



Universidad de las Ciencias Informáticas

Facultad 1

Título: Componente para la comparación de características de iris.

Trabajo de diploma para optar por el título de
Ingeniero en Ciencias Informáticas

Autores:

Mariena Ferrer Barrientos

Nathaly Rodríguez Guillén

Tutores:

Ing. Yainier Labrada Nueva

MsC. Adrian Alberto Machado Cento

Ciudad de La Habana, 2013.

Declaramos ser las autoras del trabajo titulado Componente para la comparación de características de iris de la Universidad de las Ciencias Informáticas y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo el presente a los ____ días del mes de ____ del año 2013.

Mariena Ferrer Barrientos

Nathaly Rodríguez Guillén

Ing. Yainier Labrada Nueva

Ing. Adrian Alberto Machado Cento.

Opinión del Tutor

Dedicatoria

At mi familia porque son mi espíritu y mi fuerza.

Mariana

At mis padres, a mis hermanos y a mis tios. At Mayra de la Caridad Pérez Álvarez. At los que tuvieron a mi lado todo este tiempo y aquellos que confiaron en mi.

Nathaly

Agradezco a

Mis padres Conrado Ferrer y Emelia Barrietos que con esfuerzo y desvelo han dedicado sus vidas para que sea una persona de bien, dándome su apoyo y su confianza en todo momento.

Mi hermana Soraya Ferrer, por ser esa hermana mayor súper protectora, mi mano derecha en todo momento. Por brindarme la posibilidad de ser tía de dos niños que me vuelven loca, Yusnel y Yausnelis.

Toda mi familia en general que de una forma u otra me han brindado su apoyo o han mostrado interés y preocupación por mí, en especial mis tíos Rosendo y Olga.

Todas las personas que he conocido, específicamente al club de las perras, (para no decir lo demás), que han estado conmigo en todos estos años, son mi segunda familia. En las buenas y en las malas, quienes me ayudaron, apoyaron y aconsejaron en los momentos más difíciles con los que compartí una magnífica etapa de mi vida y por las que me han aguantado en este último año Yelaysi, Lianny, Yasnelis, Leydis y mi compañera de tesis Nathaly por no haberse desmayado. A todas mis amistades por habernos soportado mutuamente los viejos Isunnis, Eliatne, Carmen, Martha, Deyanira, María, Madiela.

Adrián Hernández por su ayuda incondicional, con todas mis preguntas y dudas, al profesor Campanioni.

Mi gente de Santiago Rosanela, Yasmari, Dayenis, Eloydis y Madelaine que siempre están a la espera de cuando volveré.

Mariena

Agradezco a

En primer lugar le agradezco a mi abuela que aunque ya no está conmigo físicamente, su espíritu me guía y me sigue cuidando.

Mi mamita por darme la oportunidad de nacer, por guiarme, educarme y siempre estar conmigo, dándole los buenos días al sol.

Mis tíos, tía Tita y tío Nene, a mis hermanitas lindas y a mis sobrinas. Al resto de mi inmensa familia. A Mayra y a “Landy”. Todos los de AKDA.

Chiqui, Ada, Maritza López. Mis amistades Yumi y Adianis. Michel ya su Evelio.

Mis amistades por confiar en mí y apoyarme. Yelito por soportar todas mis malcriadeces. Las chicas del apartamento. A mi Compañera de tesis.

Mis tutores por su ayuda y colaboración.

Yoandy por soportarme y ser, aunque él no lo vea así, un amigo.

Ariel porque a pesar de todo aún sigue a mi lado como un amigo.

Tapita, Leo, Angelito, Jorge, Ariel, a todos los que hicieron de mis noches en el 6 algo especial.

Todas mis amistades de Bauta.

Y por último pero no menos especial, le agradezco a una persona, que aunque yo no sea su hija de sangre me dio la oportunidad de poder llamarlo Papá, a esa persona gracias.

Nathaly

Resumen:

La biometría hace uso de las características únicas de una persona para permitir su reconocimiento automático. Se han desarrollado sistemas biométricos basados en el reconocimiento de huellas dactilares, la voz, la fisiología de las manos o la cara, la firma manuscrita, y el iris.

El reconocimiento del iris en los últimos años ha mostrado un gran auge debido a los excelentes resultados obtenidos, demostrando una superioridad en la práctica de métodos biométricos con respecto a otros métodos.

En el presente trabajo se muestra el desarrollo de un componente que permite extraer características distintivas para completar el proceso de comparación biométrica basado en imágenes de iris en el departamento de Biometría, perteneciente al Centro de Identificación y Seguridad Digital de la Universidad de las Ciencias Informáticas. Para el desarrollo de dicho componente se utilizaron los principales algoritmos de extracción de características invariantes a la iluminación, según un análisis estadístico y el dominio de la frecuencia. Además, se proponen métodos para el reconocimiento del iris de bajo costo computacional y robusto a cambios de iluminación postura y expresión con el objetivo de ser capaz de identificar automáticamente a una persona mediante un iris de entrada y comparándolo con los iris existentes en una base de datos. Los métodos propuestos se encuentran divididos en cuatro etapas fundamentales: 1) extracción de características del iris por medio de la segmentación y normalización 2) extracción o codificación al vector extraído y 3) clasificación de patrones por medio de la distancia de Hamming de los vectores característicos extraídos.

Palabras claves:

Biometría, Segmentación, Normalización, Comparación de Iris.

Índice

| | |
|---|----|
| Introducción | 1 |
| Capítulo 1: Fundamentación teórica. | 5 |
| 1.1 Introducción..... | 5 |
| 1.2 El iris humano..... | 5 |
| 1.3 Historia del arte..... | 5 |
| 1.3.1 Sistemas homólogos..... | 6 |
| 1.3.1.1 Nivel internacional. | 7 |
| 1.3.1.1.1 IrisGuard..... | 7 |
| 1.3.1.1.2 BioID. | 7 |
| 1.3.1.1.3 Iris Recognition Immigration System (IRIS). | 8 |
| 1.3.1.2 Nivel nacional | 9 |
| 1.4 Proceso para la comparación del iris. | 9 |
| 1.4.1 Captura de la imagen..... | 10 |
| 1.4.2 Pre procesamiento de la imagen..... | 12 |
| 1.4.2.1 Segmentación..... | 13 |
| 1.4.2.1.1 Segmentación usando transformada de Hough..... | 14 |
| 1.4.2.2 Normalización..... | 15 |
| 1.4.2.2.1 Normalización usando Modelo de hoja de goma. | 15 |
| 1.4.3 Extracción de características..... | 16 |
| 1.4.3.1 Filtros de Gabor. | 17 |
| 1.4.4 Comparación de plantillas..... | 19 |
| 1.4.4.1 Distancia de Hamming. | 19 |
| 1.5 Estandarización y biometría..... | 20 |
| 1.5.1 Estándar BioAPI 2.0..... | 20 |
| 1.5.2 Estándar ISO/IEC 19794-6:2005..... | 21 |
| 1.5.3 Estándar ANSI / INCITS 358..... | 21 |
| 1.6 Metodologías, lenguajes y herramientas de desarrollo. | 21 |
| 1.6.1 Metodologías de desarrollo de software..... | 21 |
| 1.6.1.1 RUP..... | 22 |
| 1.6.1.2 XP..... | 23 |
| 1.6.1.3 RUP vs XP..... | 23 |

| | | |
|--|--|----|
| 1.6.2 | Lenguaje de modelado..... | 24 |
| 1.6.3 | Herramientas de modelado..... | 24 |
| 1.6.3.1 | Herramienta CASE..... | 24 |
| 1.6.3.1.1 | Visual Paradigm para UML..... | 24 |
| 1.6.4 | Lenguaje de programación..... | 25 |
| 1.6.4.1 | Lenguaje MatLab. | 25 |
| 1.6.5 | Entorno Integrado de Desarrollo. | 25 |
| 1.7 | Conclusiones parciales..... | 26 |
| Capítulo II: Análisis y diseño..... | | 27 |
| 2.1 | Introducción..... | 27 |
| 2.2 | Exploración..... | 27 |
| 2.2.1 | Descripción del sistema. | 27 |
| 2.2.2 | Arquitectura de software. | 29 |
| 2.2.3 | Usuarios del sistema..... | 29 |
| 2.2.4 | Historias de usuarios..... | 30 |
| 2.2.5 | Requisitos no funcionales. | 33 |
| 2.3 | Planificación..... | 34 |
| 2.3.1 | Estimación de esfuerzos por historias de usuarios..... | 34 |
| 2.3.2 | Plan de iteraciones..... | 35 |
| 2.3.3 | Plan de duración de las iteraciones..... | 35 |
| 2.4 | Conclusiones parciales..... | 36 |
| Capítulo III: Implementación y prueba..... | | 37 |
| 3.1 | Introducción..... | 37 |
| 3.2 | Fase de diseño..... | 37 |
| 3.2.1 | Funciones implementadas para el desarrollo del componente..... | 39 |
| 3.2.2 | Estándar de programación..... | 40 |
| 3.2.3 | Tareas de ingeniería..... | 41 |
| 3.2.4 | Requisitos para la integración del componente..... | 44 |
| 3.3 | Pruebas..... | 45 |
| 3.3.1 | Diseño de pruebas..... | 45 |
| 3.3.2 | Pruebas de aceptación..... | 45 |
| 3.3.3 | Etapa de verificación..... | 51 |

| | |
|-----------------------------------|----|
| 3.3.3.1 Resultados obtenidos..... | 51 |
| 3.4 Conclusiones parciales..... | 52 |
| Conclusiones | 53 |
| Recomendaciones | 54 |
| Referencias Bibliográficas..... | 55 |
| Glosario de Términos..... | 59 |
| Anexo | 61 |

Índice de Figuras

| | |
|---|----|
| Figura 1: Iris humano. | 5 |
| Figura 2: Control de seguridad en las fronteras de los Emiratos Árabes. | 7 |
| Figura 3: Niña afgana, Sharbat Gula. | 9 |
| Figura 4: Pasos para el procesamiento y comparación de la imagen. | 10 |
| Figura 5: Esquema propuesto por Flom y Safir en 1987. | 11 |
| Figura 6: Esquema propuesto por Daugman (Izquierda) y Wildes (Derecha). | 11 |
| Figura 7: Ejemplo de detección de los bordes del iris. | 13 |
| Figura 8: Iris segmentado a partir de la transformada de Hough. | 14 |
| Figura 9: Modelo Daugman de hoja de goma. | 15 |
| Figura 10: Iris normalizado. | 16 |
| Figura 11: Un par de cuadratura de filtros de Gabor 2D (izquierda) componente real o incluso filtro simétrico caracterizado por un coseno modulado por una gaussiana; (derecha) componente imaginario o filtro simétrico raro caracterizado por una condición seno modulada por una gaussiana. | 18 |
| Figura 12: Metodología RUP. | 22 |
| Figura 13: Modelo de Dominio. | 28 |
| Figura 14: Arquitectura de software. | 29 |
| Figura 15: Diagrama de componente. | 38 |
| Figura 16: Imágenes de ojos fotografiados con cámara NIR, pertenecientes a la base de datos CASIA. ... | 46 |
| Figura 17: Gráfica sobre los resultados de las pruebas. | 52 |

Índice de Tablas

| | |
|--|----|
| Tabla 1: Tareas de investigación. | 2 |
| Tabla 2: Comparativa de técnicas biométricas. (1) | 6 |
| Tabla 3: Historia de usuario Transformar imagen a escala de grises. | 30 |
| Tabla 4: Historia de usuario Segmentar imagen de iris. | 30 |
| Tabla 5: Historia de usuario Normalizar imagen de iris. | 31 |
| Tabla 6: Historia de usuario Extraer las características..... | 32 |
| Tabla 7: Historia de usuario Exportar vector característico. | 32 |
| Tabla 8: Historia de usuario calcular la distancia de Hamming. | 33 |
| Tabla 9: Estimación de esfuerzos por Historias de usuarios. | 34 |
| Tabla 10: Estimación de esfuerzos por Historias de usuarios. | 35 |
| Tabla 11: HU1_Transformar imágenes a escala de grises..... | 41 |
| Tabla 12: HU2_Segmentar imagen de iris. | 42 |
| Tabla 13: HU3_Normalizar imagen de iris..... | 42 |
| Tabla 14: HU4_Extraer características de iris. | 43 |
| Tabla 15: HU5_Exportar vector característico..... | 43 |
| Tabla 16: HU6_Calcular distancia de Hamming..... | 43 |
| Tabla 17: Caso de prueba para la HU Transformar a escala de grises. | 46 |
| Tabla 18: Caso de prueba para la HU Segmentar imagen de iris..... | 47 |
| Tabla 19: Caso de prueba para la HU Normalizar imagen de iris..... | 48 |
| Tabla 20: Caso de prueba para la HU Extraer las características. | 48 |
| Tabla 21: Caso de prueba para la HU Exportar vector característico..... | 49 |
| Tabla 22: Caso de prueba para la HU Calcular distancia de Hamming..... | 50 |
| Tabla 23: Resultados de las pruebas..... | 51 |

Introducción

Con el desarrollo de las tecnologías de la información, los sistemas inteligentes para la identificación y verificación de personas se han convertido en una herramienta muy útil.

