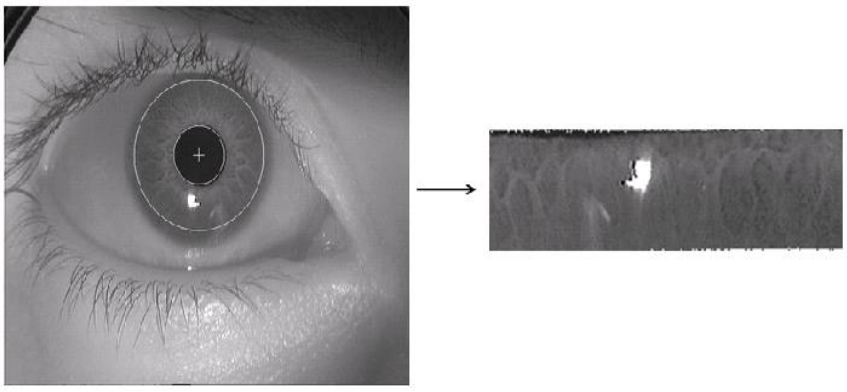

Figura 5: Imagen normalizada.

Modelo rubber sheet

La técnica más empleada en la normalización del iris es la conocida como modelo rubber sheet o lámina de goma. La misma consiste en extraer la textura del iris y deformarla para que quede rectangular. Para ello cada punto del iris es referenciado con un par de coordenadas pseudopolares (r, θ), donde r ϵ [0, 1] representa la distancia del punto al contorno de la pupila y θ ϵ [0. . . 2π) el ángulo respecto al centro del iris. (2)

1.3.3 Codificación

La codificación es la etapa que se encarga de extraer la información del iris que permita diferenciar una persona de otra. El objetivo de esta etapa es generar una matriz binaria conocida como iriscode[3].

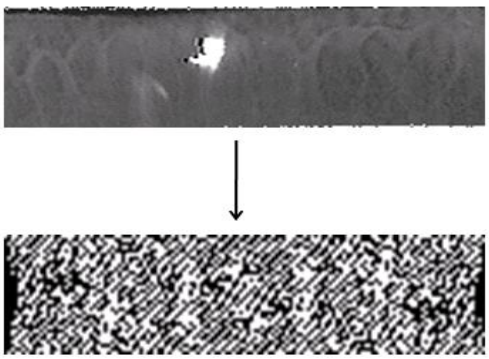

Figura 6: Imagen codificada.

Transformada de Fourier y filtro de Gabor

Una de las técnicas empleadas para resolver la etapa de codificación es aplicar una transformada a la imagen para trabajar en el espacio de frecuencia y luego filtrarla para lograr un acercamiento de todas las intensidades a valores extremos. Para ello primeramente se emplea la Transformada Discreta de Fourier (DFT) para cambiar la representación de los datos del plano espacial al de frecuencia, lo que se logra con la fórmula representada en la siguiente figura:

$$F(u,v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi(ux/M+vy/N)}. \quad (5)$$

Figura 7: Fórmula de la transformada de Fourier.

[3]Muestra de información obtenida del iris humano a partir de transformaciones matemáticas.

Luego para filtrar la imagen se utiliza el filtro de Gabor que constituye un filtro lineal cuya respuesta de impulso es una función sinusoidal multiplicada por una función gaussiana. En el dominio de las frecuencias, el filtro de Gabor G(u, v) se define como la función Gaussiana:

$$G(x,y) = \mathcal{F}^{-1}\{\hat{G}\} = \exp\left(-\pi\left(\frac{x^2}{\alpha^2} + \frac{y^2}{\beta^2}\right)\right)\exp\left(-2\pi i\left(u_0 x + v_0 y\right)\right) \quad (5)$$

Figura 8: Fórmula del filtro de Gabor.

Luego, teniendo en cuenta que la imagen tiene los valores extremos de la intensidad de gris en sus píxeles resaltada, se genera el iriscode asignando 0 ó 1 a cada píxel en dependencia del signo de cada uno de ellos. (5)

DCT (Discrete Cosine Transform)

Otra de las técnicas que se emplean en esta etapa es el uso de la DCT. Esta transformada está bastante relacionada con la DFT, con la diferencia de que es una transformada real, debido a que los vectores base se componen exclusivamente de funciones coseno muestreadas. Además, la DCT minimiza algunos de los problemas que surgen con la aplicación de la DFT a series de datos.

DWT (Discrete Wavelets Transform)

Otro de los métodos utilizados en el mundo para la codificación es la Transformada Discreta de Wavelets, la cual tiene como objetivo separar las componentes de la imagen en baja frecuencia (que le otorgan a la señal la mayor parte de su información o bien, le dan una especie de identidad) y alta frecuencia (que incorporan características más particulares). Este procedimiento se realiza a través de filtros, los cuales se aplican a cada elemento del vector que se transforma, lo que se conoce como DWT-1D. Para aplicar la transformada de wavelets a una imagen se hace necesario aplicar la DWT-1D a cada fila y columna que la conforman y realizar recursivamente este proceso una cantidad *n* de niveles con el objetivo de obtener una información más detallada.

1.3.4 Comparación

La comparación consiste en comparar dos códigos del iris a fin de decidir si ambos fueron generados por el mismo iris o no. Esta etapa pone fin al proceso de identificación por reconocimiento del iris.

Distancia de Hamming

La técnica de mayor aceptación para realizar la comparación en estos sistemas es el cálculo de la distancia de Hamming que consiste en realizar una operación XOR a cada bit de ambos códigos obteniendo cero si los bits son iguales y uno en caso contrario, finalmente, se divide la cantidad de unos obtenidos entre la cantidad total de bits analizados, dando como resultado un valor entre cero y uno que representa el nivel de disimilitud entre ambos códigos y el cual se compara con un umbral de decisión a fin de determinar si ambos códigos pertenecen a una misma persona o no. La distancia de Hamming se define como se muestra en la siguiente figura.

$$\mathrm{HD} = \frac{\|(codeA \bigotimes codeB) \bigcap maskA \bigcap maskB\|}{\|maskA \bigcap maskB\|} \quad (6)$$

Figura 9: Distancia de Hamming.

1.4 Aplicaciones y tendencias actuales de los sistemas de reconocimiento del iris.

1.4.1 Ámbito internacional.

Uno de los pioneros en el desarrollo de sistemas de reconocimiento del iris fue John Daugman, quien desarrolló los procesos de creación de algoritmos de reconocimiento mediante el iris, necesarios para la adquisición de la imagen y la puesta en el mercado de instrumentos necesarios para tal fin. Estos algoritmos se utilizaron para iniciar la comercialización de esta tecnología en conjunto con una primera versión del sistema IrisAccess diseñado y fabricado por LG Electronics, en Corea. Los algoritmos de Daugman son la base de la mayoría de los sistemas de reconocimiento del iris que se introdujeron en el mercado hasta 2006. (6)

Otros sistemas fueron desarrollados por Boashash& Boles, Lim, Monro, Masek, entre otros, pero no alcanzaron el nivel de éxito y expansión comercial de Daugman.

Hay que destacar que existen otras técnicas diferentes de la de Daugman, como la de comparación de histogramas, la de análisis de texturas o la desarrollada por Wildes que propone un enfoque distinto en cuanto a la adquisición de las imágenes y al ajuste del modelo de la imagen de estudio.

Hoy en día, prácticamente todas las compañías que trabajan en el reconocimiento del iris incluyen los algoritmos planteados por Daugman, como lo son LG, Oki, Panasonic, Sagem, IrisGuard, Sarnoff, IRIS, Privium, CHILD Project, CanPass, y Clear.

Entre los principales productos que se pueden destacar de estas empresas se encuentran: el lector de reconocimiento del iris ICAM4000 de LG con un precio alrededor de los 3855 dólares, también está el BM-ET200 de Panasonic que incorpora un nuevo sistema de cámara de reconocimiento del iris que incluye las últimas tecnologías en identificación de personas y con un precio alrededor de los 3190 euros y por último, el InSightDuo de la compañía tecnológica de Silicon Valley, AOptix, que es un sistema biométrico de reconocimiento de personas a través del análisis del rostro y del iris que tiene como fin ser utilizado por las compañías aéreas en el proceso de embarque de pasajeros. Este último ha sido instalado en Reino Unido y Qatar, además de un edificio de "alta seguridad" de Washington y en puertas fabricadas por la firma alemana Kaba, por un precio de US$ 50.000 dólares. (7)

Estos sistemas son muy útiles en aeropuertos, controles de entrada, salida en almacenes, laboratorios, oficinas, hospitales y cárceles. Además, en la telefonía celular ya se encuentra disponible la identificación del iris en lugar de un número PING para acceder al teléfono y también es muy común el empleo de esta técnica en empresas privadas que desean mantener un alto nivel de seguridad en áreas restringidas como la compañía U.S. Government Solutions.

Pero no solo grandes compañías se dedican al estudio y aplicación de esta paradigmática técnica, pues se han realizado diversas investigaciones sobre este tipo de sistemas, entre ellas se destaca:

- *Sistema de reconocimiento de personas mediante su patrón de iris basado en la transformada de Wavelet*: Investigación realizada como proyecto de fin de carrera de Rafael Coomonte Belmonte, de la Escuela Técnica Superior de Ingenieros de Telecomunicación de la Universidad Politécnica de Madrid; en la cual se desarrolla un sistema sobre la base del funcionamiento de la transformada de Wavelet y se emplea el método de la distancia de Hamming para comparar los códigos del iris, utilizando herramientas privadas. El

costo de esta investigación ascendió a un total de 44 646,66 euros con un nivel de acierto de un 99%.

- *Reconocimiento de iris:* Investigación realizada por Marcos González Urbano de la Universidad de Barcelona, y basada en los algoritmos de J. Daugman. En el proyecto se propone una implementación con herramientas libres y con un costo total de 7220 euros.
- *Reconocimiento del iris:* Investigación de Laura Florián Cruz y Fredy Carrazana Athó de la Escuela Académico Profesional de Informática de la Universidad de Trujillo, que emplea los algoritmos de Daugman y herramientas privadas, lo que implica un mayor coste, aunque no se especifica el mismo.
- *Sistema de Reconocimiento de Iris:* Trabajo realizado por Lucas D. Terissi, Lucas Cipollone y Patricio Baldino de la Universidad Nacional de Rosario, Argentina, en el cual se proponen nuevas técnicas como el operador de Canny y la Transformada de Hough para la etapa de segmentación y un análisis multidimensional utilizando wavelets Gabor para la codificación, con los cuales se obtienen resultados satisfactorios. Con el fin de reducir el costo, se emplea una cámara digital convencional, aunque una cámara de este tipo supera la cifra de 150 dólares.
- *Reconocimiento de Iris usando Transformadas Wavelets:* Trabajo realizado por Jesus Calzado Canteño en el cual se emplean filtros de mediana, filtros de Canny, algoritmos de etiquetados y detección de círculos además de funciones Wavelets tipo Haar. El proyecto emplea herramientas privadas lo que implica que su costo total ascienda a 18 515 dólares.
- *Implementación de un Sistema de Identificación de Personas en Tiempo Real por Reconocimiento de Iris:* Investigación de Marcelo Luis Mottalli de la Universidad de Buenos Aires, en la cual se utiliza una versión optimizada del operador integro-diferencial de Daugman apoyado por el algoritmo de Bresenham, incluyendo los filtros de Log-Gabor para la codificación y la distancia de Hamming para la comparación de los iriscodes. Para este trabajo se emplearon herramientas libres como la OpenCV para el procesamiento de imágenes, alcanzando resultados satisfactorios y muy bajas tasas de error en la identificación.

1.4.2 Ámbito nacional.

En nuestro país no se tiene conocimiento de la implementación de un sistema de este tipo que sea funcional, aunque si se registran evidencias de investigaciones relacionadas con el tema, incluyendo las realizadas en la Universidad de las Ciencias Informáticas (UCI).

- *Algoritmo de segmentación para biblioteca de autenticación biométrica a partir del reconocimiento del iris:* Investigación que sienta las bases para la elaboración de una biblioteca de autenticación mediante el análisis de patrones oculares para el proyecto Seguridad del CEDIN, en la que se desarrolla un mecanismo de localización del iris basado en el operador integro-diferencial de John Daugman, empleando herramientas libres.
- *Algoritmo de codificación para biblioteca de autenticación biométrica a partir del reconocimiento de patrones del iris:* Investigación que da solución a la etapa de codificación de la biblioteca de autenticación mediante el análisis de patrones oculares para el proyecto Seguridad del CEDIN, haciendo uso de la Transformada de Fourier y los filtros de Gabor y apoyada sobre herramientas libres.
- *Desarrollo de algoritmo de comparación para biblioteca de autenticación biométrica por reconocimiento de patrones del iris:* En esta investigación se desarrolla un algoritmo de comparación eficiente, para el que se emplea la Distancia de Hamming y el cual fue integrado con los componentes desarrollados en las investigaciones anteriores para obtener una versión inicial de la biblioteca de autenticación de usuarios por reconocimiento de patrones del iris del Seguridad del CEDIN, respetando siempre el uso de herramientas libres.