La biometría es el estudio de métodos para el reconocimiento único de personas a través de sus facultades físicas o comportamientos. Este interesante campo de investigación, de gran amplitud, ha sido, es y será utilizado para realizar labores de seguridad, de distinta índole y grado, despertando el interés en el ámbito público y privado, aunque con ciertas críticas relacionadas con la privacidad personal. (1)

Dentro del gran ámbito de la biometría donde se puede destacar, el reconocimiento de huellas dactilares, la fisiología de las manos o la cara, la firma manuscrita, el reconocimiento de voz; se puede resaltar el reconocimiento de iris como una herramienta biométrica para el reconocimiento de personas de manera unívoca y de suma precisión.

El reconocimiento de iris es uno de los métodos biométricos más confiables que existen en la actualidad, sus soluciones comerciales están basadas en la toma de imágenes con una alta resolución, ofreciendo una óptima garantía de fiabilidad en entornos condicionados. Los estudios han demostrado una fiabilidad de 1 entre 16 millones de encontrar dos iris exactamente iguales. (2)

Cuba no está ajena al desarrollo de la biometría, la cual se manifiesta en un sinnúmero de aplicaciones para la investigación criminal, la protección y la seguridad, demostrando ser un novedoso método de identificación humana por excelencia, de ahí el especial interés de propiciar el intercambio entre profesionales cubanos y extranjeros con reconocida pericia en el desarrollo de aplicaciones biométricas. Actualmente existe en Cuba el Centro de Aplicaciones de Tecnologías de Avanzadas (CENATAV), donde se realizan estudios de los patrones biométricos para el reconocimiento de personas, así como también en la Asociación Cubana de Reconocimiento de Patrones. (3)

En la Universidad de las Ciencias Informáticas (UCI) se encuentra el Centro de Identificación y Seguridad Digital (CISED), el cual cuenta con el departamento de Biometría, que se encarga de estudiar los patrones para el reconocimiento único de personas basados en rasgos físicos o de conducta, como: huellas dactilares, firma manuscrita, reconocimiento facial y el reconocimiento de iris. En dicho departamento se pretende implementar las funcionalidades concernientes a los procesos de comparación de características del iris humano.

Después de planteada la situación problemática se puede enunciar el **problema de investigación** a partir de la siguiente interrogante ¿Cómo comparar características distintivas de iris para el proceso de identificación en el departamento de Biometría?

Determinando para el presente trabajo de diploma como **objeto de estudio** el proceso de comparación de iris.

Para dar solución al **problema de investigación** se plantea como objetivo general desarrollar un componente para la comparación de características distintivas en iris que se utilice en el proceso de identificación biométrica.

El desarrollo del componente tendrá como **campo de acción** la comparación de iris utilizando algoritmos de comparación de iris.

Los objetivos específicos que se persiguen son:

- ✓ Elaborar un marco teórico de la investigación.
- ✓ Analizar las tendencias actuales para la comparación de características en iris.
- ✓ Diseñar el componente para la comparación de características del iris.
- ✓ Implementar el componente para la comparación de características del iris.
- ✓ Validar la solución del componente de comparación de iris.

Para dar cumplimiento a los objetivos planteados se planificaron las siguientes **tareas de investigación:**

Tareas a cumplir:

Tabla 1: Tareas de investigación.

| No. | Tarea | Responsable | Fecha inicio | Fecha fin | Observaciones |
|-----|---|---------------------------|------------------|---------------|---------------|
| 1 | Análisis de los principales estándares que se involucran en la comparación de iris. | Nathaly Rodríguez Guillén | Septiembre 2012. | Octubre 2012. | |
| 2 | Análisis y selección los distintos algoritmos existentes | Mariena Ferrer | Septiembre | Octubre | |

| | | | | | |
|---|---|---|-----------------|-----------------|--|
| | para la comparación de características en iris. | Barrientos | 2012. | 2012. | |
| 3 | Análisis y selección de las tecnologías y herramientas apropiadas que permitan el desarrollo del componente para la comparación de características en iris. | Mariena Ferrer Barrientos | Octubre 2012. | Noviembre 2012. | |
| 4 | Diseño del componente para la comparación de iris. | Nathaly Rodríguez Guillén | Noviembre 2012. | Enero 2013. | |
| 5 | Implementación del algoritmo seleccionado para la comparación de características en iris. | Nathaly Rodríguez Guillén | Enero 2013. | Mayo 2013. | |
| 6 | Realización de las pruebas al componente de comparación de características en iris verificando su correcto funcionamiento. | Nathaly Rodríguez Guillén. Mariena Ferrer Barrientos | Mayo 2013. | Junio 2013. | |

Para el desarrollo de la investigación y lograr caracterizar a fondo el objeto de estudio, así como para garantizar el conocimiento del estado del arte, su evolución y relación con otros fenómenos se utilizaron los siguientes **métodos científicos**.

Dentro de los métodos científicos utilizados se encuentra el **histórico-lógico** el cual posibilita la comprensión lógica del objeto de estudio haciendo un análisis riguroso de sus antecedentes y el proceso evolutivo por el cual han transitado todas las tecnologías relacionadas con el reconocimiento de iris.

Se utiliza el método **analítico-sintético** que permite la consulta de diversas fuentes bibliográficas y la extracción de los elementos más importantes que se relacionan con el objeto de estudio.

La **justificación de la investigación** del trabajo se basa en el desarrollo de un componente que permita realizar la comparación de características de iris. Además, forma parte de los esfuerzos del CISED para

crear sistemas propios de identificación, representando un paso importante para encontrar soluciones alternativas y de buena calidad a los costosos sistemas de identificación existentes en el mercado.

El contenido está estructurado en tres capítulos, los cuales se agrupan de la manera siguiente:

Capítulo 1-Fundamentación teórica: Este capítulo incluye un estudio del estado del arte del tema tratado; se abordan elementos teóricos de la investigación tales como iris, proceso de comparación de iris, metodologías, lenguajes y herramientas de desarrollo que serán utilizadas para implementar la solución.

Capítulo 2-Análisis y diseño: En este capítulo se presentan las fases de análisis y diseño definidas por la metodología de Programación eXtrema (XP por sus siglas en inglés), para dar solución al problema científico. Se selecciona la arquitectura para desarrollar la solución. Se identifican las historias de usuarios y los requisitos no funcionales, se realiza el plan de iteraciones y plan de entregas.

Capítulo 3-Implementación y prueba: Este capítulo da cumplimiento a los planes trazados a través de las fases diseño y prueba, se codifica la solución diseñada y finalmente se realizan las pruebas de aceptación.

Capítulo 1: Fundamentación teórica.

1.1 Introducción.

En el presente capítulo se define un conjunto de ideas y conceptos que enmarcan al lector en el contexto en el que se desarrolla la investigación. A través de conceptos y definiciones adquiridos en el estado del arte y las tendencias actuales en el desarrollo de sistemas automatizados de comparación de iris, se pretende lograr una base teórica sólida. Además de realizar un análisis comparativo de las principales metodologías existentes, hacer un estudio de las tecnologías y herramientas que se utilizan a lo largo de la presente investigación y su desarrollo. También se realiza una pequeña descripción del proceso de comparación de características del iris, así como la descripción de los diferentes algoritmos para la comparación de iris.

1.2 El iris humano.

El iris es la membrana coloreada y circular del ojo (Ver Figura 1) que separa la cámara anterior de la cámara posterior. Posee una apertura central de tamaño variable que comunica las dos cámaras llamada pupila. Su función principal es controlar la cantidad de luz que penetra en el ojo. Corresponde a la porción más anterior de la túnica vascular, la cual forma un diafragma contráctil delante del cristalino. Se ubica tras la córnea, entre la cámara anterior y el cristalino, al que cubre en mayor o menor medida en función de su dilatación. (4) (5)

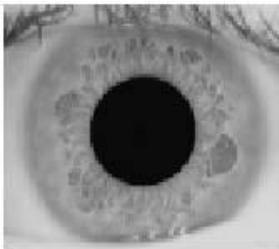


Figura 1: Iris humano.

1.3 Historia del arte.

La primera utilización del iris se presentó en París, donde los criminales eran clasificados de acuerdo al color de sus ojos siguiendo la propuesta del oftalmólogo francés Alphonse Bertillon. En 1936, el oftalmólogo Frank Bruch, propuso el concepto de usar patrones del iris como método para reconocer personas (6). Después de investigar y documentar el uso potencial del iris como instrumento para identificar personas, los oftalmólogos Flom y Safir en 1985 propusieron la idea de que no existen dos iris

semejantes y en el año 1987 fue que patentaron este concepto; y posteriormente, en 1994, junto al Dr. Daugman patentaron los algoritmos respectivos. (2)

En el 2005, la amplia patente que cubría el concepto básico de reconocimiento de iris expiro, proporcionando oportunidades comerciales para otras compañías que han desarrollado sus propios algoritmos para el reconocimiento de iris. (7)

En la siguiente tabla (Tabla 2 - Comparativa de técnicas biométricas) se puede encontrar una comparativa entre los más importantes sistemas biométricos actualmente en vigor, donde se percibe claramente como el reconocimiento de iris ofrece una de las mejores prestaciones. (1)

Tabla 2: Comparativa de técnicas biométricas. (1)

| | Fiabilidad | Facilidad de uso | Prevención de ataques | Aceptación | Estabilidad | Promedio |
|-----------------------------|------------|------------------|-----------------------|------------|-------------|----------|
| Reconocimiento de iris | Muy alta | Media | Muy alta | Media | Alta | Alto |
| Reconocimiento de retina | Muy alta | Baja | Muy alta | Baja | Alta | Medio |
| Reconocimiento dactilar | Alta | Alta | Alta | Alta | Alta | Alto |
| Reconocimiento geo. Mano | Media | Alta | Alta | Alta | Media | Medio |
| Reconocimiento de escritura | Media | Alta | Media | Muy alta | Baja | Medio |
| Reconocimiento de voz | Alta | Alta | Media | Alta | Media | Medio |
| Reconocimiento facial | Media | Alta | Media | Muy alta | Media | Medio |

1.3.1 Sistemas homólogos.

Actualmente con el desarrollo de las tecnologías se cuenta con herramientas y sistemas de alta potencia

que permiten realizar procesos de manera automatizada, y los sistemas de identificación basados en iris se han extendido a diversas esferas de la sociedad. A continuación se describen algunos componentes que realizan el proceso de comparación de iris.

1.3.1.1 Nivel internacional.

1.3.1.1.1 IrisGuard.

El control de seguridad en las fronteras de los Emiratos Árabes ha estado operando un sistema de seguimiento en los Emiratos Árabes Unidos (EAU) desde 2001 (IrisGuard), cuando la EAU puso en marcha una iniciativa a nivel nacional sobre la seguridad en las zonas fronterizas. Hoy en día, todo el país de los EAU, aeropuertos y puertos marítimos de entrada y salida están equipados con estos sistemas. Todos los extranjeros que poseen un visado para entrar en los EAU se procesan a través de cámaras de iris instaladas en todos los puntos de inspección de inmigración (primarios y auxiliares). Hasta la fecha, el sistema ha detenido a más de 330.000 personas que intentaban entrar en los EAU con documentos de viaje fraudulentos. (8)



Figura 2: Control de seguridad en las fronteras de los Emiratos Árabes.

1.3.1.1.2 BioID.

La tecnología de reconocimiento del iris ha sido aplicada por BioID Technologies SA en Pakistán para el proyecto del ACNUR (agencia de la ONU para refugiados), cuyo objetivo es controlar la distribución de ayuda a los refugiados afganos. Los refugiados son repatriados por la ACNUR en cooperación con el Gobierno de Pakistán, y se les da una cantidad de dinero para comenzar de nuevo. Para asegurarse de que a la gente no se les paga más de una vez, su iris se escanea, y el sistema detectará a los refugiados en el siguiente intento. La base de datos cuenta con más de 1,3 millones de plantillas del código del iris y

alrededor de 4000 registros por día. El tiempo de comparación del iris de un individuo con los 1,3 millones de plantillas se realiza en 1,5 segundos. (9)

BioID Technologies es una identificación de la empresa Soluciones y Servicios. Fundada en el año 2002 ha desarrollado una variedad de soluciones y referencias de clientes incorporados en los proyectos de identificación a gran escala. Su equipo de gestión cuenta con más de 20 años de experiencia en el desarrollo e implementación de soluciones de tecnología biométricas basadas en todos los mercados verticales. La compañía se centra en soluciones de identidad que responden a necesidades específicas de los clientes en lugar de defender una tecnología particular. (10)

1.3.1.1.3 Iris Recognition Immigration System (IRIS).

IRIS hace uso del hecho de que el patrón del iris en el ojo de cada persona (la parte coloreada del ojo de cada persona) es único. Un pasajero registrada mira en una cámara especial en la barrera de IRIS en el pasillo de la llegada de inmigración, y el sistema compara el patrón del iris con otros almacenados en una base de datos segura. Si el patrón coincide con el correspondiente en la base de datos y el registro sigue siendo válido, el pasajero puede entrar inmediatamente en el Reino Unido.

IRIS es el primer sistema de control biométrico habilitado automatizada frontera instalada en el Reino Unido. Tecnologías tienen una vida finita y este sistema ahora ha estado operando desde hace casi 7 años.

IRIS ha proporcionado grandes beneficios a nuestra operación fronteriza, pero ahora estamos estableciendo nuestras necesidades de negocio a largo plazo para nuestros sistemas de compensación automatizadas. (11)

Este sistema de reconocimiento se utilizó para identificar a la “niña afgana” (Sharbat Gula) 28 años después de que el fotógrafo Steve McCurry sacara en la portada de la revista *National Geographic* una imagen suya con 12 años. (12) (13)

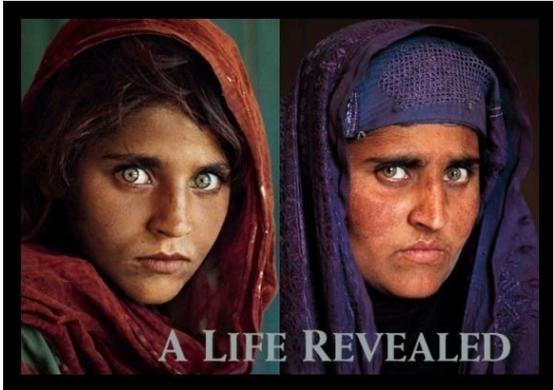


Figura 3: Niña afgana, Sharbat Gula.

1.3.1.2 Nivel nacional

CENATAV es un centro orientado a la investigación teórica y aplicada en el área de Reconocimiento de Patrones y Minería de Datos. El departamento de Biometría está involucrado en el desarrollo de nuevos métodos de identificación basados en diversas tecnologías biométricas, así como su introducción en los diferentes ámbitos de nuestra sociedad. En el centro se comenzaron estudios sobre la identificación de personas por la textura del iris en el año 2005, que transitó por todas las etapas de un sistema típico de reconocimiento, pero en la actualidad no existe ningún sistema cubano de reconocimiento de iris en explotación nacional. A pesar de ello, en estos momentos se desarrolla un proyecto de investigación teórica sobre el tema, donde está en vías de terminación dos estados del arte y otro sobre diferentes aspectos del tema de reconocimiento de iris. (14)

1.4 Proceso para la comparación del iris.

Todo sistema biométrico está compuesto de un conjunto de fases o etapas, al igual que el resto de los métodos. El estudio a realizar, dentro del ámbito del reconocimiento de iris se centrará en la comparación de las características tomadas del iris. En el siguiente diagrama se puede observar esquemáticamente dicho proceso.

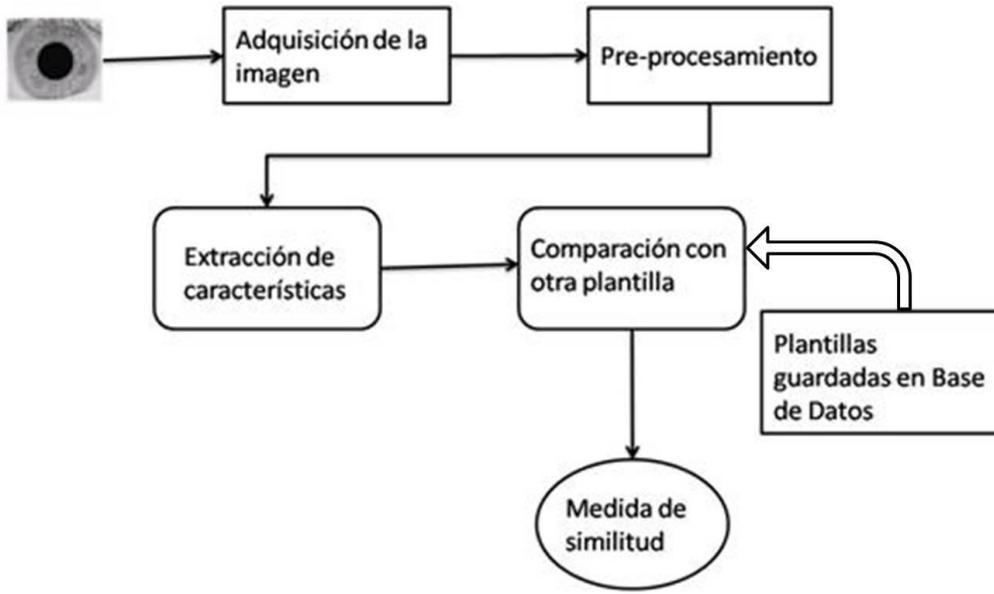


Figura 4: Pasos para el procesamiento y comparación de la imagen.

Todas las propuestas existentes en la actualidad contienen cuatro etapas fundamentales a saber: a) captura de la imagen del iris, b) procesamiento de la imagen, c) extracción del patrón numérico identificador y d) discriminación y reconocimiento de patrones. En este sentido, la literatura es variada en relación a los métodos de localización del iris, las técnicas para la extracción de los patrones, la robustez frente a patrones borrosos (ruido) y la duración del procesamiento computacional.

1.4.1 Captura de la imagen.

El primer esquema propuesto para la adquisición del iris se trataba de un sistema que controlaba la iluminación mediante cuatro puntos de enfoque, la reflexión producida por el ojo en una lente proporcionaba la imagen buscada. En la Figura 8 mostramos el sistema propuesto por los doctores Flom y Safir en 1987 basado en la iluminación directa y captura de la imagen frontalmente. (15)

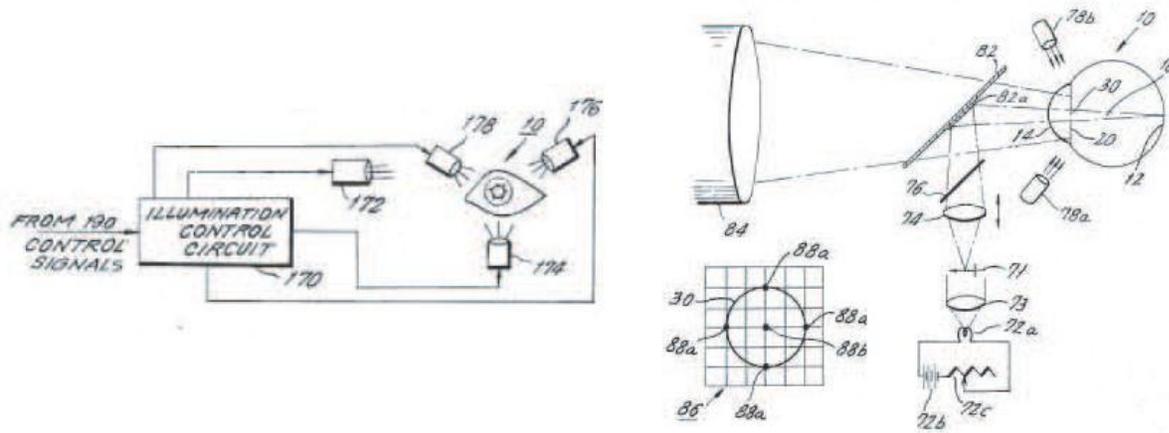
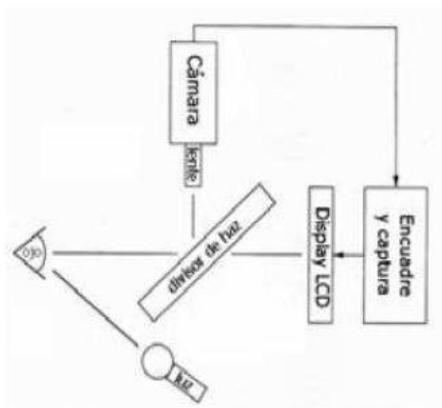
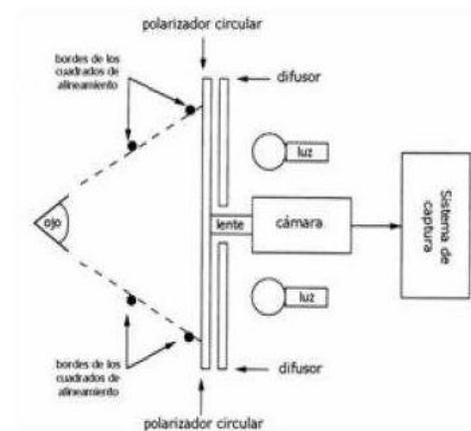


Figura 5: Esquema propuesto por Flom y Safir en 1987.

Posteriormente surgieron los sistemas de Daugman y Wildes que funcionaban capturando la imagen con el reflejo en un espejo, después de haber iluminado el ojo de forma diagonal (Daugman) o iluminando con dos puntos de luz y colocando la cámara enfrente al ojo. En la Figura 9 podemos observar los sistemas propuestos por Daugman (16) y Wildes (17).



(Sistema Daugman)



(Sistema Wildes et al)

Figura 6: Esquema propuesto por Daugman (Izquierda) y Wildes (Derecha).

En ambos sistemas los niveles de iluminación son bastantes bajos. Para confort del usuario, se suelen realizar aperturas de diafragma relativamente grandes con poca profundidad de campo. Ambos sistemas aprovechan las ventajas que proporciona la toma de secuencias de imágenes capturándose una secuencia de vídeo y eligiendo la mejor imagen.

Aún con estos sistemas, es necesario que el usuario sea cooperativo ya que si no el posicionamiento del iris se puede ver afectado por problemas de oclusión y desviación de ángulo. (16)

Actualmente existen cámaras y dispositivos para capturar de forma digital la imagen de un iris, así como bases de datos con imágenes de ojos.

Debido a la naturaleza del proceso de adquisición de pueden dar por diversos motivos imperfecciones en la imagen capturada, por ello se proponen medidas de calidad para discernir si una imagen es válida para su análisis o si es necesario adquirir una nueva. Existen muchas medidas de calidad propuestas en la actualidad: (15)

- ✓ Medidas de oclusión: miden la zona del iris disponible a ser analizada que no es tapada por párpados, pestañas u otros elementos.
- ✓ Medidas de borrosidad: miden la borrosidad de la imagen debido al movimiento del ojo en el momento de la captura o al enfoque de la cámara que realiza la captura.
- ✓ Medidas de contraste: miden los cambios en la escala de grises de las imágenes a imagen a ser analizadas.
- ✓ Medidas de dilatación: miden la relación entre iris y pupila.
- ✓ Medidas de desalineamiento angular: mide la inclinación del ojo hacia algún lateral de la imagen en la captura.

Estas medidas surgen como una ayuda al procesado de la imagen ya que para poder segmentar una imagen de iris o poder hacer otras operaciones necesarias para la identificación o verificación de una imagen, son necesarios unos mínimos de calidad, sino sería imposible el análisis.

1.4.2 Pre procesamiento de la imagen.

La etapa de pre-procesamiento toma una gran importancia en esta técnica, ya que la labor de adaptar la señal conlleva:

- ✓ La localización del iris dentro de la imagen.
- ✓ La detección de los bordes del iris. En este caso hay que tratar con dos bordes: el exterior (frontera con la esclerótica) y el interior (límite de la pupila).
- ✓ Eliminación de las partes de la imagen no deseadas.

- ✓ Compensación del tamaño del iris, debido a la distancia del sujeto respecto al objetivo, y de la dilatación o contracción de la pupila.

Por estos motivos es necesario aplicar un procesamiento a la imagen del iris antes de utilizarlas en la etapa de codificación (extracción de características), el cual puede ser dividido en dos etapas:

- ✓ Segmentación, donde se localiza la imagen del iris.
- ✓ Normalización, por la cual se obtiene una imagen del iris que es independientemente del tamaño de la pupila y permite la comparación entre diferentes iris.

1.4.2.1 Segmentación.

El procedimiento automático de localización se basa en detectar los círculos interior y exterior del iris que definen sus bordes. El proceso en si es una tarea compleja, puesto que la forma del objeto a segmentar no tiene una forma regular y sus límites no están siempre bien definidos.

La etapa de segmentación es fundamental para el éxito de un sistema de reconocimiento del iris, ya que los datos mal segmentados generan un patrón de iris corrupto, lo que provocará errores de reconocimiento. El éxito de la segmentación depende de la calidad de la imagen en base a tres puntos clave:

- ✓ Sensibilidad a una alta gama de contrastes entre los bordes.
- ✓ Robustez ante irregularidades en los bordes.
- ✓ Capacidad de considerar aperturas y cierres pupilares variables.