1.5 Análisis del estado del arte.

Luego de realizado el estudio del estado actual de los sistemas de reconocimiento del iris se ha podido determinar la existencia de un gran número de investigaciones en este campo. En su gran mayoría estos sistemas son desarrollados por empresas privadas debido al alto costo que implican los mismos ya que superan los 2000 euros, valor que aumenta considerablemente una vez realizado el cambio a dólares. Los productos que se estudiaron poseen un buen grado de aceptación y de confiabilidad, el cual oscila entre el 95% y 99%, incluso se encontró uno de estos trabajos que asegura llegar a un 100% de aceptación.

Hay que destacar que la mayoría de estos sistemas se desarrollan con tecnologías privadas lo que implica un costo en licencias de software, aunque también se encontraron investigaciones que emplean herramientas libres a pesar de que los costos siguen siendo elevados como lo es el caso de una que supera los 7000 euros. También se encontró una investigación que utiliza una cámara digital convencional para disminuir los costos, objetivo que no fue logrado en su totalidad pues, como la gran mayoría, empleó herramientas privadas para el desarrollo de la solución.

En nuestro país se registran pocos resultados en cuanto a este tema, aunque se conoce de la versión inicial de una biblioteca desarrollada en la UCI cuyo principal aporte viene dado por el empleo de herramientas libres. Esta biblioteca constituye un conjunto de clases y métodos que realizan las fases fundamentales para el reconocimiento del iris, que al carecer de hardware e interfaz de usuario, representa la herramienta a utilizar para crear un sistema con gran facilidad, incorporando solamente la interfaz de usuario y el dispositivo de captura para obtener un sistema.

1.6 Bibliotecas para el procesamiento de imágenes.

1.6.1 Scimagen.

Scimagen es una biblioteca desarrollada en el Departamento de Ingeniería Eléctrica del Instituto Politécnico Nacional de México (CINVESTAV-IPN). Esta biblioteca fue creada con el objetivo de contar con su código fuente y ponerlo disponible libremente a la comunidad científica, para tener una herramienta propia para la enseñanza de procesamiento de imagen y de programación en el lenguaje C, y tener una plataforma propia para la investigación en técnicas de procesamiento de imagen. (8)

Ventajas

- Herramienta libre.
- Ofrece funciones para la rotación, cálculo de histograma, convolución espacial y filtros.

Desventajas

- Poco reconocida y utilizada en trabajos de este tipo.
- Aún se encuentra en desarrollo por lo que carece de algunas funcionalidades necesarias para el reconocimiento del iris.

1.6.2 ClaseImagenes.dll

ClaseImagenes.dll es una biblioteca desarrollada en Visual Basic .NET. Gracias al ecosistema .NET, la biblioteca puede ser utilizada desde cualquier otro lenguaje del entorno, como es C# o F#.

La biblioteca está compuesta por una gran cantidad de funciones para tratamiento digital de imágenes, desde transformaciones a escala de grises, binarización, hasta otras más complejas como grabar secuencias de transformaciones para poder automatizarlas. Esta biblioteca engloba todo el proceso de creación de una aplicación, incluye multitud de transformaciones, gestión de deshacer/rehacer imágenes, zoom, etc. (9)

Ventajas

- Fácil de utilizar.
- Cuenta con funcionalidades que ayudan al proceso de reconocimiento del iris como el cálculo de histogramas y la detección de contornos.

Desventajas

- Compatible con Microsoft .NET Framework 4.
- Poca documentación.

1.6.3 OpenCV

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999 se ha utilizado en infinidad de aplicaciones, desde sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos.

Existen cuatro grandes componentes en esta biblioteca, los cuales aportan a distinto nivel operaciones y funciones de gran interés. Estas son:

- CXCORE: El núcleo de la librería (biblioteca), en ella se contemplan todas las estructuras básicas para su posterior uso. A su vez facilita en un conjunto de operaciones algebraicas sobre matrices de gran eficiencia temporal. Gracias a esto se puede determinar que este componente es una herramienta algebraica de gran potencial y utilidad.
- Librería CV: Esta biblioteca es el grueso de la aplicación aquí se recoge las principales funcionalidades que OpenCV aporta a la visión computacional. Procesamiento de imágenes, análisis estructural, seguimiento de objetos en visión, reconocimiento de patrones y calibración de cámaras para reconstrucciones en 3D.

- Máquina de aprendizaje: Este componente ofrece una herramienta de aprendizaje para la detección y clasificación de características en imágenes, agrupamiento de datos y regresión de datos. Basado en un algoritmo Boost de rápido aprendizaje pero con la necesidad de un gran número de ejemplos.
- Aplicación de alto nivel: Ofrece funciones de alto nivel y una interfaz gráfica para la percepción de las distintas funcionalidades contempladas en el resto de los componentes, al igual que visualización de imágenes. (3)

Desde su lanzamiento, OpenCV ha visto más de 500.000 descargas de código y ha atraído a más de 5.000 miembros registrados a su grupo de usuarios.

Ventajas

- Publicada bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.
- OpenCV es multiplataforma, existiendo versiones para Linux, Mac OS X y Windows.
- Contiene más de 500 funciones que abarcan una gran gama de áreas en el proceso de visión, como reconocimiento de objetos, calibración de cámaras, visión estéreo y visión robótica.

Desventajas

- En el caso del seguimiento de objetos, no ofrece un producto completo, tan solo algunas piezas que sirven como base para montar sobre ellas un producto final.
- Otro de los inconvenientes que tiene es la necesidad de utilizar la librería IPL para tener acceso a funciones de bajo nivel.

Capítulo 2: Solución propuesta.

2.1 Introducción al capítulo

En este capítulo se hará una descripción de la propuesta de solución para el problema definido en la investigación. También se desarrollará el proceso de modelación del sistema, ofreciendo una descripción de las características del mismo. Para ello se especifican los requerimientos funcionales y no funcionales que debe tener la aplicación, se identifican los actores, se describen los principales casos de usos y se confeccionan varios diagramas para una mejor comprensión del sistema.

2.2 Propuesta de herramientas y tecnologías a emplear.

Lenguaje de modelado: UML.

Se propone utilizar el Lenguaje Unificado de Modelado (UML) por ser el lenguaje de modelado de sistemas de software que se emplea en el módulo de Seguridad del centro y porque ofrece varios beneficios entre ellos:

- Mejores tiempos totales de desarrollo (de 50 % o más).
- Modelar sistemas (y no sólo de software) utilizando conceptos orientados a objetos.
- Establecer conceptos y artefactos ejecutables.
- Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimización de costos. (10)

Metodología de desarrollo de software: RUP.

RUP es la metodología empleada por el módulo de Seguridad, motivo por el que se propone utilizar esta metodología considerando que brinda facilidades como:

- Permite tener claro y accesible el proceso de desarrollo que se sigue.
- Ayuda a minimizar el riesgo.
- Garantiza la predictibilidad de los resultados.
- Ayuda a entregar software de calidad superior a tiempo. (11)

CASE: Visual Paradigm.

Como herramienta CASE se propone Visual Paradigm pues es multiplataforma y modela todos los flujos de trabajo de RUP además de ser la que se utiliza en el centro.

Lenguaje de programación: C++.

En el centro, para el desarrollo de las aplicaciones se emplea C++ como lenguaje de programación. Por este motivo y otras de sus características como su robustez y eficiencia se propone este lenguaje para el desarrollo del módulo.

Sistema operativo: GNU/Linux.

A partir de la política de software libre vigente en el centro y teniendo en consideración la propiedad de ser multitarea, multiusuario, multiplataforma y multiprocesador se propone GNU/Linux con su distribución de Ubuntu 12.04 como sistema operativo.

IDE: Eclipse.

Al igual que las herramientas anteriores Eclipse es uno de los entornos de desarrollo más utilizados en el centro. Este IDE brindara facilidades en la vinculación de bibliotecas y en la configuración del compilador y cuenta con un buen completamiento de código. Por estas razones se propone Eclipse como entorno de desarrollo para la solución del problema.

Base de datos: CASIA-1

En el mundo existen varias bases de datos cuyo objetivo es proveer un conjunto de imágenes que permitan probar los sistemas de reconocimiento del iris. Con el objetivo de poder validar la solución y teniendo en cuenta que no se cuenta con el hardware necesario se decide utilizar la base de datos CASIA-1. Esta es una base de datos libre que contiene un total de 756 imágenes que se encuentran distribuidas en 108 carpetas de 108 individuos diferentes. Cada carpeta contiene 2 subcarpetas con 3 o 4 imágenes por cada una para un total de 7 imágenes por carpeta. Estas imágenes tienen una resolución de 320x280. También se tuvieron en cuenta otras bases de datos como la MMU y la del Departamento de Ciencias de la Computación de Palacky University. De ellas se puede decir que sus imágenes presentan mayor afectación provocada por la luz. A pesar de que las imágenes de la MMU tienen una resolución similar a las de la CASIA estas últimas brindan una mejor definición de la región de interés, o sea, el iris. Por otro lado, las imágenes de la base de datos de Palacky

University muestran una versión más “extrema” en cuanto a precisión del área capturada, ya que no hay ruido de párpados, ni pestañas, y de hecho, enmascara con negro gran parte de la información despreciable, lo que se aleja un poco de la realidad. Estos fueron argumentos que ayudaron a decidir la selección de la base de datos, incluyendo que la mayoría de los trabajos científicos relacionados con algoritmos o técnicas para detección y extracción digital del iris, utilizan la base de datos CASIA-1 como banco de pruebas para exponer sus resultados. (4)

Biblioteca: OpenCV

De las bibliotecas descritas en el capítulo anterior se propone emplear OpenCV pues tiene gran aceptación mundialmente en cuanto al reconocimiento del iris. Sus desventajas no se consideran realmente significantes pues no tienen mucha relación con el tema que se estudia, por lo que casi todo lo que se tiene con el empleo de ella son elementos positivos. Esta biblioteca ofrece varios tipos de datos que la hacen ser más simple y uniforme y que ayudan al trabajo con las imágenes lo que es muy útil para este trabajo, como es el caso de los tipos de datos IplImage (imagen de IPL) y CvMat (matriz), y otros auxiliares como CvPoint (2d punto) y CvSize (anchura y altura). Estas estructuras garantizan de forma general la implementación de los algoritmos propuestos para la solución de cada una de las etapas del proceso de reconocimiento del iris. La OpenCV también cuenta con un conjunto de funciones que permiten el manejo de las imágenes como son la cvLoadImage, cvSaveImage y cvShowImage.

2.3 Propuesta de algoritmos para la segmentación de la imagen.

Segmentación de la pupila.

La segmentación de la pupila tiene como objetivo determinar el centro del iris y la pupila y el radio de la pupila. En este trabajo se propone el algoritmo Cruz descrito en el capítulo anterior. Primeramente, antes de aplicar dicho algoritmo y con el objetivo de obtener mejores resultados se propone aplicar filtros de paso bajo y binarizar[4] la imagen teniendo en cuenta que en la región de la pupila los píxeles tienen valores de intensidades de gris muy similares a cero,

[4]Transformar la intensidad de los píxeles en 0 o 255 según un umbral de decisión definido.

este procedimiento se considera imprescindible para encontrar una variación realmente alta como se muestra en la Figura 10.

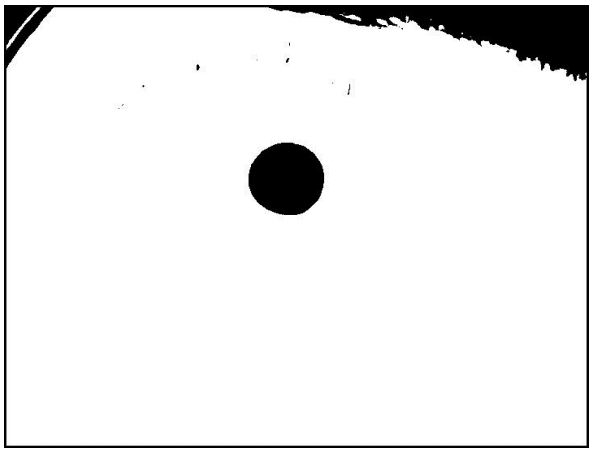

Figura 10: Imagen binarizada.

Una vez determinadas las trazas calculadas por el algoritmo Cruz para cada punto de la imagen se define como centro de la pupila el punto donde las distancias de las trazas tengan los valores más similares. Luego para determinar el radio de la pupila se propone un cambio en la concepción del algoritmo Cruz, pues este selecciona el valor de la menor de las trazas con el objetivo de garantizar que se seleccionen todos los píxeles del iris y para este trabajo se propone seleccionar el valor de la mayor de dichas trazas evitando de esta manera que exista interferencia de píxeles correspondientes a la pupila.

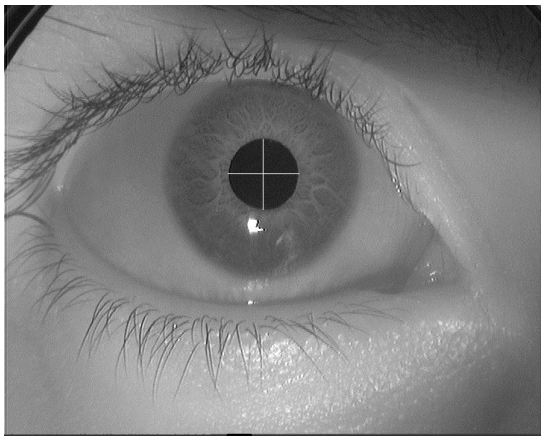

Figura 11: Representación de las trazas para determinar la pupila.

Segmentación del iris.

Una vez segmentada la pupila se procede a segmentar el iris, para ello en este trabajo se propone el empleo del operador integro-diferencial de John Daugman descrito en el Capítulo 1. Sin embargo, se decide utilizar el operador

solo en una porción de la circunferencia, el lado izquierdo y derecho, evitando de esta manera interferencia provocada por los párpados como se aprecia en la siguiente figura.

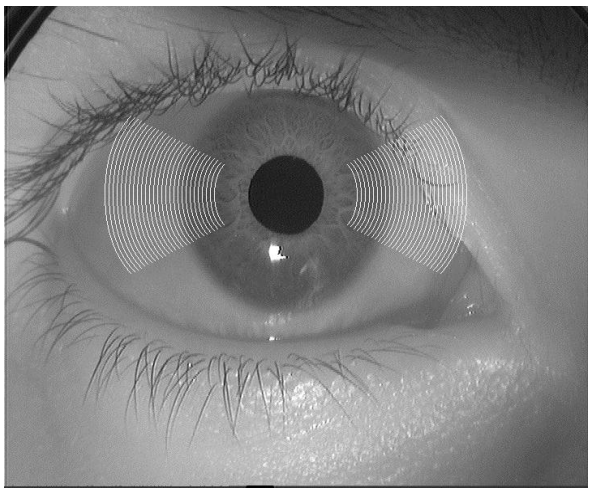

Figura 12:Región seleccionada para emplear el operador íntegro-diferencial.

Segmentación de párpados.

Para la segmentación de los párpados también se sigue el principio del operador integro-diferencial de John Daugman, sin embargo, para este trabajo se define un cambio pues no se describe una circunferencia sino una parábola, teniendo en cuenta su fórmula general.

$$y = p * (x - x_0)^2 + y_0 \quad (12)$$

Donde (x_0,y_0) representa el centro de la parábola y **p** es su amplitud. En este trabajo se proponen los siguientes valores para estas variables:

- $X_0 \in [X_C$ **- pupilRatio;** X_c **+ pupilRatio]**, siendo X_C la **x** del centro del iris y la pupila y **pupilRatio** el radio de la pupila.
- $Y_0 \in [Y_C$ **+pupilRatio; alto de la imagen]**, siendo Y_C la **y** del centro del iris y la pupila.
- **p** toma un valor de **-0.02.**

Es importante señalar que debido a que la mayoría de las imágenes presentan ruido provocado por la presencia de párpados y pestañas en la región del iris comprendida entre los ángulos de π/4 y 3π/4 se decide desechar dicha región por lo que no se detecta el párpado superior.

Finalmente, con el empleo de los algoritmos propuesto se obtiene una imagen segmentada como la que se representa en la figura siguiente.

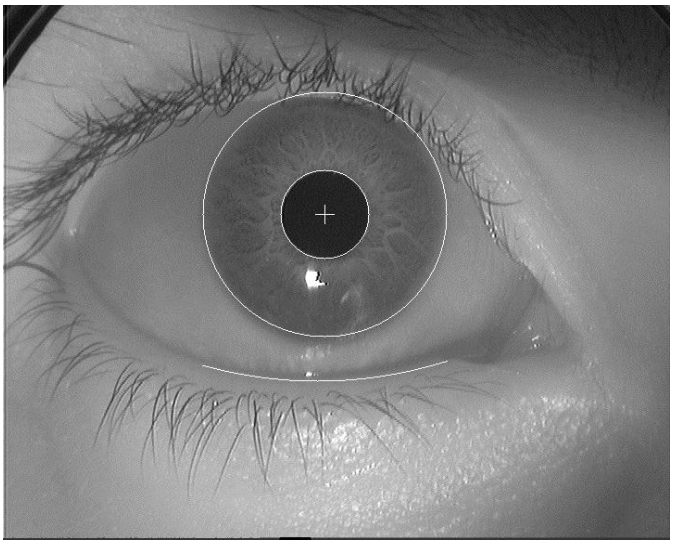

Figura 13: Imagen segmentada por el módulo libIris.

2.4 Propuesta de algoritmo para la normalización del iris.

La normalización consiste en convertir la textura del iris en una imagen rectangular de dimensiones fijas. En este trabajo, para la normalización del iris, se propone la técnica conocida como Modelo Rubber Sheet descrita en el capítulo anterior. Para desarrollar dicha técnica se propone tomar 40 circunferencias virtuales concéntricas al iris a partir de un radio inicial que coincide con el radio de la pupila más un margen de seguridad de 2 píxeles. Con el objetivo de asegurar que toda la zona de la pupila quede en la parte interior de la circunferencia y no interfiera en ningún caso en los datos, cada circunferencia incrementa un valor Δr respecto a la anterior, que vendrá definido por la diferencia existente entre el radio del iris y el radio inicial. De este modo para poder tomar 40 circunferencias virtuales y admitiendo un margen de seguridad sobre el radio del iris se propone un incremento radial de: Δr= (Radio del iris-Radio inicial)/40.

Del mismo modo, se propone tomar 240 ángulos distintos, comenzando en el ángulo φ = 210 inicial, y los ángulos elegidos en cada caso serán tales que Δφ=1; sin embargo, se pretende reconstruir solo el 75% de la información del iris eliminando los ángulos correspondientes al intervalo [π/4; 3π/4] para lograr de esta manera un ruido casi nulo por intromisión de párpados y pestañas como se puede apreciar en la siguiente figura.

Figura 14:Imagen normalizada por el módulo libIris.

Luego, con el objetivo de obtener mejores resultados en las etapas posteriores y siguiendo el procedimiento descrito para normalizar el iris, se propone generar una máscara de ruido, que marca aquellos puntos que fueron afectados por ruido provocado por el párpado inferior o por la luz de la cámara como se muestra en la siguiente figura.

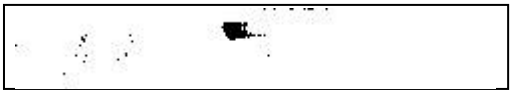

Figura 15:Máscara de ruido generada por el módulo libIris.

2.5 Propuesta de algoritmo para la codificación del iris.

La etapa de codificación consiste en generar el iriscode asociado a la imagen del iris normalizado. Para solucionar esta etapa se parte de un análisis de los posibles algoritmos a emplear, teniendo en cuenta los que fueron descritos en el Capítulo 1. Desde un primer momento se desecha la opción de emplear la DFT, ya que esta fue la solución empleada en la versión inicial de la biblioteca con que cuenta el proyecto Seguridad del CEDIN y con la que no se obtuvo un buen por ciento de aceptación, después de realizadas las pruebas.

Luego se decide utilizar la funcionalidad de la biblioteca OpenCV que calcula la DCT, pero los resultados obtenidos fueron similares a los de la versión inicial de la biblioteca del proyecto por lo que también se desecha esta opción. Posteriormente se implementa un algoritmo que ejecute el proceso de aplicar la DWT a una imagen, descrito en el capítulo anterior, pero los resultados obtenidos no fueron satisfactorios debido a su elevado costo computacional, por lo que tampoco se considera esta opción válida para solucionar la etapa de codificación en este trabajo.

Por último, se analiza un algoritmo utilizado en (4) que divide la imagen normalizada en cuadrantes, donde para cada uno se calcula el promedio de las intensidades de gris de los píxeles que lo integran, y genera el iriscode a partir de la relación de cambio que hay píxel a píxel en la textura normalizada del iris. Para este trabajo se propone este algoritmo, pero con el objetivo de obtener mejores resultados, solo se utilizan los píxeles que no representen ruido en la máscara que se genera en la etapa anterior. Luego se genera el iriscode al guardar en un arreglo el equivalente en binario de los promedios calculados, y no con la relación de cambio que plantea el algoritmo original pues esta nueva propuesta ofrece mejores resultados.

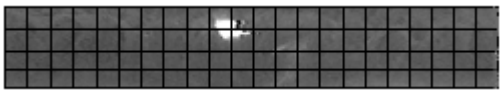

Figura 16: Imagen normalizada dividida en cuadrantes.

Se propone esta nueva variante del último algoritmo analizado como solución para la etapa de codificación, teniendo en cuenta que es un algoritmo sencillo de implementar y que requiere de poco costo computacional, además de brindar mejores resultados respecto a la versión inicial de la biblioteca.

2.6 Propuesta de algoritmo de comparación.

Finalmente, el proceso concluye con la etapa de comparación que tiene como objetivo determinar si el usuario es quien dice ser o no. Para esta etapa se propone el cálculo de la Distancia de Hamming cuyo procedimiento se describe en el Capítulo 1 y se representa en la siguiente figura.

Code A	0	0	1	1
	=	≠	=	≠
Code B	0	1	1	0
	↓	↓	↓	↓
	0	1	0	1

HD = 2/4 = 50%

Figura 17: Cálculo de la distancia de Hamming.

Teniendo en cuenta la poca probabilidad de que dos códigos del iris sean iguales aunque provengan de la misma persona, ya que pueden ocurrir variaciones en la etapa de captura, tanto en la intensidad de la luz ambiental como provocadas por cambios de la posición del ojo frente a la cámara, se propone realizar una serie de rotaciones a la izquierda y derecha del código procedente del usuario que ayuden a simular estas posibles variaciones.

Luego se debe realizar, para cada una de estas rotaciones, el cálculo de la distancia de Hamming y almacenar los resultados obtenidos para seleccionar el menor de ellos y compararlo con un umbral de decisión. Este proceso se representa en la Figura 18.

Para determinar el umbral de decisión se sigue el procedimiento descrito en (6) y se obtiene que el menor valor de Hamming alcanzado entre códigos diferentes es de 0.469, por lo que se propone dicho valor como umbral de decisión para este trabajo.

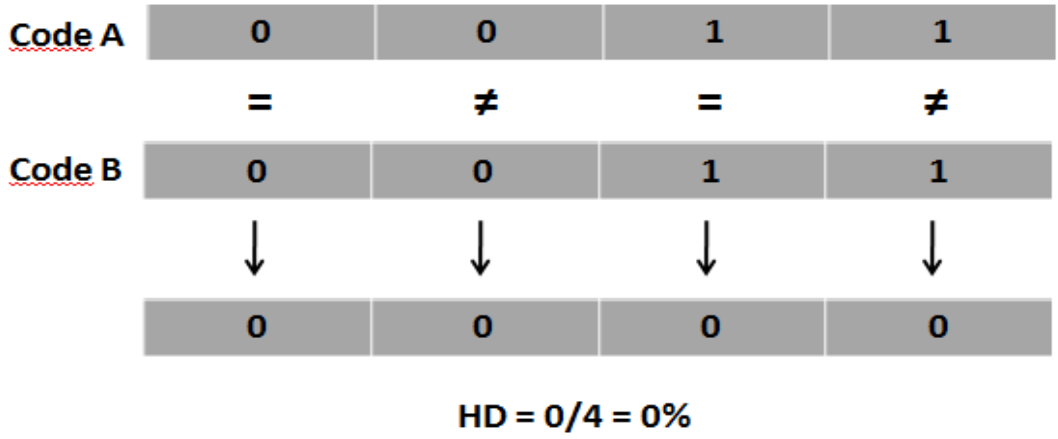

Figura 18: Cálculo de la distancia de Hamming rotando el código B una unidad hacia la derecha.