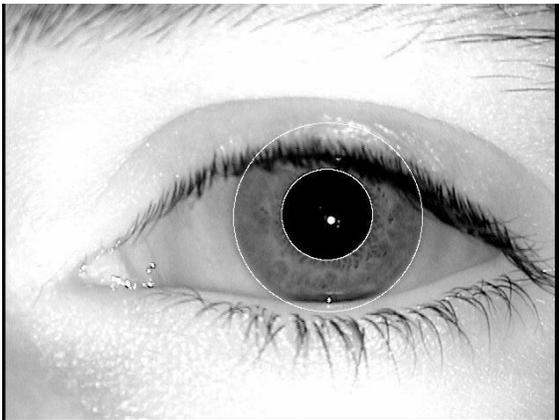


Figura 7: Ejemplo de detección de los bordes del iris.

1.4.2.1.1 Segmentación usando transformada de Hough.

La transformada de Hough es un algoritmo que se puede utilizar para determinar los parámetros de objetos geométricos simples, tales como líneas y círculos, presentes en una imagen. La transformada circular Hough puede emplearse para deducir las coordenadas de radio y el centro de la pupila y las regiones del iris. Un algoritmo de segmentación automática basada en la transformada circular de Hough es empleado por Wildes et al. (17), Kong y Zhang (18), Tisse et al. (19), y Ma et al. (20). En primer lugar, un mapa de borde se genera mediante el cálculo de las primeras derivadas de valores de intensidad en una imagen del ojo y luego la umbralización del resultado. Desde el mapa de borde, los votos se emiten en el espacio Hough para los parámetros de círculos que pasan a través de cada punto del borde. Estos parámetros son el centro de coordenadas x_c y y_c , y el radio r , que son capaces de definir cualquier círculo de acuerdo con la ecuación:

$$x_c^2 + y_c^2 - r = 0$$

Un punto máximo en el espacio de Hough se corresponde a las coordenadas del radio y el centro del círculo definido mejor por los puntos de borde. Wildes et al. (17) y Kong Zhang (18) también hacen uso de la transformada parabólica de Hough para detectar los párpados, la aproximación de los párpados superior e inferior con arcos parabólicos, que se representan como:

$$\left(- (x - h_j) \sin \theta_j + (y - k_j) \cos \theta_j\right)^2 = a_j \left((x - h_j) \cos \theta_j + (y - k_j) \sin \theta_j \right)$$

Donde a_j controla la curvatura, (h_j, k_j) es el pico de la parábola y θ_j , es el ángulo de rotación con respecto al eje x.(16)

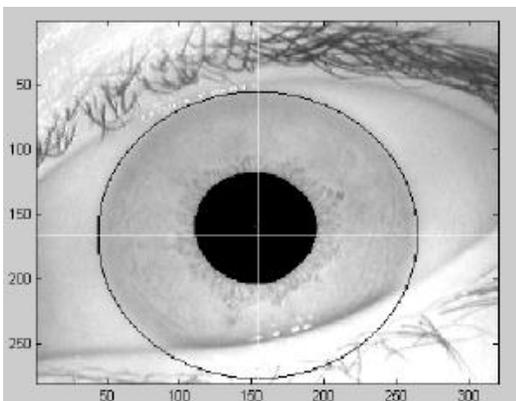


Figura 8: Iris segmentado a partir de la transformada de Hough.

1.4.2.2 Normalización.

Una vez que la región del iris es segmentado con éxito a partir de una imagen del ojo, la siguiente etapa es la de transformar la región del iris de modo que tenga dimensiones fijas con el fin de permitir las comparaciones. Las inconsistencias dimensionales entre las imágenes del ojo son principalmente debido al estiramiento del iris causado por dilatación de la pupila a partir de diferentes niveles de iluminación. Otras fuentes de inconsistencia incluyen, variando la distancia de la imagen, la rotación de la cámara, inclinación de la cabeza, y la rotación del ojo dentro de la cuenca del ojo. En el proceso de normalización se producen regiones del iris, que tienen las mismas dimensiones constantes, de modo que dos fotografías del mismo iris bajo condiciones diferentes tienen rasgos característicos en la misma ubicación espacial. (21)

Otro punto a destacar es que la región de la pupila no siempre es concéntrica dentro de la región del iris, y es por lo general ligeramente nasal (16). Esto debe tenerse en cuenta si se trata de normalizar la región del iris en forma de 'donut' debe tener radio constante.

1.4.2.2.1 Normalización usando Modelo de hoja de goma.

El modelo de hoja de goma homogéneo ideado por Daugman (22) vuelve a asignar a cada punto dentro de la región del iris para un par de coordenadas polares (r, θ) donde r es en el intervalo $[0,1]$ y θ es el ángulo de $[0,2\pi]$.

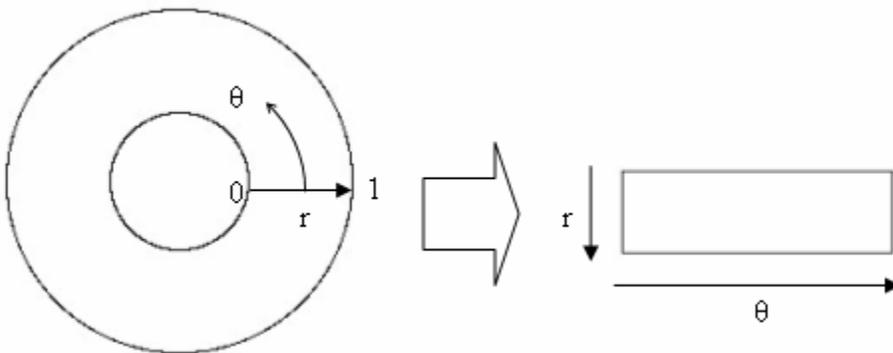


Figura 9: Modelo Daugman de hoja de goma.

La reasignación de la región del iris de (x, y) las coordenadas cartesianas a la normalizada no concéntrica representación polar se modela como:

$$I(x(r, \theta), y(r, \theta)) \rightarrow I(r, \theta)$$

Con

$$x(r, \theta) = (1 - r)x_p(\theta) + rx_l(\theta)$$

$$y(r, \theta) = (1 - r)y_p(\theta) + ry_l(\theta)$$

Donde $I(x,y)$ es la imagen del iris región, (x,y) son las coordenadas cartesianas originales, (r, θ) son las correspondientes coordenadas polares normalizadas, y x_p, y_p y x_l, y_l son las coordenadas de la pupila y el iris límite a lo largo de la θ dirección. El modelo de hoja de goma tiene en cuenta la dilatación de la pupila y las incoherencias de tamaño con el fin de producir una representación normalizada con dimensiones constantes. De esta manera, la región del iris se modela como una lámina de caucho flexible asegurado en el límite iris con el centro de la pupila como el punto de referencia (ver Figura 13). (16)

A pesar de que las cuentas homogéneas del modelo de hoja de goma de dilatación de la pupila, la distancia imagen y no concéntrica del desplazamiento, no se compensa las variaciones de rotación. En el sistema de Daugman, la rotación se tiene en cuenta durante la coincidencia por desplazamiento de las plantillas del iris en la dirección θ hasta que dos plantillas de iris están alineados. (16)

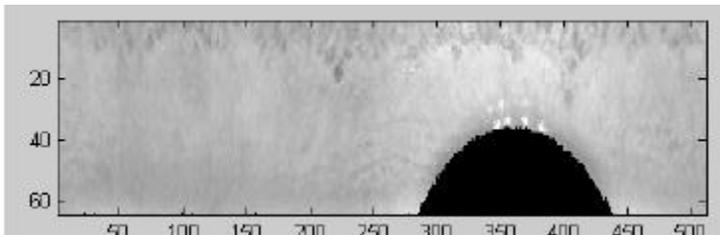


Figura 10: Iris normalizado.

1.4.3 Extracción de características.

Con el fin de proporcionar un reconocimiento preciso de los individuos, la información de mayor interés presente en un patrón de iris debe ser extraída. Sólo las características significativas del iris deben ser codificadas de manera que las comparaciones entre las plantillas se puedan hacer. La mayoría de los sistemas de reconocimiento del iris hacen uso de una descomposición de paso de banda de la imagen del iris para crear una plantilla biométrica. (16)

La plantilla que se genera en el proceso de codificación de las características también tendrá una métrica de coincidencia correspondiente, que da una medida de similitud entre dos plantillas de iris. Esta métrica debería dar un rango de valores al comparar plantillas generados a partir del mismo ojo, conocido como comparaciones intra-clase, y otro rango de valores al comparar plantillas creadas a partir de los

diafragmas diferentes, conocidos como comparaciones entre-clases. Estos dos casos deben dar valores distintos y separados, de modo que una decisión puede ser tomada con alta confianza en cuanto a si dos plantillas son del mismo iris, o a partir de dos diafragmas diferentes.

1.4.3.1 Filtros de Gabor.

Los filtros de Gabor son capaces de proporcionar la representación conjunta óptima de una señal en el espacio y la frecuencia espacial. Un filtro de Gabor se construye mediante la modulación de una onda de seno / coseno con una gaussiana. Este es capaz de proporcionar la localización conjunta óptima tanto en el espacio y la frecuencia, ya que una onda sinusoidal está perfectamente localizados en frecuencia, pero no localizados en el espacio. La modulación de la sinusoidal con una gaussiana proporciona localización en el espacio, aunque con una pérdida de la localización en frecuencia. La descomposición de una señal se lleva a cabo utilizando un par de cuadratura de filtros de Gabor, con una parte real especificado por una modulada por una gaussiana coseno, y una parte imaginaria especificado por un seno modulado por una gaussiana. Los filtros reales e imaginarios son conocidos también como los componentes simétricos incluso simétricos e impares, respectivamente. (23)

La frecuencia central del filtro es especificado por la frecuencia de la onda de seno / coseno, y el ancho de banda del filtro se especifica por la anchura de la gaussiana.

Daugman hace uso de una versión en de los filtros de Gabor en 2D (22) con el fin de codificar datos del iris del patrón. Los filtros de Gabor en 2D sobre el dominio de la imagen con una (x, y) se representa como:

$$G(x, y) = e^{-\pi[(x-x_0)^2/\alpha^2+(y-y_0)^2/\beta^2]} e^{-2\pi[u_0(x-x_0)+v_0(y-y_0)]}$$

Donde (x_0, y_0) especifica la posición de la imagen, (α, β) especifica el ancho y la longitud efectiva, y (u_0, v_0) especifica la modulación, que tiene una frecuencia espacial $\omega_0 = \sqrt{u_0^2 + v_0^2}$.

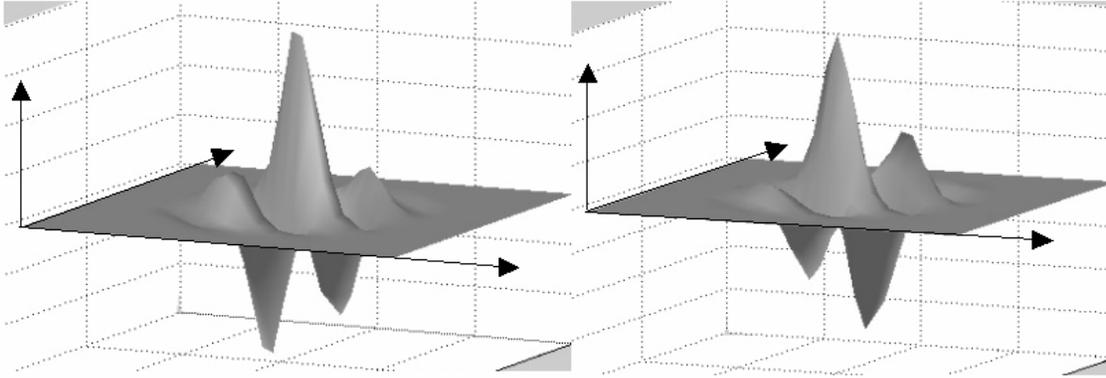


Figura 11: Un par de cuadratura de filtros de Gabor 2D izquierda) componente real o incluso filtro simétrico caracterizado por un coseno modulado por una gaussiana; derecho) componente imaginario o filtro simétrico raro caracterizado por una condición seno modulada por una gaussiana.

Daugman demodula la salida de los filtros de Gabor con el fin de comprimir los datos. Esto se hace por cuantificación de la información de fase en cuatro niveles, para cada cuadrante posible en el plano complejo. Se ha demostrado por Oppenheim y Lim (24) que la información de fase, en lugar de la información de amplitud proporciona la información más significativa dentro de una imagen. Tomando sólo la fase permitirá la codificación de la información discriminante en el iris, mientras que el descarte de información redundante como la iluminación, que está representado por la componente de amplitud.