Luego de realizar una propuesta de solución para cada una de las etapas que componen el proceso de reconocimiento del iris, se pone en práctica dicho proceso a través de una aplicación que tiene un conjunto de requisitos a los que se les da cumplimiento con la realización de casos de uso, y que se representa mediante un conjunto de clases de las que se tiene una vista proporcionada por una arquitectura y siguiendo patrones de diseño. A continuación se caracteriza dicha aplicación.

2.7 Especificación de los requisitos de software

2.7.1 Requisitos funcionales del sistema.

- RF1. Delimitar la región de la pupila en la imagen de un ojo.
- RF2. Delimitar la región del iris en la imagen de un ojo.
- RF3. Delimitar los párpados en la imagen de un ojo.
- RF4. Convertir el anillo del iris en una imagen rectangular.
- RF5. Crear el iriscode asociado a dicha imagen.
- RF6. Comparar iriscodes.

2.7.2 Requisitos no funcionales del sistema.

- Requerimientos de Software.
  - Emplear GNU/Linux como sistema operativo.
- Requerimientos de Hardware.
  - Se recomienda usar procesadores INTEL ya que la biblioteca OpenCV está optimizada para este tipo de procesador.
  - Memoria RAM de 1GB o superior.
  - Capacidad de almacenamiento de 1GB.
- Restricciones de Diseño e Implementación.
  - Emplear un conjunto de patrones que permita la reutilización del código así como facilidad en la actualización del mismo.
  - El código debe cumplir con los estándares de codificación establecidos por el CEDIN.
  - Utilizar la biblioteca OpenCV pues esta garantiza la implementación eficiente de algoritmos necesarios para la construcción de este componente.
- Requerimientos de Soporte.
  - Ofrecer servicios de mantenimiento y actualización.
- Requerimientos de Usabilidad.
  - El módulo deberá ser usado por cualquier persona que posea conocimientos básicos en el manejo de la computadora, programación y el sistema operativo GNU/Linux.
- Requerimiento asociado al Licenciamiento.
  - Se debe garantizar que el sistema se desarrolle bajo los principios del software libre.

2.8 Modelación del sistema.

2.8.1 Actores.

Un actor representa una persona o un sistema externo que guarda una relación con el sistema y que le demanda una funcionalidad (11). En este sistema se define un actor ***Aplicación***.

2.8.2 Casos de uso.

Un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema (11). En este sistema se definen los

casos de uso ***Segmentar imagen, Normalizar imagen, Codificar imagen*** y ***Comparar iriscodes.***

2.8.3 Diagrama de casos de uso del sistema.

El diagrama de casos de uso del sistema está determinado por un actor encargado de solicitar la realización de los casos de uso correspondientes para dar cumplimiento a los requisitos funcionales del sistema. Estamos en presencia de un proceso que inicia con el caso de uso ***Segmentar imagen*** en el cual se resuelven los tres primeros requisitos del sistema. Una vez realizado este caso de uso y a partir del resultado del mismo se inicia el caso de uso ***Normalizar imagen*** que da cumplimiento al siguiente requisito, luego se realiza el caso de uso ***Codificar imagen*** resolviendo el próximo requisito y finalmente se inicia el caso de uso ***Comparar iriscodes*** poniendo fin al proceso y dando cumplimiento a todas las funcionalidades del sistema.

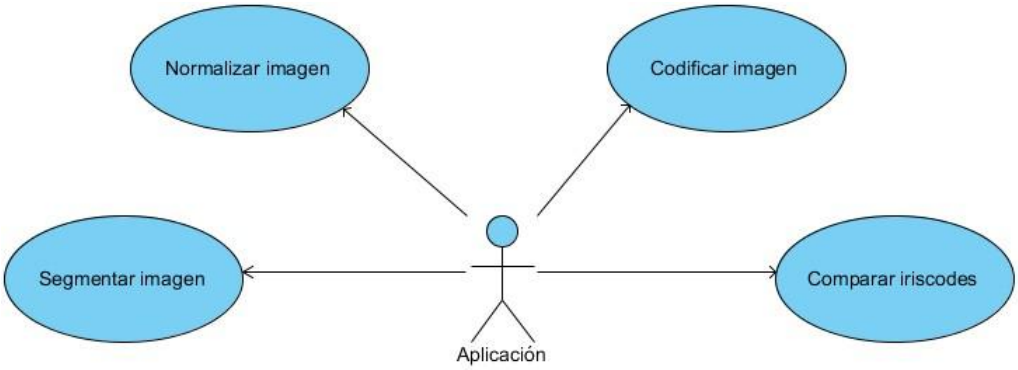

Figura 19: Diagrama de casos de uso.

2.8.4 Descripción de casos de uso del sistema.

Caso de Uso:	
CU1	Segmentar imagen
Propósito:	Delimitar las regiones de la pupila, el iris y los párpados superior e inferior, que posibiliten el comienzo de la etapa de normalización.
Actores:	Aplicación.

Resumen:	La segmentación de la imagen consiste en delimitar las regiones de la pupila, el iris y los párpados superior e inferior a partir de la imagen de un ojo.
Referencias:	R1, R2, R3
Precondiciones:	Debe realizarse la captura de la imagen que se analizará.
Complejidad:	Alta.
Prioridad:	Crítico.
Poscondiciones:	La imagen queda segmentada luego de realizarse este caso de uso.

Flujo Normal de Eventos	
Aplicación	**Módulo**
1. Solicita la captura de la imagen de un ojo del usuario.	1.1 Brinda la imagen de un ojo.
2. Solicita la segmentación de la pupila en la imagen capturada.	2.1 Delimita la región de la pupila en la imagen.
3. Solicita la segmentación del iris en la imagen capturada.	3.1 Delimita la región del iris en la imagen.
4. Solicita la segmentación de los párpados superior e inferior en la imagen capturada.	4.1 Delimita la región de los párpados superior e inferior en la imagen.
	4.2 Notifica la segmentación de la imagen.
Flujos Alternos	
1.1a Existe problema de conexión a la base de datos.	

Aplicación	Módulo
	1.1a Notifica error de captura de la imagen.

Tabla 1: Descripción del caso de uso Segmentar imagen.

Caso de Uso:	
CU2	Normalizar imagen
Propósito:	Obtener una imagen rectangular de dimensiones fijas con la textura del iris, que posibilite el comienzo de la etapa de codificación.
Actores:	Aplicación.
Resumen:	La normalización de la imagen constituye la creación de una imagen rectangular de dimensiones fijas a partir de una imagen segmentada.
Referencias:	R4
Precondiciones:	Debe haberse realizado el caso de uso Segmentar imagen.
Complejidad:	Alta.
Prioridad:	Crítico.
Poscondiciones:	La imagen queda normalizada luego de realizarse este caso de uso.

Flujo Normal de Eventos	
Aplicación	**Módulo**
1. Solicita la conversión del anillo del iris a una imagen rectangular.	1.1 Convierte la región del iris en una imagen rectangular.

	1.2 Genera una máscara de ruido.
	1.3 Notifica la normalización del iris.
Flujos Alternos	

Tabla 2: Descripción del caso de uso Normalizar imagen.

Caso de Uso:	
CU3	Codificar imagen
Propósito:	Obtener un iriscode a partir de los patrones distintivos del iris que posibilite el comienzo de la etapa de comparación.
Actores:	Aplicación.
Resumen:	La codificación de la imagen constituye la generación de un código binario denominado iriscode a partir de una imagen normalizada.
Referencias:	R5
Precondiciones:	Debe haberse realizado el caso de uso Normalizar imagen.
Complejidad:	Alta.
Prioridad:	Crítico.
Poscondiciones:	La imagen queda codificada luego de realizarse este caso de uso.
Flujo Normal de Eventos	
Aplicación	**Módulo**
1. Solicita el iriscode asociado a la	1.1 Genera el iriscode.

imagen.	
	1.2 Notifica la codificación del iris.
Flujos Alternos	

Tabla 3: Descripción del caso de uso Codificar imagen.

Caso de Uso:	
CU4	Comparar iriscodes.
Propósito:	Comparar dos códigos del iris a fin de determinar si el usuario es quien dice ser.
Actores:	Aplicación.
Resumen:	La comparación de dos códigos del iris consiste en calcular la menor distancia de Hamming existente entre ellos a partir de poder obtener el nivel de disimilitud existente entre ellos.
Referencias:	R6
Precondiciones:	Debe haberse realizado el caso de uso Codificar imagen.
Complejidad:	Alta.
Prioridad:	Crítico.
Poscondiciones:	Si el usuario pertenece al sistema se le debe permitir la entrada. De lo contrario el sistema debe avisarle de que no ha logrado identificarlo.
Flujo Normal de Eventos	
Aplicación	**Módulo**

1. Solicita el código del iris a comparar.	1.1 Brinda el código del iris.
2. Solicita el cálculo de la distancia de Hamming entre los códigos del iris.	2.1 Calcula la menor distancia de Hamming entre los códigos del iris.
	2.2 Verifica que la distancia de Hamming sea menor que el umbral de decisión.
	2.3 Notifica autenticación satisfactoria del usuario.
Flujos Alternos	
2.2a Distancia de Hamming mayor o igual que el umbral de decisión	
Aplicación	**Módulo**
	2.2a Notifica autenticación fallida del usuario.

Tabla 4: Descripción del caso de uso Comparar iriscodes.

2.9 Modelo de diseño.

2.9.1 Diagramas de interacción.

Los diagramas de interacción ilustran el modo en el que los objetos interaccionan por medio de mensajes. Existe un tipo de diagrama de interacción más especializado conocido como Diagrama de Secuencia que se encarga de visualizar la vida de los mensajes y sus efectos. (2) A continuación se representan los diagramas de secuencia correspondientes para cada caso de uso.

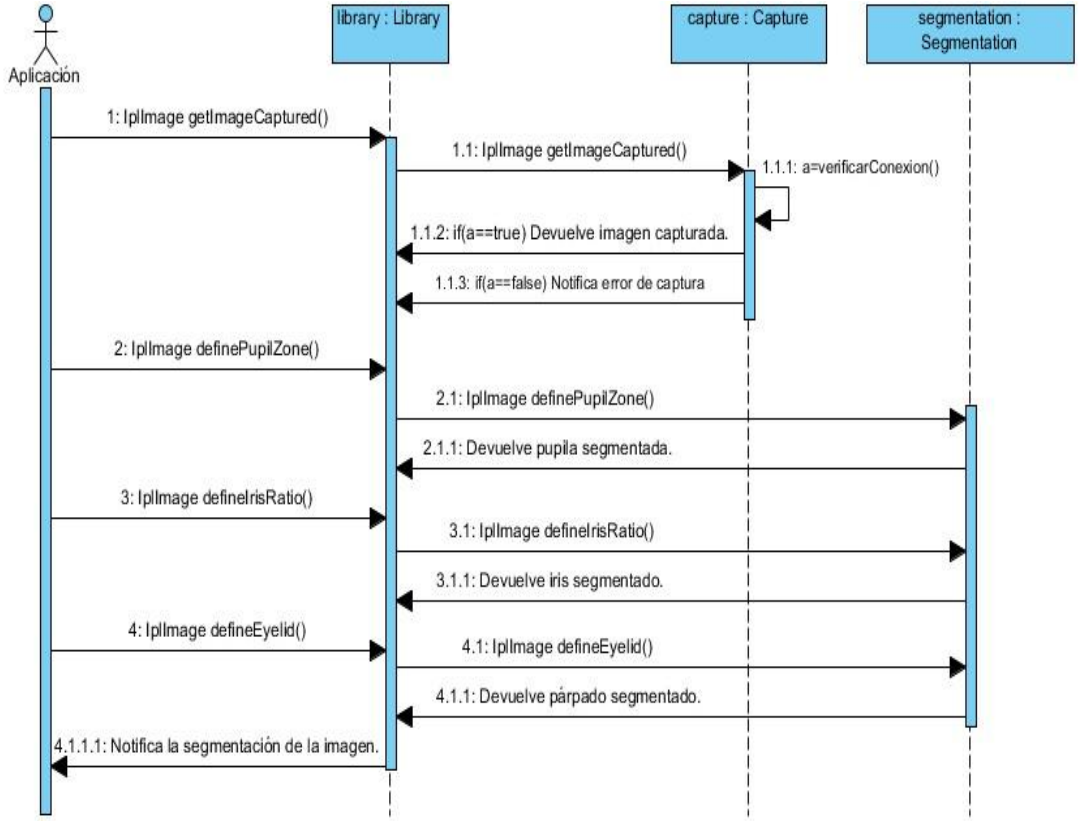

Figura 20: Diagrama de secuencia (Segmentar imagen).

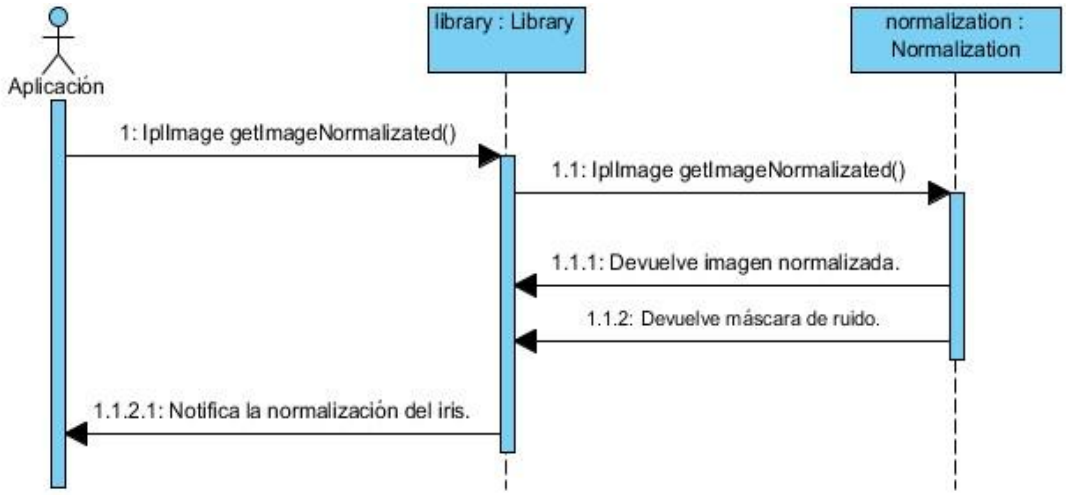

Figura 21: Diagrama de secuencia (Normalizar imagen).

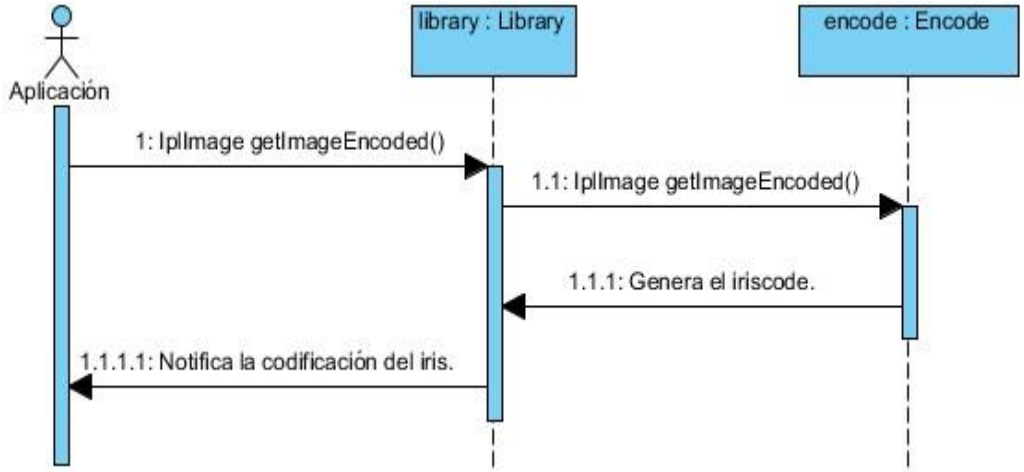

Figura 22: Diagrama de secuencia (Codificar imagen).

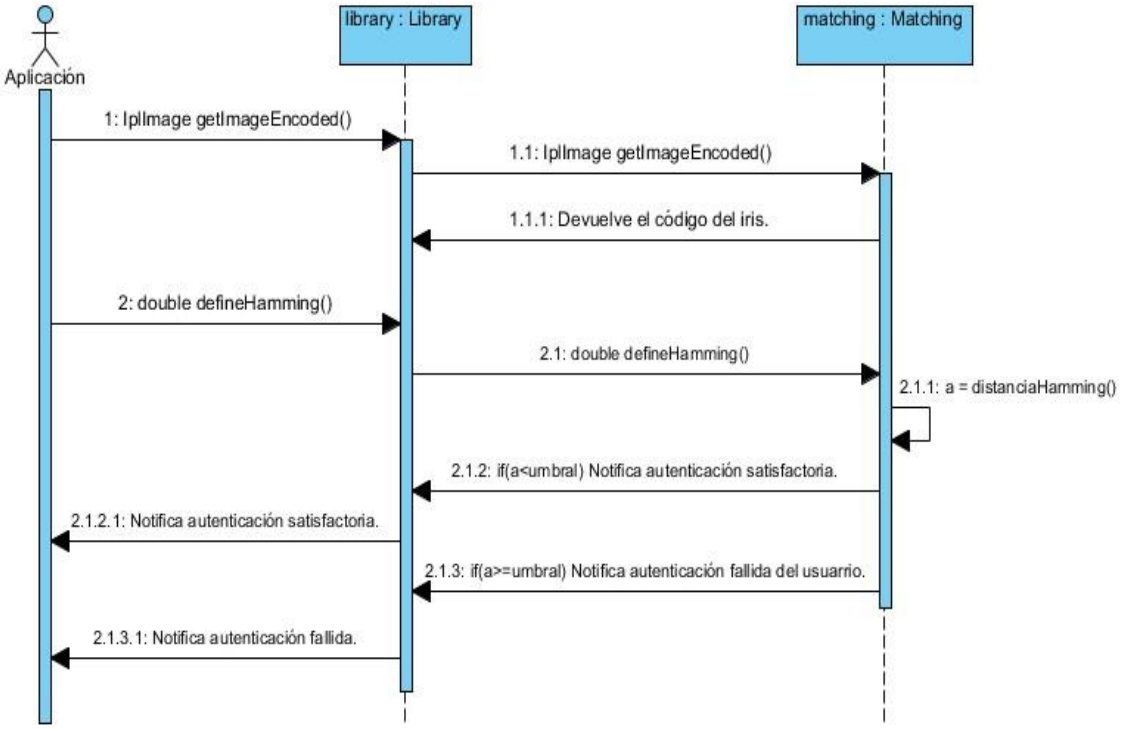

Figura 23:Diagrama de secuencia (Comparar iriscodes).

2.9.2 Diagrama de clases del diseño.

A continuación se muestra el diagrama de clases del diseño en el que se representan las clases con que cuenta el sistema, así como sus atributos y funcionalidades y la relación existente entre ellas. Para obtener información detallada de cada una de ellas ver **Anexo 2**.

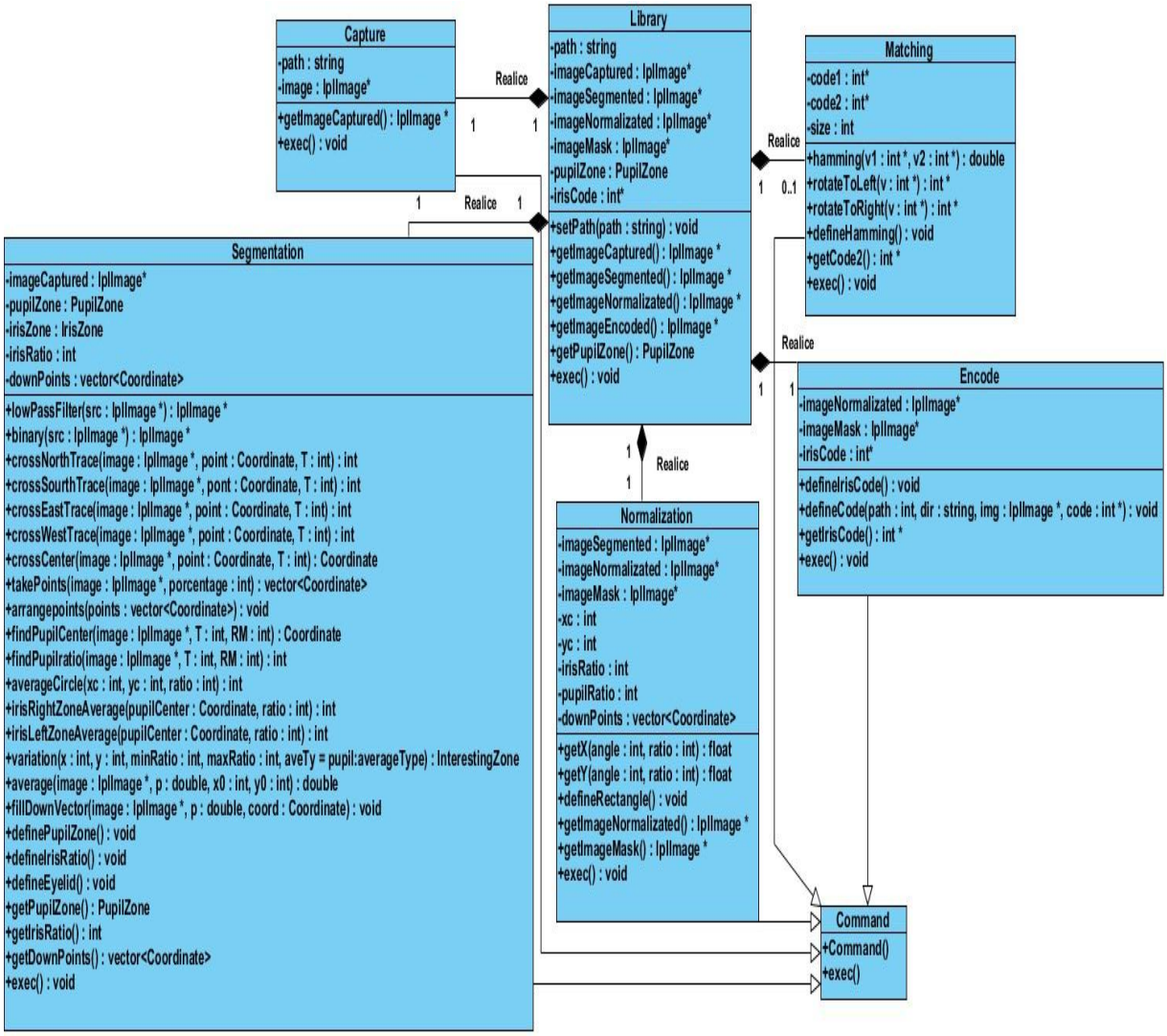

Figura 24: Diagrama de clases.

2.9.3 Arquitectura del sistema.

La Arquitectura de software es una vista del sistema que incluye los componentes principales del mismo, la conducta de esos componentes según se la percibe desde el resto del sistema y las formas en que los componentes interactúan y se coordinan para alcanzar la misión del sistema. (13)

En este trabajo se aplica una arquitectura en capas. Presenta una capa lógica donde se encuentran las clases *Segmentation, Normalization y Encode*, donde se implementan los algoritmos para solucionar las etapas que resuelven el proceso de identificación de usuarios; y una capa de acceso a datos representada por la clase *Matching* donde se accede a los

iriscodes con los que se va a comparar el iriscode generado por el módulo a fin de determinar si el usuario es quien dice ser o no.

2.8.5 Patrones de diseño.

Un patrón de diseño es un mecanismo de la ingeniería de software que brinda robustez y flexibilidad a un conjunto de clases. Constituye una solución predeterminada a un problema de diseño común para las aplicaciones de software. (5)

En este trabajo se emplean dos patrones. El *Facade* o *Fachada*, el cual pertenece al grupo de los patrones estructurales y provee una interfaz que posibilita que cualquier aplicación use las funciones que dicha interfaz es capaz de brindarle. La clase *Library*, actúa como una interfaz y provee las funcionalidades de las clases *Capture*, *Segmentation, Normalization, Encode* y *Matching*. También se emplea el *Command* o *Comando*, patrón de comportamiento que encapsula una función específica, permitiendo que esta función se ejecute en cualquier comando sin que importe el comportamiento del mismo. La clase *Command* define una función *execute* que funciona como una interfaz para la ejecución de la operación permitiendo que cada una de sus clases derivadas actúe como un comando.

Capítulo 3: Implementación y pruebas.

3.1 Introducción al capítulo.

En el presente capítulo se realizará una representación de la implementación del sistema a partir de diagramas como el de despliegue y el de componentes, además de una descripción de las pruebas realizadas al software para validar la solución propuesta en el capítulo anterior.

3.2 Implementación.

El flujo de trabajo Implementación describe cómo los elementos del Modelo de Diseño se implementan en términos de componentes y cómo estos se organizan de acuerdo a los nodos específicos en el Modelo de Despliegue. (11)

3.2.1 Diagrama de despliegue.

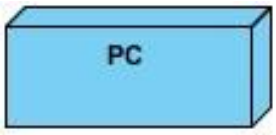

Figura 25:Diagrama de despliegue.