Estos cuatro niveles se representan mediante dos bits de datos, por lo que cada píxel en el patrón de iris normalizado corresponde a dos bits de datos en la plantilla de iris. Un total de 2.048 bits se calcula para la plantilla, y un número igual de bits de enmascaramiento se generan con el fin de enmascarar las regiones dañadas dentro del iris. Esto crea un compacto de 256 byte plantilla, que permite el almacenamiento eficiente y la comparación de iris. El sistema Daugman hace uso de coordenadas polares para la normalización, tanto en forma polar los filtros se dan como:

$$H(r, \theta) = \epsilon^{-i\omega(\theta-\theta_0)} \epsilon^{-(r-r_0)^2/\alpha^2} \epsilon^{-i(\theta-\theta_0)^2/\beta^2}$$

Donde (α, β) son los mismos que en la ecuación y 1.5 (r_0, θ_0) especifica la frecuencia central del filtro.

La demodulación y proceso de cuantificación de fase se puede representar como:

$$h_{\{Re,Im\}} = sgn_{\{Re,Im\}} \int_p \int_{\phi} I(p, \phi) \epsilon^{-i\omega(\theta_0-\phi)} \epsilon^{-(r_0-p)^2/\alpha^2} \epsilon^{-(\theta_0-\phi)^2/\beta^2} p dp d\phi$$

Donde $h_{\{Re,Im\}}$ puede ser considerado como un bit cuyo valor complejo real e imaginario dependen del signo de la integral 2D, y $I(p, \theta)$ es la cruda imagen del iris en un sistema de coordenadas polares adimensional. (25)

1.4.4 Comparación de plantillas.

1.4.4.1 Distancia de Hamming.

Para la comparación entre plantillas la métrica utilizada es distancia de Hamming que da una medida de la cantidad de bits que son los mismos entre dos patrones de bits. La distancia de Hamming incorpora el enmascaramiento de ruido, de modo que sólo los bits sin ruido se utilizan en el cálculo. El uso de la distancia de Hamming de dos patrones de bits, es una decisión que puede ser tomada en cuanto a si los dos patrones se generaron a partir de los diafragmas diferentes o del mismo.

Al comparar el bit patrones X e Y, la distancia Hamming (HD), se define como la suma de los bits en desacuerdo (suma de la O exclusiva entre X e Y) sobre N, el número total de bits en el patrón de bits.

$$HD = \frac{1}{N} \sum_{j=1}^N x_j (XOR) y_j$$

Dado que la región del iris de un individuo contiene características con altos grados de libertad, cada región iris producirá un patrón de bits que es independiente a la producida por otro iris, por otra parte, dos códigos de iris producido a partir del mismo iris son altamente correlacionados.

Con el fin de dar cuenta de las incoherencias rotacionales, cuando se calcula la distancia de Hamming de dos modelos, una plantillase desplaza a nivel de bit a izquierda y derecha calculándose una serie de valores de la distancia de Hamming a partir de sucesivos cambios.

Si los patrones de dos bits son completamente independientes, tales como plantillas de iris generados a partir de los diafragmas diferentes, la distancia de Hamming entre los dos patrones debe ser igual a 0,5. Esto ocurre porque la independencia implica que los dos patrones de bits serán totalmente aleatorios, por lo que hay 0,5 oportunidades de establecer cualquier bit a 1, y viceversa. Por lo tanto, la mitad de los bits estarán de acuerdo y la otra mitad no estarán de acuerdo entre los dos patrones. Si dos patrones se derivan del mismo iris, la distancia de Hamming entre ellos estará cerca de 0,0, ya que están altamente correlacionados y los bits deberían ponerse de acuerdo entre los dos códigos de iris.

La distancia de Hamming es la métrica correspondiente empleado por Daugman, y el cálculo de la distancia de Hamming se presenta solo con los bits que se generan a partir de la región iris real. (26) (27)

1.5 Estandarización y biometría.

En los últimos años se ha notado una preocupación creciente por las organizaciones reguladoras respecto a elaborar estándares relativos al uso de técnicas biométricas en el ambiente informático. Esta preocupación es reflejo del creciente interés industrial por este ámbito tecnológico, y a los múltiples beneficios que su uso aporta. No obstante a ello, aún la estandarización continua siendo deficiente y como resultado de ello, los proveedores de soluciones biométricas continúan suministrando interfaces de software propietarios para sus productos, lo que dificulta a las empresas el cambio de producto o vendedor.

Los estándares utilizados para el funcionamiento del sistema son: estándar BioAPI 2.0, el estándar ISO/IEC 19794-6:2005 y ANSI / INCITS 358. Que cumplen con las características generales de los estándares, especificando a su vez una serie de características propias como son:

- ✓ Formato para el intercambio de datos biométricos.
- ✓ Un formato común de archivo que proporciona independencia de la plataforma utilizada, así como una clara separación entre la lógica de procesamiento y la de presentación.
- ✓ Aplicación de interfaces de programación y de perfiles.
- ✓ Definiciones de medida y cálculo del rendimiento.
- ✓ Métodos para evaluar el rendimiento.
- ✓ Requisitos para realizar informes de resultados. (28)

1.5.1 Estándar BioAPI 2.0.

Estandariza el modo en el que las aplicaciones se comunican con los dispositivos biométricos y la forma en la que los datos son almacenados y utilizados, ofreciendo a los desarrolladores un conjunto común de llamadas a funciones para interactuar modularmente con los distintos dispositivos biométricos y algoritmos.

Las funciones de BioAPI cubren aspectos como el entrenamiento, la verificación e identificación de usuarios, la captura de datos, el proceso de los mismos, la comparación de patrones y el almacenamiento de la información biométrica. Establece un alto nivel de abstracción que permite a los desarrolladores

olvidarse de los detalles particulares de fabricación de los distintos productos y de las tecnologías empleadas por los diferentes fabricantes.

La implementación de referencia es un paquete de *software* que contiene la instanciación del Framework de BioAPI. El Framework gestiona el mantenimiento de los BSP y la redirección de las llamadas de las funciones del API. Los *Biometric Service Provider* (BSP) encapsulan el hardware o software que implementen los BioAPI *Units* (abstracción de un dispositivo biométrico) para posteriormente presentarlas a la aplicación que las requiera. (29) (30)

1.5.2 Estándar ISO/IEC 19794-6:2005.

Define un formato de intercambio de datos biométricos bajo la denominación de tecnología de la Información – formatos de intercambio de datos biométricos. Con este proyecto se consigue la interoperabilidad de sistemas, de forma que si se modifica algún dispositivo, este se puede sustituir por otro que cumpla el formato.

Utiliza una especificación de la imagen con los pasos específicos del proceso previo y de la segmentación para la imagen, que puede ser cruda o comprimida; contiene solamente la información del iris y es mucho más compacto que la primera. Estos estándares también definen las estructuras y los encabezados de datos para apoyar el almacenaje de la información interoperable y proporcionan interoperabilidad entre proveedores proporcionando un método compacto de representación del iris humano. (30)

1.5.3 Estándar ANSI / INCITS 358.

Estándar creado en 2002 por ANSI y BioAPI *Consortium*, formato de intercambio de imágenes del iris. Presenta una interfaz de programación de aplicación que garantiza que los productos y sistemas que cumplen este estándar son interoperables entre sí. (31)

1.6 Metodologías, lenguajes y herramientas de desarrollo.

El entorno de desarrollo, metodología, lenguaje de programación y modelado, así como herramienta CASE, definido para la construcción del componente, fue producto de un estudio realizado por el equipo de desarrollo. Se brindará una breve descripción de cada herramienta a utilizar.

1.6.1 Metodologías de desarrollo de software.

Todo desarrollo de software es riesgoso y difícil de controlar, siéndolo aún más, cuando no se siga una guía que pauté los pasos a seguir y muestre cómo actuar ante las dificultades. Este papel lo juegan las metodologías de desarrollo de software. Sin ellas, el resultado de un proyecto sería desastroso, con

clientes insatisfechos con el resultado, malas planificaciones con fechas límite imposibles de cumplir y desarrolladores aún más insatisfechos.

Dada la variedad de metodologías que existe en la actualidad y la clasificación de las mismas, son muchos los factores a tener en cuenta a la hora de seleccionar una metodología para que esta se ajuste al ambiente de desarrollo y ayude, en vez de entorpecer, a obtener los resultados esperados.

Basándose en este planteamiento, se hizo un estudio de las mismas, para encontrar cuál se adapta mejor a una aplicación de pequeña escala con poco personal y requisitos cambiantes. Para ello se analizaron las metodologías RUP (Proceso Racional Unificado) y XP (Programación eXtrema).

1.6.1.1 RUP.

La metodología de desarrollo RUP forma parte de las llamadas metodologías pesadas. Entre sus principales características figuran que cuenta con una forma disciplinada de asignar tareas y responsabilidades, pretende implementar mejores prácticas, propone un desarrollo iterativo y el uso de la arquitectura basada en componentes además de un riguroso control de cambios, se basa en el modelado visual y le da gran importancia a la verificación de la calidad del software. El proceso de ciclo de vida de RUP se divide en cuatro fases llamadas Inicio, Elaboración, Construcción y Transición. Estas fases se dividen en iteraciones, cada una de las cuales produce una pieza de software demostrable Figura 7. (27)(33)

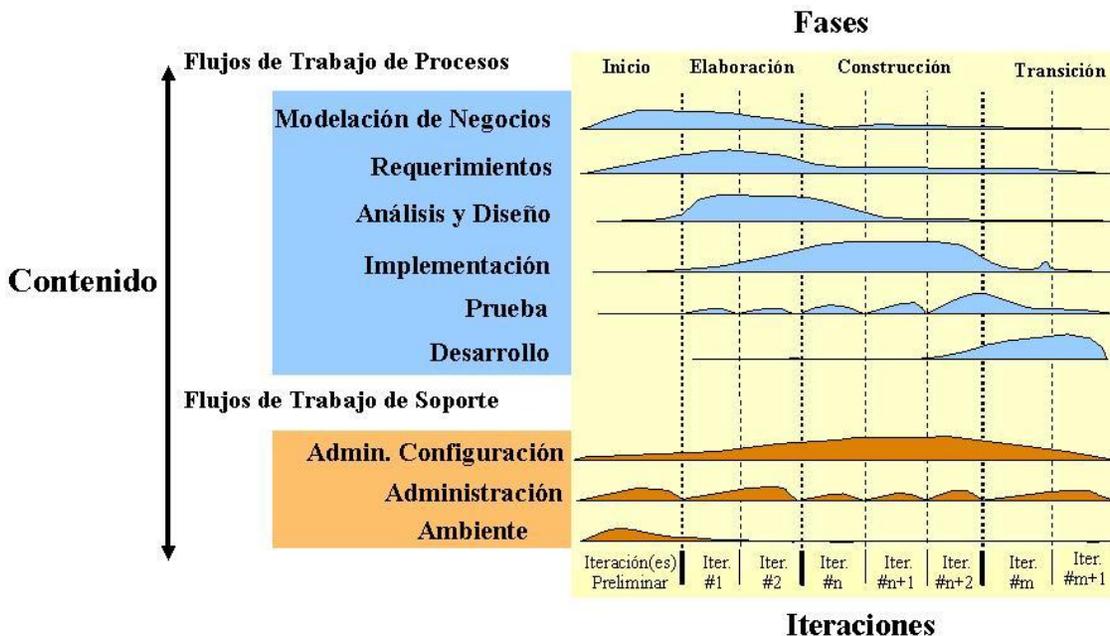


Figura 12: Metodología RUP.

RUP es una metodología poco factible para ser utilizada en proyectos pequeños, pues puede que no sea posible cubrir los costos de dedicación del equipo de desarrollo, ya que tendrían que invertir una gran cantidad de horas y esfuerzo para hacer las cosas según lo que dicen los procesos.

1.6.1.2 XP.

La programación extrema (XP) se basa en la simplicidad, la comunicación y el reciclado continuo de código. Promueve los valores de comunicación directa entre las personas; el coraje de exponer sus dudas, miedos, experiencias sin "embellecer" estas de ninguna de las maneras; el intento de mantener el software lo más simple posible y la capacidad de respuesta ante los cambios que se van haciendo necesarios a lo largo del camino. Los objetivos de XP son muy simples: la satisfacción del cliente. Esta metodología trata de dar al cliente el software que él necesita y cuando lo necesita. Por tanto, se debe responder muy rápido a las necesidades del cliente, incluso cuando los cambios sean al final del ciclo de la programación. El segundo objetivo es potenciar al máximo el trabajo en grupo. Tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software. (32) (33)

1.6.1.3 RUP vs XP.

Para la selección de la metodología de desarrollo a utilizar, se hace necesario analizar el entorno en el que se construirá el producto.

En primer lugar, es necesario destacar que los usuarios finales son los propios desarrolladores del departamento de Biometría, otro de los factores es que los requisitos asociados a la comparación de características pueden sufrir cambios debido al propio proceso de desarrollo del departamento de Biometría, lo cual arrojará nuevas necesidades y por ende nuevas funcionalidades a incluir en el componente.