3.2.2 Diagrama de componentes.

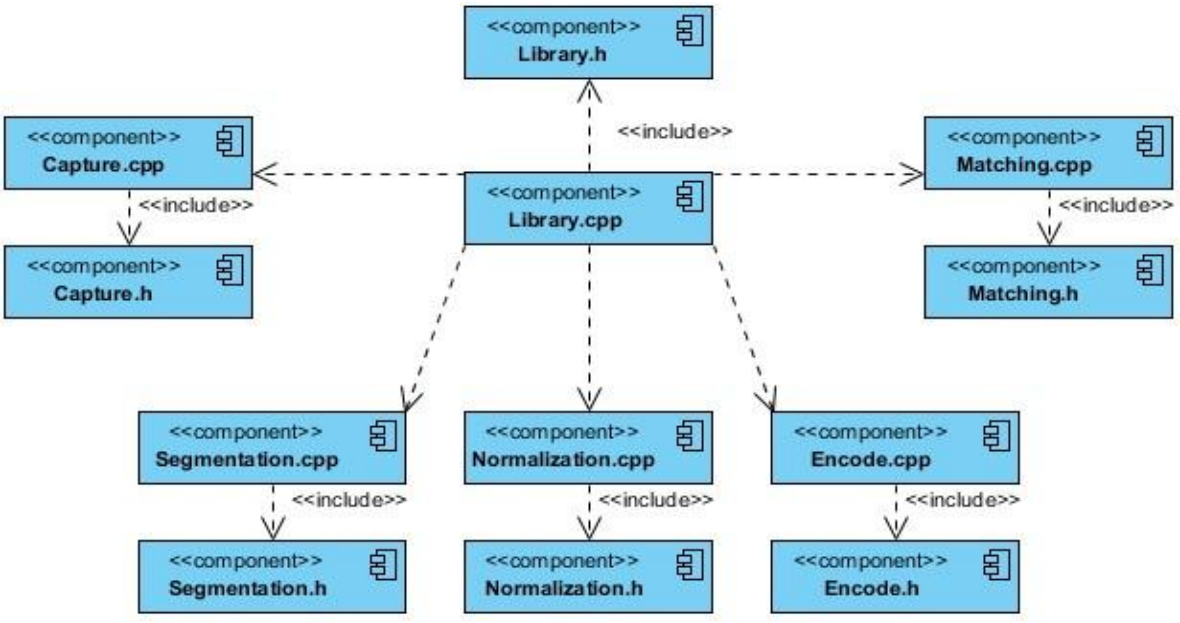

Figura 26: Diagrama de componentes.

3.2.3 Estándar de codificación.

El estándar de codificación que se emplea es el propuesto por el centro CEDIN.

3.3 Modelo de prueba.

3.3.1 Descripción de los casos de prueba.

La realización de pruebas se puede definir como una actividad en la que un sistema o componente es ejecutado bajo ciertas condiciones previamente especificadas y donde los resultados son observados y registrados, para finalmente realizar una evaluación de los aspectos del sistema.

Existen diferentes niveles de pruebas, entre ellos las pruebas de sistema, las cuales tienen como objetivo verificar el software funcionando en conjunto. Los casos de uso sirven como fundamento para desarrollar casos de pruebas de sistema. Las pruebas de caso de uso son una técnica que ayuda a identificar casos de prueba que ejerciten el sistema entero transición a transición desde el principio al final. (14)

A continuación se detalla el proceso de prueba definido para cada caso de uso con el fin de probar el correcto funcionamiento del sistema bajo las diferentes condiciones a las que puede someterse.

Nombre del caso de uso:	Segmentar imagen.	
Entrada	**Resultados esperados**	**Resultados obtenidos**
Caso de Prueba #1		
Se introduce una imagen cuyo radio del iris es menor que 40 píxeles.	Acción fallida. El sistema muestra el mensaje “Error: Capture failure”.	Prueba exitosa. Se obtienen los resultados esperados.
Caso de Prueba #2		
Se solicita segmentar una imagen capturada.	El sistema retorna una imagen con la pupila, el iris y el párpado inferior segmentados.	Prueba exitosa. Se obtienen los resultados esperados.

Tabla 5: Casos de prueba para caso de uso Segmentar imagen.

Nombre del caso de uso:	Normalizar imagen.	
Entrada	**Resultados esperados**	**Resultados obtenidos**
Caso de Prueba #1		
Se introduce una incorrecta imagen de entrada por mala segmentación de la imagen.	Acción fallida. El sistema muestra el mensaje “Error: Segmentation failure”.	Prueba exitosa. Se obtienen los resultados esperados.
Caso de Prueba #2		
Se solicita normalizar una imagen segmentada cuyo iris contiene insuficiente información para ser procesado.	Acción fallida. El sistema muestra el mensaje “Error: Capture failure or Segmentation failure”.	Prueba exitosa. Se obtienen los resultados esperados.
Caso de Prueba #3		
Se solicita normalizar una imagen segmentada.	El sistema retorna una imagen del iris rectangular de dimensiones fijas.	Prueba exitosa. Se obtienen los resultados esperados.
Caso de Prueba #4		
Se solicita normalizar varias imágenes segmentadas con diferentes dimensiones y radios del iris.	El sistema retorna siempre una imagen del iris rectangular de dimensiones fijas asociada a la imagen de entrada.	Prueba exitosa. Se obtienen los resultados esperados.

Tabla 6: Casos de prueba para caso de uso Normalizar imagen.

Nombre del caso de uso:	Codificar imagen.	
Entrada	**Resultados esperados**	**Resultados obtenidos**
Caso de Prueba #1		

Se solicita codificar una imagen normalizada.	El sistema retorna el iriscode asociado a la misma.	Prueba exitosa. Se obtienen los resultados esperados.
Caso de Prueba #2		
Se solicita codificar dos veces la misma imagen normalizada.	El sistema retorna el mismo iriscode en ambos casos.	Prueba exitosa. Se obtienen los resultados esperados.
Caso de Prueba #3		
Se solicita codificar dos imágenes normalizadas diferentes.	El sistema retorna dos iriscode diferentes, cada uno asociado a una imagen.	Prueba exitosa. Se obtienen los resultados esperados.

Tabla 7: Casos de prueba para caso de uso Codificar imagen.

Nombre del caso de uso:	Comparar iriscodes.	
Entrada	**Resultados esperados**	**Resultados obtenidos**
Caso de Prueba #1		
Se solicita comparar dos iriscodes pertenecientes a la misma persona.	El sistema muestra el mensaje “Successful authentication”.	Prueba exitosa. Se obtienen los resultados esperados.
Caso de Prueba #2		
Se solicita comparar dos iriscodes pertenecientes a diferentes personas.	El sistema muestra el mensaje “Unsuccessful authentication”.	Prueba exitosa. Se obtienen los resultados esperados.
Caso de Prueba #3		
Se solicita identificar un iriscode dentro de una base	El sistema muestra el mensaje “Successful	Prueba fallida. De los 108 iricodes 4 no

de datos que contiene iriscodes.	authentication" si el iriscode es identificado.	fueron identificados correctamente.

Tabla 8: Casos de prueba para caso de uso Comparar iriscodes.

3.3.2 Validación de la solución.

Con el objetivo de validar la solución propuesta en el capítulo anterior se realizaron pruebas de identificación a fin de determinar si el módulo libIris alcanza un nivel de efectividad aceptable según los por cientos analizados en el estudio del estado del arte. Para ello se empleó la base de datos CASIA-1 descrita en el Capítulo 1.

Las pruebas de identificación consisten en comparar un código del iris con una base de datos de códigos del iris y decidir acerca de la identidad de dicho iris, o sea, a qué usuario registrado en la base de datos pertenece. Este proceso da lugar a cuatro resultados posibles:

- **Identificación correcta:** ocurre cuando el iris es correctamente identificado en la base de datos.
- **Rechazo correcto:** ocurre cuando el iris a buscar no está registrado en la base de datos, y el sistema correctamente falla en encontrar dicho iris en la base.
- **Identificación errónea:** ocurre cuando un iris es identificado erróneamente como otro iris registrado en la base.
- **Falso rechazo:** ocurre cuando la búsqueda falla en encontrar un código del iris previamente registrado en la base.(1)

Para la realización de las pruebas de identificación se tomó una muestra de 108 imágenes de la base de datos CASIA-1, una de cada carpeta, y se realizaron para cada una de ellas 755 verificaciones para un total de 81540. De las 108 imágenes procesadas 2 arrojaron falso rechazo, lo que equivale a un 2% aproximadamente, otras 2 arrojaron identificación errónea, lo que equivale a otro 2% aproximadamente y las restantes 104 imágenes arrojaron identificación correcta, lo que equivale a un 96% aproximadamente y lo que representa un resultado aceptable según el análisis del estado del arte.

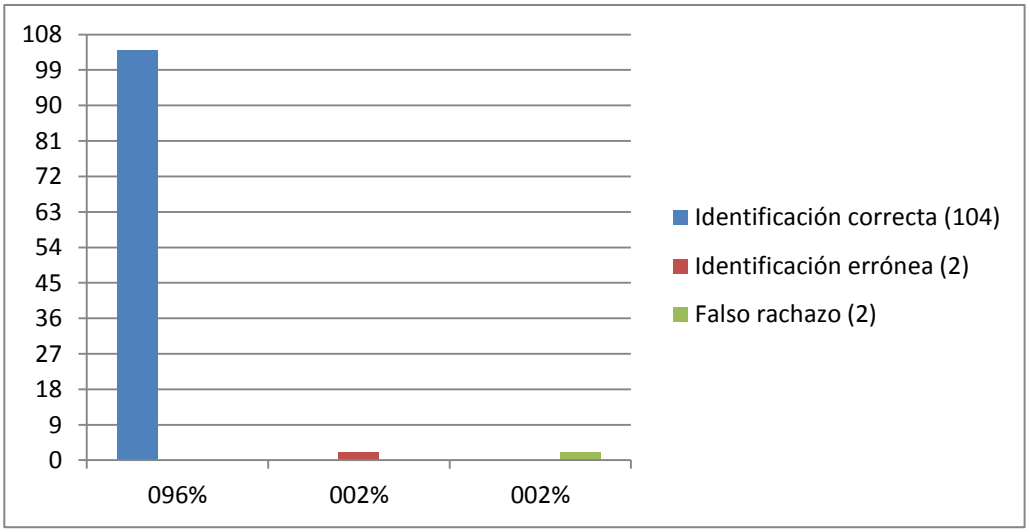

Figura 27: Gráfica de resultados de las pruebas de identificación.

Otro de los resultados obtenidos fue la mejora en cuanto a tiempo de ejecución, siendo este uno de los requisitos más importantes en un sistema de este tipo; o sea, el módulo que se propone es tres veces más rápido que la versión inicial de la biblioteca de autenticación biométrica del proyecto Seguridad. Para validar dicho resultado se tomó una muestra de 10 imágenes las cuales fueron procesadas por el módulo libIris y la versión inicial de la biblioteca, donde se obtuvo que para la etapa de Segmentación solamente, el módulo tarda un promedio de 667.6 milisegundos y la biblioteca un promedio de 2498.1 milisegundos; mientras que para el proceso de generar el iriscode el módulo demora un promedio de 839.9 milisegundos y la biblioteca un promedio de 2573.7 milisegundos, como se puede apreciar en las gráficas siguientes.

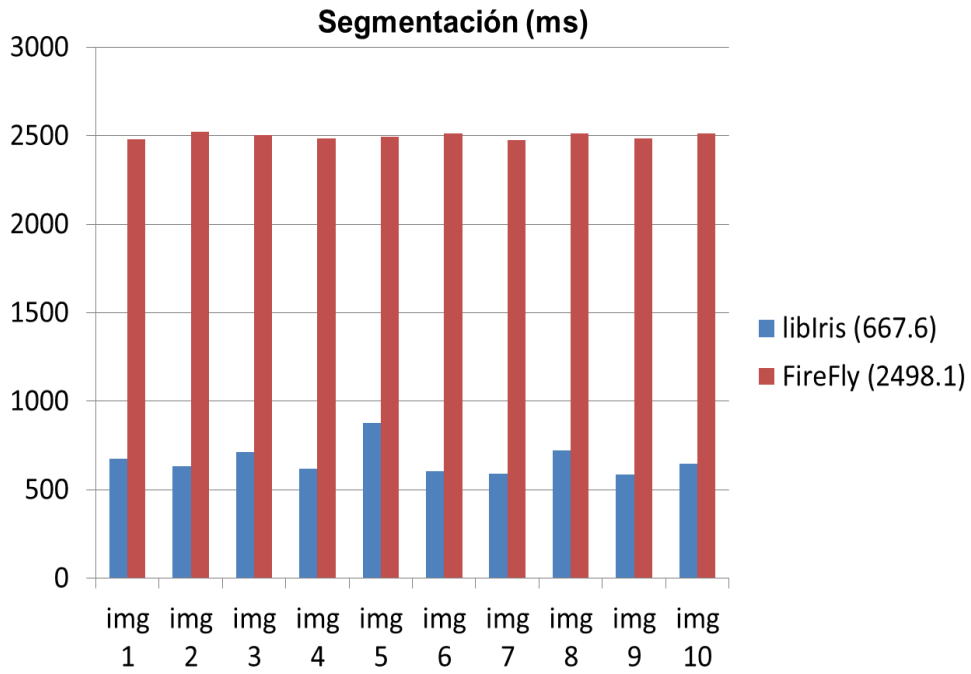

Figura 28: Gráfica de resultados para las pruebas de tiempo para la etapa de segmentación.

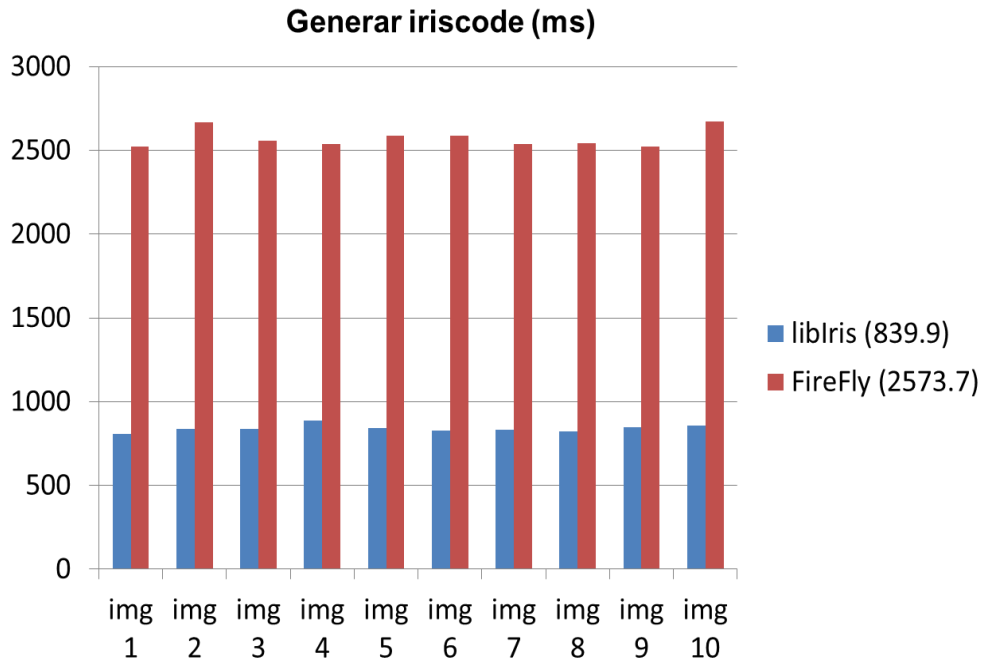

Figura 29: Gráfica de resultados para las pruebas de tiempo para generar el iriscode.

Conclusiones Generales

En este trabajo se desarrolló un módulo de autenticación biométrica por reconocimiento de patrones del iris para el módulo de seguridad del CEDIN, que contempla todas las etapas necesarias desde la captura de la imagen hasta la verificación final de la identidad del usuario, pasando por las etapas de pre-procesado, extracción de características y comparación.

Recomendaciones

A continuación se recomiendan posibles aspectos a tener en cuenta para continuar con la mejora y perfeccionamiento del módulo:

- Garantizar el hardware sobre el que va a trabajar el módulo, para realizar nuevas pruebas a fin de determinar el comportamiento del mismo en un ambiente real.

Referencias Bibliográficas

1. **Florian Cruz, Laura y Carranza Athó, Fredy.** *RECONOCIMIENTO DEL IRIS.* Trujillo, Perú : s.n., 2006.

2. **Mottalli, Marcelo Luis.** Implementación de un Sistema de Identificación de Personas en Tiempo Real por Reconocimiento de Iris. [En línea] Octubre de 2008. [Citado el: 20 de Septiembre de 2012.] www-2.dc.uba.ar/grupinv/imagenes/archivos/tesisMottalli2008.pdf.

3. **Tejedor Gómez, Jesús.** *Análisis Comparativo de Algoritmos en Segmentación de Iris.* España : Universidad Carlos III de Madrid, 2009.

4. **Rocchietti, Marco A. y Scerbo, Alejandro L.A.** *"Algoritmos de origen computacional como una alternativa eficaz para la segmentación y comparación digital del iris humano".* s.l. : Universidad Nacional de Córdoba, Marzo 2012.

5. **Peláez González, Ariel.** *Algoritmo de codificación para biblioteca de autenticación biométrica a partir del reconocimiento de patrones del iris.* La Habana, Cuba : s.n., 2011.

6. **Febles Parker, Michel Evaristo.** *Desarrollo de algoritmo de comparación para biblioteca de autenticación biométrica por reconocimiento de patrones del iris.* La Habana, Cuba : s.n., 2012.

7. **Elespectador.com.** tecno.americaeconomia.com. [En línea] Lun, 26 de Septiembre de 2011 . http://tecno.americaeconomia.com/noticias/reconocimiento-del-iris-para-la-seguridad-de-la-industria-aeronautica.

8. **Cornejo Herrera, Julio, Lara López, Adriana y Landa Becerra, Ricardo.** *Una biblioteca para procesamiento de imagen: scimagen.* México : CIE2002, 2002.

9. **M, Luis .** algoimagen.blogspot.com. [En línea] jueves 9 de mayo de 2013. http://algoimagen.blogspot.com/2013/05/biblioteca-de-clases-para-tratamiento.html.

10. **Carlos.** profesores.fi-b.unam.mx. [En línea] http://profesores.fi-b.unam.mx/carlos/aydoo/uml.html.

11. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** *El proceso unificado de desarrollo de software.*

12. **Kudriávtsev, V. A. y Demidóvich, B. P.** Capítulo IV. Líneas de segundo grado (cónicas). *Breve Curso de Matemáticas Superiores.* URSS : Mir Moscú, 1989.

13. **autores, C. d.** Tycon Software Ingineering. [En línea] Enero de 2011. [Citado el: 2 de Febrero de 2013.] http://www.tycon.com.ar/Tecnolog%C3%ADa/tabid/58/Default.aspx.

14. **Software, Laboratorio Nacional de Calidad del.** *GUÍA DE VALIDACIÓN Y VERIFICACIÓN.* España : Instituto Nacional de Tecnologías de la Comunicación., Noviembre 2009.

Bibliografía

1. **Ana Fred, J. F. (2009).** Biomedical Engineeiring Systems and Tecnologies . Porto, Portugal.

2. **Belmonte, R. C. (2006).** Sistema de reconocimiento de personas mediante su patrón de iris basado en la transformada de wavelet. Madrid, España.

3. **Daugman., J. G. (1988).** Complete Discrete 2D Gabor Transforms by Neural Networks for Image Analysis and Compression. Cabridge: Invited Paper.

4. **García, J. O. (2010).** Reconocimiento Automático de Patrones del Iris. Madrid, España.

5. **Glyn James, D. B. (2002).** Matemática avanzada para ingeniería. México.

6. **González, M. (2010).** Reconocimiento de iris. Barcelona.

7. **Laura Florian Cruz, F. C. (2006).** Reconocimiento del iris. Trijullo, Perú.

8. **Masek, L. (2003).** Recognition of human iris patterns for Biometric Identification. Western, Australia.

9. **Monro, D. M. (2007).** DCT Based Iris Recognition.

10. **Mottalli, M. L. (2008).** Implementación de un sistema de identificación de personas en tiempo real por reconocimiento de iris. Buenos Aires, Argentina.

11. **Movellan, J. R. (2008).** Tutorial on Gabor Filter.

12. **Muller, C. (2007).** IV Congreso Latinoamericano de ingeniería biométrica. Islas Margaritas, Venezuela.

13. **Rafael C Gonzalez, R. E. (1992).** Digital Image Processing. New Jersey: Prentice Hall.

14. **Rosario Almeyda, P. L. (2004).** Reconocimiento de iris. Recuperado el 20 de Enero de 2011, de Reconocimiento de iris.: http://iie.fing.edu.uy/investigacion/grupos/gti/timag/trabajos/2004/recon_iris

15. **Stan Z Li, Z. S. (2005).** Advances in biometric Person Authentication. Beijin.

16. **Teressi, L. D. (2000).** Sistema de Reconocimiento de Iris. Rosario, Argentina.

Anexos

Anexo 1: Imágenes de los resultados arrojados por el módulo.

Ejemplo 1.

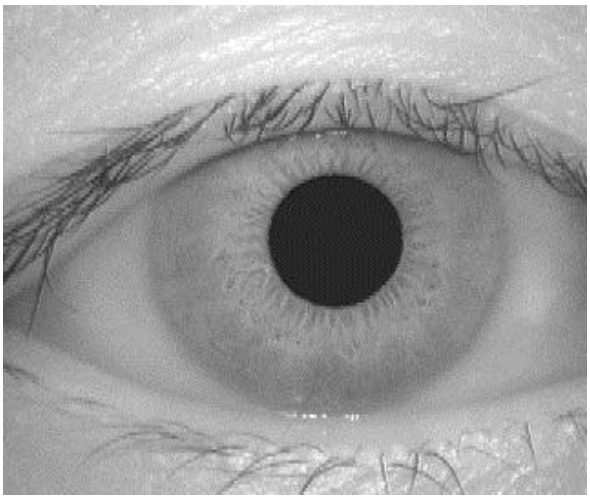

Figura 30:Imagen de entrada ejemplo 1.

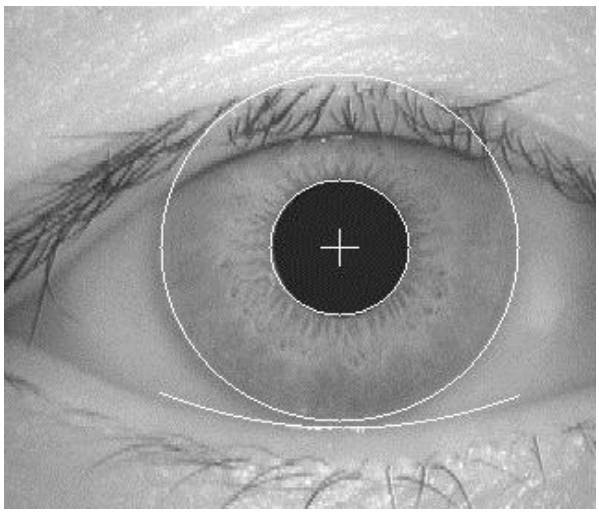

Figura 31:Imagen segmentada ejemplo 1.

Figura 32:Imagen normalizada ejemplo 1.

Figura 33:Máscara de ruido ejemplo 1.

Ejemplo 2.

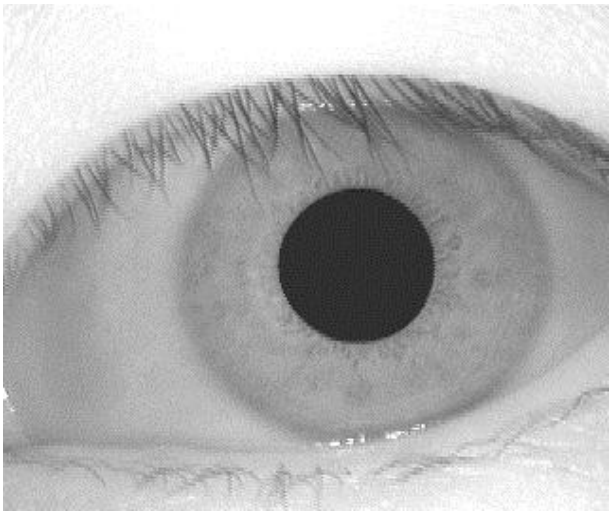

Figura 34:Imagen de entrada ejemplo 2.