Además, se hace necesario una solución rápida, pues el componente que se propone formará parte de los cimientos del sistema de identificación del departamento de Biometría y por lo tanto deberá estar funcional desde el momento en que dicho proyecto inicie su proceso de desarrollo.

En adición a lo antes descrito se cuenta con poco personal de desarrollo, lo cual dificulta la adopción de roles en cada etapa así como la generación de múltiples artefactos.

Se adoptaron los valores de XP como comunicación, sencillez, retroalimentación, coraje y respeto. La comunicación entre el equipo de desarrollo y los clientes será maximizada y efectuada de forma directa e interpersonal con el fin de evitar los problemas y errores causados por la mala adopción de la misma. Se

realizará un trabajo tan simple como sea posible, favoreciendo la comunicación, reduciendo el código sin utilizar y garantizando la calidad. Se estará constantemente midiendo el sistema para conocer cuánto se acerca a las funcionalidades necesarias mediante pruebas por parte de los usuarios. Los valores antes mencionados contribuirán a enfrentar con coraje cada cambio en los requisitos y refactorizaciones al sistema.

Lo antes descrito evidencia cómo las características del componente a desarrollar se inclinan a adoptar las prácticas y valores de la metodología XP, por lo cual se seleccionó la misma como rectora del proceso de desarrollo del software.

1.6.2 Lenguaje de modelado.

El Lenguaje Unificado de Modelado (UML por sus siglas en inglés), es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema que involucra una gran cantidad de software. Describe lo que supuestamente hará el sistema, pero no dice cómo implementarlo. Las herramientas CASE de modelado con UML permiten representar el software mediante diagramas que se generan durante las diferentes etapas del proyecto.(34)(35)

1.6.3 Herramientas de modelado.

1.6.3.1 Herramienta CASE.

Dentro de las herramientas claves en el desarrollo de aplicaciones informáticas se encuentran las herramientas de Ingeniería de Software Asistida por Ordenador (CASE) por sus siglas en inglés, las cuales son las encargadas de ayudar en el ciclo de desarrollo, con el fin de aumentar la productividad y reduciendo el costo en términos de tiempo y dinero. En el ciclo de desarrollo pueden ayudar en el proceso de diseño del proyecto, en el cálculo de costes, pueden implementar una parte del código, compilación automática y documentación. En el presente trabajo se utilizó Visual Paradigm para UML.(36)

1.6.3.1.1 Visual Paradigm para UML.

Una de las herramientas CASE más usadas es la suite creada por *Visual Paradigm International* (VPI). VPI es un proveedor de soluciones informáticas que incluye organizaciones para desarrollar aplicaciones de calidad, rápidas y baratas. Está compuesta por productos que facilitan a las organizaciones la visualización y diseño de diagramas. Sus soluciones se enfocan en eliminar la complejidad, aumentando así la productividad y disminuyendo el tiempo de desarrollo de las aplicaciones informáticas.

Aunque Visual Paradigm para UML (VPI) en su versión 8.0 *Enterprise Edition*. Es una herramienta muy completa y permite la representación de varios artefactos, no se aprovechan todas sus capacidades entre

las que se encuentra la generación de código debido a que no genera código para el lenguaje de programación M o lenguaje de programación MatLab como suele llamarse en algunas bibliografías, el cual será utilizado para la implementación del componente. Por tal motivo VPI se utiliza para la realización del Diagrama de componentes que sirve para un mejor entendimiento del funcionamiento y relación entre los archivos del componente desarrollado.(36)(37)

1.6.4 Lenguaje de programación.

Un lenguaje de programación es un conjunto de símbolos, reglas sintácticas y semánticas que definen su estructura, así como el significado de sus elementos y expresiones. Es utilizado para controlar el comportamiento físico y lógico de una máquina. En el presente trabajo se utilizó el lenguaje MatLab.

1.6.4.1 Lenguaje MatLab.

El lenguaje MatLab es un lenguaje de alto nivel para matrices con sentencias para el control de flujo, creación de funciones y estructuras de datos, funciones de entrada/salida y algunas características de programación orientada a objetos. Este lenguaje permite tanto la programación a pequeña escala para la creación rápida de programas, como programación a larga escala para la realización de aplicaciones complejas.(38)

Como lenguaje es muy sencillo basado en vectores, arreglos y matrices, esto hace que la expresión de coeficientes de filtros y demoras de buffers sean muy simples de expresar y comprender. En los últimos años se ha añadido la extensión para programación orientada a objetos y el diseño de interfaces gráficas. Este lenguaje es propio de la herramienta MatLab.(38)

1.6.5 Entorno Integrado de Desarrollo.

Un Entorno Integrado de Desarrollo (IDE), por sus siglas en inglés, 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.

Teniendo en cuenta que MatLab es un programa interactivo para computación numérica y visualización de datos y está basado en un sofisticado software de matrices para el análisis de sistemas de ecuaciones, orientado para llevar a cabo proyectos en donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos, se considera como IDE a utilizar para la implementación del componente.(39)

MatLab integra análisis numérico, cálculo matricial, proceso de señal y visualización gráfica en un entorno completo donde los problemas y sus soluciones son expresados del mismo modo en que se escribirían tradicionalmente, sin necesidad de hacer uso de la programación adicional.(39)

El nombre de MatLab proviene de la construcción de los términos *Matrix Laboratory* y fue inicialmente concebido para proporcionar fácil acceso a las librerías *Linpack* y *Eispack*, las cuales representan hoy en día dos de las librerías más importantes en computación y cálculo matricial.(39)

MatLab es un sistema de trabajo interactivo cuyo elemento básico de trabajo son las matrices. El programa permite realizar de modo rápido la resolución numérica de problemas en un tiempo mucho menor que si se quisiesen resolver estos mismos problemas con lenguajes de programación.

1.7 Conclusiones parciales.

En el presente capítulo se plantearon conceptos fundamentales para lograr una mejor comprensión de la problemática, así como las temáticas que permiten el desarrollo de la investigación. Se realizó una descripción del proceso de comparación de iris, así como de los algoritmos y estándares utilizados. Se describe la metodología que regirá el desarrollo del componente de comparación de características de iris y se abordaron las tecnologías y herramientas seleccionadas para el desarrollo del componente.

Con el estudio realizado se sentaron las bases que permitirán la agilización del desarrollo del componente de comparación de características en iris seleccionando la herramienta CASE Visual Paradigm, el IDE MatLab R2012a. Donde XP será de suma importancia para guiar el proceso de desarrollo del componente respectivamente.

Capítulo II: Análisis y diseño.

2.1 Introducción.

La solución a desarrollar, debe ser fruto de un correcto análisis y una amplia comprensión de todos los elementos que se relacionan con el tema de la comparación de características en imágenes de iris, profundizándose en el estudio de las características que posibilitan desarrollar el mismo.

El objetivo que se persigue con la elaboración de este capítulo es mostrar la evolución de la solución durante las fases iniciales de Exploración y Planificación, además de presentar los diferentes artefactos generados en las mismas, los cuales serán premisas cruciales para la entrega final del componente.

En este capítulo se interpretan las necesidades del sistema especificándolas mediante historias de usuario y requisitos no funcionales. Se expone la estimación de cada una de ellas, se define el tiempo que va a demorar cada iteración que permitirá la realización del componente, así como también se da a conocer el plan de entrega del componente.

2.2 Exploración.

Para comenzar el desarrollo de un producto de software es necesario asegurarse de que es posible hacerlo. Se debe tener confianza en que las herramientas seleccionadas ayudarán a la culminación del trabajo. Se debe creer que una vez que el código se haga, este puede utilizarse cada día. Cada miembro del equipo debe confiar en sus propias habilidades y en las habilidades de los otros.

La fase de exploración ayuda a resolver todos estos conflictos. Es en ella donde los clientes plantean a grandes rasgos las historias de usuario que son de interés para el producto. Durante la misma, los desarrolladores interactúan con las herramientas a utilizar durante todo el proceso, exploran activamente las posibilidades de arquitectura y experimentan los límites de las tecnologías a utilizar. (40)

Para dar cumplimiento a los objetivos trazados y asumir todos los objetivos planteados al inicio de esta investigación, la realización del análisis, diseño e implementación de componentes y utilitarios para la comparación de características de iris, deben permitir la integración a cualquier sistema de esta índole. El logro de ese producto garantizará un mejor rendimiento en los procesos de comparación de características de iris.

2.2.1 Descripción del sistema.

Para el funcionamiento del componente el usuario debe indicar el directorio donde se encuentra la base de datos de imágenes de prueba a las que se les extraerá el vector de características para su

comparación posteriormente. Si las imágenes cargadas son en colores, deben ser llevadas a escala de grises para un mejor procesamiento de las mismas. En caso de que estén procesadas no es necesario llevarlas al nivel de grises.

Culminado este proceso las imágenes están listas para aplicarles la segmentación y la normalización. Una vez obtenidos los vectores de características invariantes de cada imagen se procede aplicarles el algoritmo de Distancia de Hamming para comparar los vectores, ya que este obtiene de cada imagen solo los principales componentes, procedimiento que reduce el costo computacional en la próxima etapa que es la de clasificación.

Conseguido el vector de características de todas las imágenes, se pueden exportar a un fichero .mat para su posterior análisis.

A partir de una breve descripción de los procesos del negocio expuestos, se puede observar que el mismo tiene un bajo nivel de estructuración. Por lo que esta investigación se basará en un modelo de dominio, que permitirá mostrar al usuario, los principales conceptos que se operan en el dominio de la aplicación en desarrollo. De esta forma los usuarios utilizarán un vocabulario común, para lograr entender el contenido en que se enmarca la aplicación. Este modelo va a contribuir más adelante a describir las clases más importantes dentro del contexto la aplicación.

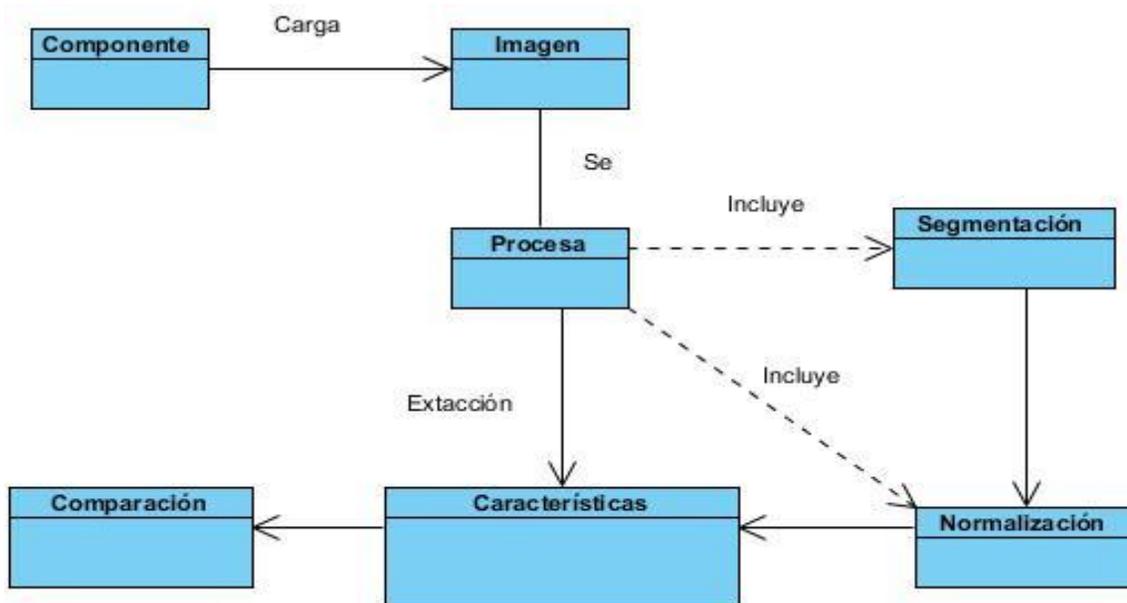


Figura 13: Modelo de Dominio.

2.2.2 Arquitectura de software.

El patrón de arquitectura propuesto para estructurar el componente de comparación de características de iris es el de Tuberías y Filtros, que por lo general es apropiada para sistemas que implementan transformaciones de datos en pasos sucesivos.

La estructura que provee para procesar flujos de datos la arquitectura Tuberías y Filtros, es que cada paso de procesamiento se encapsula en un filtro, donde los datos se pasan usando los tubos entre filtros adyacentes. Recombinado los filtros se pueden construir distintas familias de sistemas relacionados. Para el componente, el flujo de datos se comporta de la siguiente manera con la entrada de una imagen. La misma pasará por un filtro donde será segmentada mediante el componente `segmentiris.m`, luego esa imagen segmentada será normalizada usando el componente `normaliseiris.m`, donde por último será procesada en los componentes `shiftbits.m` y `gethammingdistance.m` para la extracción de las características y la comparación respectivamente.

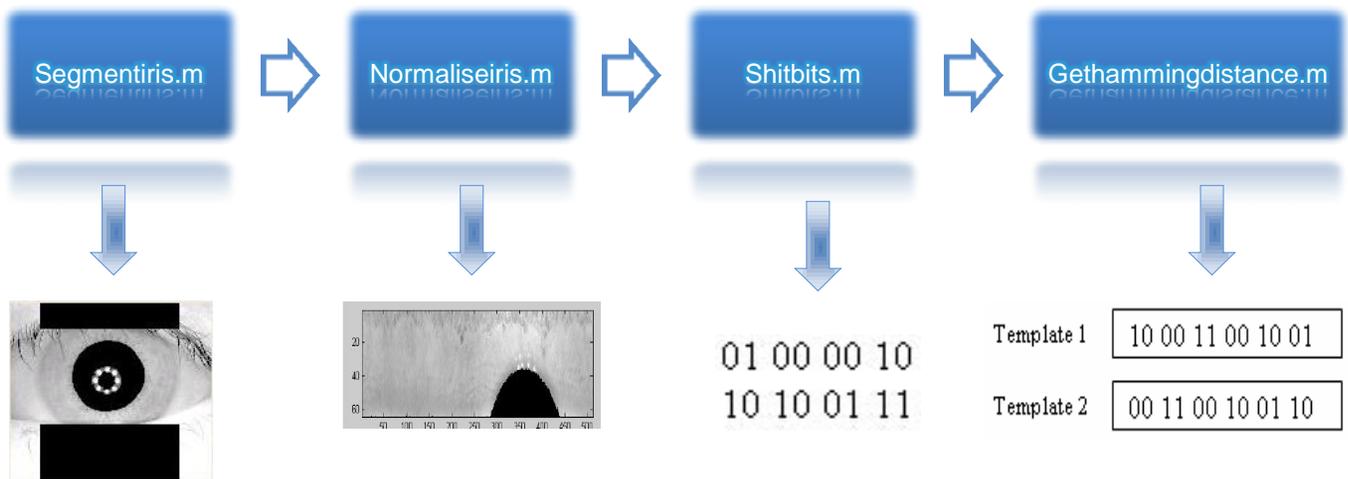


Figura 14: Arquitectura de software.

2.2.3 Usuarios del sistema.

Los actores del sistema no son parte de él. Pueden intercambiar información con él. Pueden ser un recipiente pasivo de información. Pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado.

| Actor del sistema | Descripción |
|-------------------|--|
| Desarrollador | Representa a la persona que interactúa con el componente para cargar |

las imágenes a procesar y extraer los vectores característicos de las mismas.

2.2.4 Historias de usuarios.

Las historias de usuario (HU) representan las funcionalidades que el cliente desea que estén presentes en el sistema; por lo tanto, todo el trabajo futuro debe girar en torno a satisfacer estas expectativas. Las historias de usuario son la primera y fundamental entrada para el proceso de desarrollo de software basado en la metodología XP.(41) La descripción de las mismas se muestra a continuación:

Tabla 3: Historia de usuario Transformar imagen a escala de grises.

| Historia de usuario | |
|---|---|
| Número: HU1 | Nombre de historia de usuario: Transformar imagen a escala de grises |
| Modificación de historia de usuario: Ninguna | |
| Usuario: Nathaly Rodríguez Guillén, Mariena Ferrer Barrientos. | Interacción asignada: 1 |
| Prioridad en Negocio: Alta. | Puntos Estimados: 1 |
| Riesgo en Desarrollo: Alta. | Puntos Reales: 1 |
| Descripción: En caso de que las imágenes a analizar sean en color, se debe transformar la imagen a escala de grises para un mejor procesamiento. El procedimiento consiste en transformar los colores en tonalidades de grises que van desde el negro. | |
| Observaciones: En caso de que las imágenes a analizar sean procesadas, no es necesario transformar la imagen a escala de grises. | |

Tabla 4: Historia de usuario Segmentar imagen de iris.

| Historia de usuario | |
|---------------------|---|
| Número: HU2 | Nombre de historia de usuario: Segmentar imagen de iris. |

| | |
|---|--------------------------------|
| Modificación de historia de usuario: Ninguna | |
| Usuario: Nathaly Rodríguez Guillén, Mariena Ferrer Barrientos. | Interacción asignada: 1 |
| Prioridad en Negocio: Alta | Puntos Estimados: 1 |
| Riesgo en Desarrollo: Alta | Puntos Reales: 1 |
| Descripción: Se aplica un operador que sirve para encontrar dos bordes: el borde pupilar y el borde del iris, a través de un método iterativo denominado "de grueso a delgado". Luego de una manera similar para detectar bordes curvilíneos es usada para localizar los bordes de los párpados. | |
| Observaciones: El resultado de todas estas operaciones de localización es el aislamiento del iris de otras regiones de la imagen. | |

Tabla 5: Historia de usuario Normalizar imagen de iris.

| Historia de usuario | |
|---|--|
| Número: HU3 | Nombre de historia de usuario: Normalizar imagen de iris. |
| Modificación de historia de usuario: Ninguna | |
| Usuario: Nathaly Rodríguez Guillén, Mariena Ferrer Barrientos. | Interacción asignada: 1 |
| Prioridad en Negocio: Alta | Puntos Estimados: 1 |
| Riesgo en Desarrollo: Alta | Puntos Reales: 1 |
| Descripción: El proceso de normalización no es más que trasladar la imagen bidimensional de un espacio de coordenadas cartesianas a otro en coordenadas polares. En síntesis es una transformación que permite desenrollar en una cinta la textura que representa el tejido pigmentado de interés. | |
| Observaciones: | |

Tabla 6: Historia de usuario Extraer las características.

| Historia de usuario | |
|--|--|
| Número: HU4 | Nombre de historia de usuario: Extraer las características. |
| Modificación de historia de usuario: Ninguna | |
| Usuario: Nathaly Rodríguez Guillén, Mariena Ferrer Barrientos. | Interacción asignada: 2 |
| Prioridad en Negocio: Alta | Puntos Estimados: 1 |
| Riesgo en Desarrollo: Alta | Puntos Reales: 1 |
| Descripción: Luego de tener el vector característico de cada imagen se le aplica el algoritmo de Daugman para obtener una plantilla con la cual se harán las comparaciones. | |
| Observaciones: | |

Tabla 7: Historia de usuario Exportar vector característico.

| Historia de usuario | |
|---|--|
| Numero: HU 5 | Nombre de historia de usuario: Exportar el vector característico. |
| Modificación de historia de usuario: Ninguna | |
| Usuario: Nathaly Rodríguez Guillén, Mariena Ferrer Barrientos | Iteración asignada: 2 |
| Prioridad en Negocio: Alta | Puntos estimados: 1 |
| Riesgo en desarrollo: Medio | Puntos reales: 1 |
| Descripción: Luego de obtener los vectores de características, estos se pueden guardar en un fichero .mat para su posterior análisis. Se debe indicar la dirección específica donde los desea guardar. | |
| Observaciones: | |

Tabla 8: Historia de usuario calcular la distancia de Hamming.

| Historia de usuario | |
|---|--|
| Número: HU6 | Nombre de historia de usuario: Calcular la distancia de Hamming |
| Modificación de historia de usuario: Ninguna | |
| Usuario: Nathaly Rodríguez Guillén, Mariena Ferrer Barrientos. | Interacción asignada: 2 |
| Prioridad en Negocio: Alta | Puntos Estimados: 1 |
| Riesgo en Desarrollo: Alto | Puntos Reales: 1 |
| Descripción: Se halla la distancia de Hamming desde cada vector de características de las imágenes de la base de datos al vector de la imagen que se quiere reconocer. | |
| Observaciones: El método de la distancia de Hamming es utilizado en la fase de comparación pero se utiliza para comprobar que los vectores extraídos son adecuados para el reconocimiento. | |

2.2.5 Requisitos no funcionales.

Los requerimientos no funcionales son cualidades o propiedades que el sistema debe tener, definen propiedades emergentes del sistema, tales como el tiempo de respuesta, las necesidades de almacenamiento, la fiabilidad, pueden especificar también la utilización de una herramienta CASE en particular, un lenguaje de programación o un método del desarrollo. (42)

RNF1. Requerimientos en el diseño y la implementación.

Lenguaje de programación: MatLab.

IDE: MatLab R2012a.

Para el modelado de UML se utilizará: Visual Paradigm 8.0.

RNF2. Requerimientos de seguridad.

Confiablez: Sólo a los usuarios autorizados se les garantizarán el acceso a la información y que los dispositivos o mecanismos utilizados para lograr la seguridad no ocultarán o retrasarán a los usuarios para obtener los datos deseados en un momento dado.

Disponibilidad: El sistema propuesto debe ser capaz de adaptarse con el fin de alcanzar un equilibrio interno frente a los cambios externos del entorno.

RNF3. Software.

El sistema será usado en Sistema Operativo Windows XP o superior.

2.3 Planificación.

Se planea garantizar que siempre se esté haciendo lo más importante que queda por hacer, a fin de mantenerse coordinados de manera efectiva con otras personas y para responder rápidamente a eventos inesperados (40).

El propósito de la fase de planificación es establecer un acuerdo entre los clientes y desarrolladores sobre el menor tiempo en que la mayor cantidad de HU pueda ser realizada.

Para ello se realizó la planificación con el objetivo de maximizar el valor del software producido a partir de la puesta en producción de las características más importantes lo antes posible, se realiza una estimación del esfuerzo requerido para la implementación de las HU y se define el tiempo de entrega y cada iteración.

2.3.1 Estimación de esfuerzos por historias de usuarios.

A continuación se muestra la estimación del esfuerzo por cada HU propuesta para el desarrollo de la aplicación:

Tabla 9: Estimación de esfuerzos por Historias de usuarios.

| No. | Historia de usuario | Estimación (semanas) |
|-----|------------------------------------|----------------------|
| 1 | Transformar a escalas de grises. | 2 |
| 2 | Segmentar imagen de iris. | 4 |
| 3 | Normalizar imagen de iris. | 4 |
| 4 | Extraer las características. | 5 |
| 5 | Exportar el vector característico. | 2 |
| 6 | Calcular la distancia de Hamming. | 4 |

2.3.2 Plan de iteraciones.

Después de ser identificadas y descritas las HU y estimar el esfuerzo dedicado a la realización de cada una de ellas, se procede a la planificación de la fase de implementación estableciendo una división de dos iteraciones.

Iteración 1: En esta iteración se realizarán las historias de usuarios más complejas y que inciden críticamente en la lógica de la aplicación. Una vez concluida esta iteración el componente se encontrará en un estado plenamente funcional.

Iteración 2: En esta iteración se realizarán las historias de usuarios que son menos complejas y que no inciden críticamente en la lógica de la aplicación. Una vez concluida esta iteración el componente se encontrará completamente concluido y listo para su explotación.

2.3.3 Plan de duración de las iteraciones.

El plan de duración de las iteraciones se encarga de mostrar las HU en el orden en que se implementarán en cada iteración así como la duración estimada de las mismas.

Tabla 10: Estimación de esfuerzos por Historias de usuarios.

| Iteración | Código | Historias de usuarios | Duración estimada (semanas) |
|-------------|--------|------------------------------------|-----------------------------|
| Iteración 1 | HU 1 | Transformar a escalas de grises. | 10 |
| | HU 2 | Segmentar imagen de iris. | |
| | HU 3 | Normalizar imagen de iris. | |
| Iteración 2 | HU4 | Extraer las características. | 11 |
| | HU 5 | Exportar el vector característico. | |
| | HU 6 | Calcular la distancia de Hamming. | |

La iteración 1 tendrá una duración total de 2 meses y 2 semanas.

La Iteración 2 tendrá una duración total de 3 meses y 1 semana.

2.4 Conclusiones parciales.

En este capítulo se realizaron las diferentes HU, se definió la estimación de esfuerzo por HU permitiendo obtener un plan de duración y un plan de entrega con la realización del componente. La arquitectura de Tuberías y Filtros utilizada, brinda la posibilidad de ir transformando un flujo de datos en un proceso comprendido por varias fases secuenciales, siendo la entrada de cada una la salida de la anterior.

Capítulo III: Implementación y prueba.

3.1 Introducción.

La implementación de *software* en XP es un proceso que se realiza de forma iterativa, obteniendo como resultado un producto funcional que debe ser sometido a pruebas y mostrado al cliente para permitir una retroalimentación por parte de los desarrolladores. El siguiente capítulo está dedicado a detallar las 2 iteraciones llevadas a cabo durante la etapa de construcción del componente, exponiéndose cada una de las tareas designadas por HU. De igual modo se exponen las pruebas de aceptación efectuadas sobre el componente.