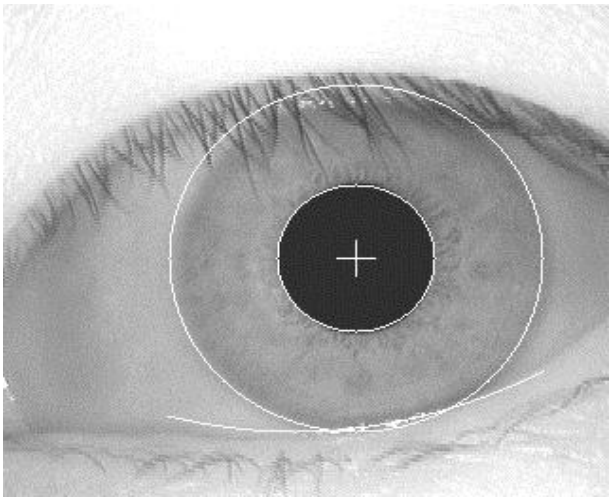

Figura 35:Imagen segmentada ejemplo 2.

Figura 36:Imagen normalizada ejemplo 2.

Figura 37:Máscara de ruido ejemplo 2.

Anexo 2: Descripción de las principales clases.

Nombre: Library	
Tipo de clase: Interfaz	
Atributo	**Tipo**
Path	String
imageCaptured	IplImage*
imageSegmented	IplImage*
imageNormalized	IplImage*
imageMask	IplImage*
pupilZone	PupilZone
irisCode	int*

Nombre:	**Library(string path)**
Descripción:	Constructor por defecto de la clase con la dirección de la imagen.
Nombre:	**~Library()**
Descripción:	Destructor de la clase.
Nombre:	**setPath(string path)**
Descripción:	Modifica la dirección donde se encuentra la imagen.
Nombre:	**getImageCaptured()**
Descripción:	Devuelve una imagen de un ojo.
Nombre:	**getPupilZone()**
Descripción:	Devuelve la zona de la pupila.

Nombre:	**getImageNormalizated()**
Descripción:	Devuelve una imagen con el iris normalizado o sea una barra rectangular con la textura del iris segmentado.
Nombre:	**getImageEncoded()**
Descripción:	Devuelve una imagen con el iriscode generado.
Nombre:	**getImageSegmented()**
Descripción:	Devuelve una imagen con el iris segmentado.
Nombre:	**exec()**
Descripción:	Se ejecuta la aplicación determinando si el usuario es quien dice ser o no.

Tabla 9: Descripción de la clase Library.

Nombre: Capture	
Tipo de clase: Entidad	
Atributo	**Tipo**
Path	String
Image	IplImage*
Nombre:	**Capture(string path)**
Descripción:	Constructor por defecto de la clase con la dirección de la imagen.
Nombre:	**~Capture()**
Descripción:	Destructor de la clase.
Nombre:	**getImageCaptured()**

Descripción:	Devuelve una imagen de un ojo.
Nombre:	**exec()**
Descripción:	Se ejecuta la aplicación obteniendo la imagen de un ojo.

Tabla 10: Descripción de la clase Capture.

Nombre: Segmentation	
Tipo de clase: Entidad	
Atributo	**Tipo**
imageCaptured	IplImage*
pupilZone	PupilZone
irisZone	IrisZone
irisRatio	Int
downPoints	vector<Coordinate>
Nombre:	**Segmentation(IplImage* imageCaptured)**
Descripción:	Constructor por defecto de la clase con la imagen capturada.
Nombre:	**~Segmentation()**
Descripción:	Destructor de la clase.
Nombre:	**lowPassFilter(IplImage* src)**
Descripción:	Aplica un filtro paso bajo a la imagen pasada por parámetro.
Nombre:	**binary(IplImage* src)**

Descripción:	Convierte la imagen pasada por parámetro en una imagen binaria.
Nombre:	**crossNorthTrace(IplImage* image, Coordinate point, int T)**
Descripción:	Devuelve la distancia del centro del iris y la pupila al límite exterior de la pupila para la traza norte.
Nombre:	**crossSourthTrace(IplImage* image, Coordinate point, int T)**
Descripción:	Devuelve la distancia del centro del iris y la pupila al límite exterior de la pupila para la traza sur.
Nombre:	**crossEastTrace(IplImage* image, Coordinate point, int T)**
Descripción:	Devuelve la distancia del centro del iris y la pupila al límite exterior de la pupila para la traza este.
Nombre:	**crossWestTrace(IplImage* image, Coordinate point, int T)**
Descripción:	Devuelve la distancia del centro del iris y la pupila al límite exterior de la pupila para la traza oeste.
Nombre:	**crossCenter(IplImage* image, Coordinate point, int T)**
Descripción:	Devuelve la distancia del centro del iris y la pupila al límite exterior de la pupila luego de realizar los recorridos en el sentido de las trazas.
Nombre:	**takePoints(IplImage* image, int porcentage)**
Descripción:	Devuelve los puntos de la imagen que se encuentran en un intervalo determinado por el por ciento pasado por parámetro.

Nombre:	**arrangePoints(vector <Coordinate>points)**
Descripción:	Ordena los puntos pasados por parámetros de menor a mayor según su intensidad de gris.
Nombre:	**findPupilCenter(IplImage* image, int T, int RM)**
Descripción:	Devuelve el centro de la pupila.
Nombre:	**findPupilRatio(IplImage* image, int T, int RM)**
Descripción:	Devuelve el radio de la pupila.
Nombre:	**averageCircle(int xc, intyc, int ratio)**
Descripción:	Calcula el promedio de intensidades de gris para una región circular determinada por el centro y el radio pasados por parámetros.
Nombre:	**irisRightZoneAverage(Coordinate pupilCenter, int ratio)**
Descripción:	Calcula el promedio de intensidades de gris para la región derecha de la imagen.
Nombre:	**irisLeftZoneAverage(Coordinate pupilCenter, int ratio)**
Descripción:	Calcula el promedio de intensidades de gris para la región izquierda de la imagen.
Nombre:	**variation(int x, int y, intminRatio, intmaxRatio, average TypeaveTy = pupil)**
Descripción:	Devuelve la mayor variación de gris en la imagen.
Nombre:	**average(IplImage* image, double p, int x0, int y0)**
Descripción:	Calcula el promedio de intensidades de gris para una parábola.

Nombre:	**fillDownVector(IplImage* image, double p, Coordinate coord)**
Descripción:	Llena un vector con las coordenadas del párpado inferior.
Nombre:	**definePupilZone()**
Descripción:	Delimita la región de la pupila.
Nombre:	**defineIrisRatio()**
Descripción:	Delimita la región del iris.
Nombre:	**defineEyelid()**
Descripción:	Delimita el párpado inferior.
Nombre:	**getPupilZone()**
Descripción:	Devuelve la región de la pupila.
Nombre:	**getIrisZone()**
Descripción:	Devuelve la reigón del iris.
Nombre:	**getIrisRatio()**
Descripción:	Delimita el radio del iris.
Nombre:	**getDownPoints()**
Descripción:	Devuelve las coordenadas del párpado inferior.
Nombre:	**exec()**
Descripción:	Se ejecuta la aplicación obteniendo la imagen de un ojo con la región del iris, la pupila y el párpado inferior delimitada.

Tabla 11: Descripción de la clase Segmentation.

Nombre: Normalization	
Tipo de clase: Entidad	
Atributo	**Tipo**
imageSegmented	IplImage*
imageNormalized	IplImage*
imageMask	IplImage*
Xc	Int
Yc	Int
irisRatio	Int
pupilRatio	Int
upPoints	vector<Coordinate>
downPoints	vector<Coordinate>

Nombre:	**Normalization(IplImage* image Segmented,int xc,int yc,int irisRatio, int pupilRatio, vector<Coordinate>upPoints, vector<Coordinate>downPoints)**
Descripción:	Constructor por defecto de la imagen segmentada, el punto del centro del iris y la pupila, el radio de la pupila y del iris y las coordenadas de los párpados superior e inferior.
Nombre:	**~Normalization()**
Descripción:	Destructor de la clase.
Nombre:	**getX(int angle, int ratio)**
Descripción:	Devuelve el valor de x en coordenadas cartesianas

	de un punto, dado el valor de su radio y ángulo en coordenadas polares.
Nombre:	**getY(int angle, int ratio)**
Descripción:	Devuelve el valor de y en coordenadas cartesianas de un punto, dado el valor de su radio y ángulo en coordenadas polares.
Nombre:	**defineRectangle()**
Descripción:	Devuelve el iris normalizado o sea una barra rectangular con la textura del iris segmentado.
Nombre:	**getImageNormalized()**
Descripción:	Devuelve una imagen con la imagen del iris normalizado.
Nombre:	**getImageMask()**
Descripción:	Devuelve una imagen con la máscara de ruido generado por párpados y pestañas.
Nombre:	**exec()**
Descripción:	Se ejecuta la aplicación obteniendo una imagen de dimensiones fijas con la textura del iris normalizado y su máscara de ruido asociada.

Tabla 12: Descripción de la clase Normalization.

Nombre: Encode	
Tipo de clase: Entidad	
Atributo	**Tipo**
imageNormalized	IplImage*
imageMask	IplImage*

irisCode	int*
Nombre:	**Encode(IplImage* imageNormalizated, IplImage* imageMask)**
Descripción:	Constructor por defecto de la clase con imagen normalizada y la máscara de ruido.
Nombre:	**~Encode()**
Descripción:	Destructor de la clase.
Nombre:	**defineIrisCode()**
Descripción:	Genera el iriscode, o sea, una estructura binaria obtenida a partir de patrones del iris.
Nombre:	**defineCode(int path, string dir, IplImage* img, int* code)**
Descripción:	Convierte el código pasado por parámetro en un código binario.
Nombre:	**getIrisCode()**
Descripción:	Devuelve el iriscode asociado a una imagen.
Nombre:	**exec()**
Descripción:	Se ejecuta la aplicación obteniendo el iriscode de la imagen segmentada.

Tabla 13: Descripción de la clase Encode.

Nombre: Matching	
Tipo de clase: Entidad	
Atributo	**Tipo**

code1	int*
code2	int*
Size	Int

Nombre:	**Matching(int* code1)**
Descripción:	Constructor por defecto de la clase con el código generado por el iris.
Nombre:	**~Matching()**
Descripción:	Destructor de la clase.
Nombre:	**Hamming(int* v1, int* v2)**
Descripción:	Devuelve el cálculo de la distancia de hamming entre dos códigos del iris
Nombre:	**rotateToLef(int* v)**
Descripción:	Devuelve el arreglo que se le pasa por parámetro rotado hacia la izquierda.
Nombre:	**rotateToRigth(int* v)**
Descripción:	Devuelve el arreglo que se le pasa por parámetro rotado hacia la derecha.
Nombre:	**defineHamming()**
Descripción:	Devuelve la menor de las distancias de haming obtenidas a partir de haber realizado todas las comparaciones entre las rotaciones a la izquierda y derecha entre los códigos del iris
Nombre:	**getCode2()**
Descripción:	Devuelve un arreglo que contiene el segundo código del iris a comparar.

Nombre:	**exec()**
Descripción:	Se ejecuta la aplicación determinando si el usuario es quien dice ser o no.

Tabla 14: Descripción de la clase Matching.

Glosario de Términos

A

- Aplicación: Terminología técnica para referirse a un programa de software.
- Arquitectura: Indica la estructura, funcionamiento e interacción entre las partes del software.
- Autenticación: Verificación de la identidad de una persona o de un proceso para acceder a un recurso o poder realizar determinada actividad.

C

- CEDIN: Siglas que identifican el Centro de Informática Industrial de la facultad 5 de la Universidad de las Ciencias Informáticas.
- Cliente: Un sistema o proceso que solicita a otro sistema o proceso que le preste un servicio.
- Contraseña: Información confidencial constituida por una cadena de caracteres, usada para la autenticación de un usuario o en el acceso a un recurso.
- Control de acceso: Mecanismo que en función de la identificación ya autenticada permite acceder a datos o recursos.

H

- Hardware: Componente físico de una computadora o de una red, en contraposición con los programas o elementos lógicos que lo hacen funcionar.

I

- IDE: Siglas en inglés que identifican un Entorno de Desarrollo Integrado (*Integrated Development Environment)*. Es un entorno de programación que ha sido empaquetado como un programa de aplicación, es decir, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica.
- Identificación del usuario: Procedimiento de reconocimiento de la identidad de un usuario.

P

- Programa: Conjunto de instrucciones que una vez ejecutadas realizan una o varias tareas en una computadora.

R

- Recurso: Cualquier parte componente de un sistema de información.

S

- SCADA: Siglas que identifican a un sistema desarrollado para el control y la supervisión de datos (Supervisory Control And Data Adquisition).
- Software: Conjunto de programas, documentos, procesamientos y rutinas asociadas con la operación de un sistema de computadoras, es decir, la parte intangible o lógica de una computadora.

U

- Usuario: Sujeto o proceso autorizado para acceder a datos o recursos.