3.2 Fase de diseño.

Siguiendo las guías de la metodología XP, debe aparecer un diseño que describa qué clases hay y cómo interactúan. La metodología XP no requiere la representación del sistema mediante diagramas de clases utilizando el Lenguaje Unificado de Modelado (UML); en su lugar usa una técnica para representar clases, esta es la que usa tarjeta Clase-Responsabilidad-Colaboración (CRC). No obstante, el uso de diagramas UML puede aplicarse siempre y cuando influyan en el mejoramiento de la comunicación, no sea un peso su mantenimiento, no sean extensos y se enfoquen en la información importante. (43)

Programar en MatLab es usar una serie de comandos que permitan realizar una tarea o función específica. Estos pueden ser escritos uno por uno a través de la línea de comandos:

Archivos -M: Comandos y Funciones.

Los archivos que contienen instrucciones de MatLab se llaman archivos-M porque siempre tienen una extensión de ".m" como la última parte de su nombre de archivo. Un archivo -M consiste de una secuencia de instrucciones normales de MatLab, que probablemente incluyen referencias a otros archivos -M, se puede llamar recursivamente a sí mismo. Se pueden crear archivos-M utilizando un procesador de palabras o un editor de texto.

Existen dos tipos de archivos -M, de comandos y funciones. Los archivos de comandos, automatizan secuencias largas de comandos. Los archivos de funciones, permiten añadir a MatLab funciones adicionales expandiendo así la capacidad de este programa. Ambos, comandos y funciones, son archivos ordinarios de texto ASCII.

Archivo de Comandos.

Cuando un archivo de comandos es invocado, MatLab simplemente ejecuta los comandos encontrados en dicho archivo. Las instrucciones en un archivo de comando operan globalmente en los datos en el espacio de trabajo. Los comandos son utilizados para hacer análisis, resolver problemas, o diseñar secuencias largas de comandos que se conviertan en interactivas.

Archivo de Funciones.

Un archivo -M que contiene la palabra *function* al principio de la primera línea, es un archivo de función. En una función, a diferencia de un comando, se deben de pasar los argumentos. Las variables definidas y manipuladas dentro de la función son locales a esta y no operan globalmente en el espacio de trabajo. Los archivos de funciones se utilizan para extender a MatLab y crear nuevas funciones para MatLab utilizando el lenguaje propio de MatLab.

Con lo antes mencionado utilizando la metodología XP es innecesario usar tarjetas CRC si se desarrolla con MatLab. La no utilización de estas está dada a que la implementación en MatLab es mediante comandos y funciones y no emplea clases. El programa se implementa mediante la invocación de funciones.

Para un mejor entendimiento del funcionamiento y relación entre los archivos del componente desarrollado se emplea un diagrama de componentes.

Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Además, muestran la organización y las dependencias lógicas entre un conjunto de componentes de *software*. Para la construcción del componente se modeló el siguiente diagrama.

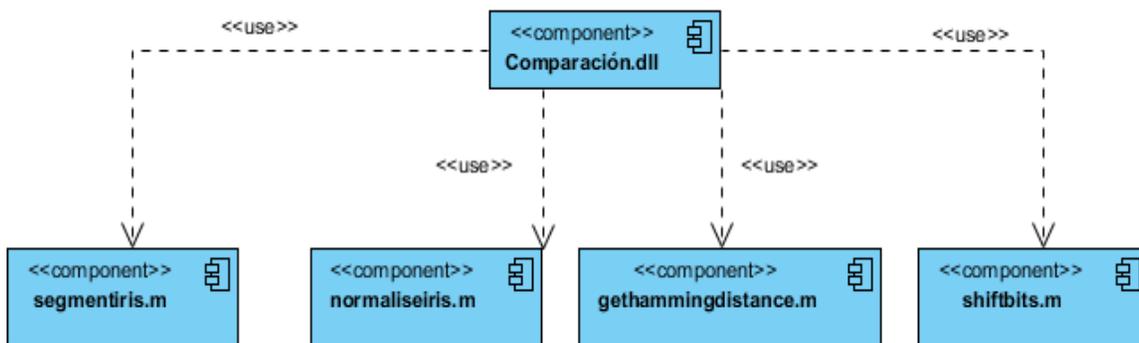


Figura 15: Diagrama de componente.

3.2.1 Funciones implementadas para el desarrollo del componente.

El componente desarrollado como parte de la solución de esta investigación es una Biblioteca de enlace dinámico, dll (*Dynamic Link Library* por sus siglas en inglés). Dicho componente cuenta con una serie de funcionalidades distribuidas en varios subcomponentes utilizando códigos desarrollados por Libor Masek. (44)

Subcomponente segmentiris.m contiene la implementación de la función:

segmentiris(eyeimage)

Esta función se encarga de la segmentación automática de la región del iris de una imagen del ojo. También aísla las áreas del ruido como los párpados y pestañas. Además de devolver las coordenadas del centro de iris y de la pupila.

Subcomponente normaliseiris.m contiene la implementación de la función:

normaliseiris(image, x_iris, y_iris, r_iris, x_pupil, y_pupil, r_pupil, eyeimage_filename, radpixels, angulardiv)

Esta función se encarga de realizar la normalización de la región del iris, es decir transforma de coordenadas cartesianas a coordenadas polares, la cual permite convertir la imagen segmentada de un área circular en un nuevo registro gráfico de forma rectangular, donde el iris se desarrolla en una cinta.

Subcomponente shiftbits.m contiene la implementación de la función:

shiftbits(template, noshifts, nscales)

Esta función se encarga de filtrar la imagen normalizada por medio de filtros de Gabor y se cuantifican sus cualidades a través del promedio de desviación absoluta, generando un conjunto de datos en forma de matriz con los patrones del iris.

Subcomponente guardarVector.m contiene la implementación de la función:

GuardarVector (LugarAGuardarVector, Template, Mask)

Esta función se encarga de guardar los vectores. El parámetro template y Mask son la plantilla y la máscara de ruido que el usuario deberá guardar. En el parámetro LugarAGuardarVector se debe especificar la ruta completa con el nombre seleccionado y la extensión, por ejemplo C:/Users/Administrador/Desktop/Procesadas/vectorBD.mat.

Subcomponente gethammingdistance.m contiene la implementación de la función:

gethammingdistance(template1, mask1, template2, mask2, scales)

Esta función devuelve la distancia entre dos plantillas generas a partir del vector característico, demostrando si se generan a partir del mismo diafragma o de diafragmas diferentes.

3.2.2 Estándar de programación.

Es necesario establecer un criterio fijo que proporcione reglas para la creación de nombre para variables y métodos del tal forma que ninguno de los integrantes del equipo dudase en ningún momento del nombre que debiera poner a un método o del nombre que tiene que utilizar para llamar a un método desarrollado por otros programadores del equipo.

Por esto es necesario establecer un estándar de codificación de tal forma que los programadores sigan los mismos criterios para desarrollar el código.

Es decisivo, para poder plantear con éxito la propiedad colectiva del código. Esta sería impensable sin una codificación basada en estándares que haga que todo el mundo se sienta cómodo con el código escrito por cualquier otro miembro del equipo (45).

Definición de ciclos

- ✓ Utilizar constantes con nombre en lugar de números, como por ejemplo en lugar de emplear *for i = 1 to 7* se debe utilizar *for i = 1 To Numerolimagenes* para que resulten fáciles de comprender.

Comentarios

- ✓ Encima de cada función debe ir un comentario que explique el propósito de la misma para un mejor entendimiento.
- ✓ Alinear todos los comentarios de final de línea en la misma posición de tabulación.

Sangría

- ✓ En las líneas de construcción lógica debe ser de 4 espacios, por ejemplo:

If expression1

statements1

Else if expression2

statements2

Else

statements3

End

Espacios

- ✓ Utilizar un espacio antes y después de cada operador.
- ✓ Usar un espacio luego de una coma.
- ✓ No usar espacios entre las variables o nombre de función y los corchetes o paréntesis.

Nombres de Funciones

- ✓ Todas las palabras deben comenzar con letra mayúscula y separadas por un guión bajo.

Nombre de Variables

- ✓ Todas las palabras deben comenzar con letras mayúsculas y unidas una de otras.

Nombre de Componentes visuales

- ✓ La primera palabra define el tipo de componente y comienza con minúscula, las siguientes van unidas a la primera y comienzan con letra mayúscula.

3.2.3 Tareas de ingeniería

XP plantea que la implementación de un *software* se hace iterativamente, obteniendo al culminar cada iteración un producto funcional, que debe ser probado y mostrado al cliente. Durante el transcurso de las iteraciones, se realiza la implementación de las HU definidas por el cliente y descritas por el equipo de desarrollo en la etapa de Exploración. Como parte de este plan, se descomponen estas HU en tareas de la ingeniería las cuales son asignadas a los programadores para ser implementadas durante la iteración correspondiente.

A continuación se describen las tareas correspondientes a las historias de usuario.

Tabla 11: HU1_Transformar imágenes a escala de grises.

Tarea de ingeniería

| | |
|--|--|
| Número tarea: 1 | Número historia de usuario: HU1 |
| Nombre tarea: Transformar imágenes a escala de grises. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 3/dic/2012 | Fecha fin: 15/dic/2012 |
| Programador responsable: Nathaly Rodríguez Guillén y Mariena Ferrer Barrientos | |
| Descripción: Desarrollar el proceso de transformar a escalas de grises las imágenes de la base de datos. | |

Tabla 12: HU2_Segmentar imagen de iris.

| | |
|---|--|
| Tarea de ingeniería | |
| Número tarea: 2 | Número historia de usuario: HU2 |
| Nombre tarea: Segmentar imagen de iris. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 13/ene/2013 | Fecha fin: 2/feb/2013 |
| Programador responsable: Nathaly Rodríguez Guillén y Mariena Ferrer Barrientos | |
| Descripción: Desarrollar el proceso de segmentar la imagen para encontrar los bordes de la pupila y del iris. | |

Tabla 13: HU3_Normalizar imagen de iris.

| | |
|---|--|
| Tarea de ingeniería | |
| Número tarea: 3 | Número historia de usuario: HU3 |
| Nombre tarea: Normalizar imagen de iris. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 4/feb/2013 | Fecha fin: 2/mar/2013 |
| Programador responsable: Nathaly Rodríguez Guillén y Mariena Ferrer Barrientos | |

Descripción:

Desarrollar el proceso de normalizar la imagen segmentada para encontrar definir valores fijos, es decir llevar la imagen de coordenadas polares a coordenadas cartesianas.

Tabla 14: HU4_Extraer características de iris.

| Tarea de ingeniería | |
|--|--|
| Número tarea: 4 | Número historia de usuario: HU4 |
| Nombre tarea: Extraer las características. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 4/mar/2013 | Fecha fin: 5/abr/2013 |
| Programador responsable: Nathaly Rodríguez Guillén y Mariena Ferrer Barrientos | |
| Descripción: Desarrollar el proceso de extraer las características de la imagen normalizada. | |

Tabla 15: HU5_Exportar vector característico.

| Tarea de ingeniería | |
|--|--|
| Número tarea: 5 | Número historia de usuario: HU5 |
| Nombre tarea: Exportar el vector característico. | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 8/abr/2013 | Fecha fin: 19/abr/2013 |
| Programador responsable: Nathaly Rodríguez Guillén y Mariena Ferrer Barrientos | |
| Descripción: Desarrollar la funcionalidad de guardar los vectores característicos en un fichero .mat para su posterior análisis. | |

Tabla 16: HU6_Calcular distancia de Hamming.

| Tarea de ingeniería | |
|------------------------|--|
| Número tarea: 6 | Número historia de usuario: HU6 |

| | |
|--|-------------------------------|
| Nombre tarea: Calcular la distancia de Hamming | |
| Tipo de tarea: Desarrollo | Puntos estimados: 1 |
| Fecha inicio: 29/abr/2013 | Fecha fin: 24/may/2013 |
| Programador responsable: Nathaly Rodríguez Guillén y Mariena Ferrer Barrientos | |
| Descripción: Desarrollar el proceso de comparación mediante la distancia de Hamming usando dos plantillas. | |

3.2.4 Requisitos para la integración del componente.

Implementados los algoritmos se decide crear el componente para la plataforma .NET de Microsoft. La misma se selecciona debido a la experiencia que tienen los desarrolladores del departamento de Biometría en el trabajo con esta tecnología, además de ser la definida para el trabajo en dicho departamento.

.NET brinda la posibilidad de elegir el lenguaje de programación de las aplicaciones entre un gran número que incluye lenguajes populares como *C#, C++, Visual Basic, Perl y Python*.

Para un correcto despliegue se debe tener en cuenta:

- ✓ La máquina donde se vaya a utilizar debe tener el mismo sistema operativo que la máquina donde se compiló la dll.
- ✓ Garantizar que el compilador de Matlab (*Matlab Compiler Runtime (MCR)*) esté instalado en la máquina, si el MCR no está instalado debe ejecutar MCRInstaller.exe. En caso de tener el Matlab instalado se puede localizar en:

C:\Program Files\MATLAB\R2012a\toolbox\compiler\deploy\win32\MCRInstaller.exe

El uso de archivos .dll ayuda en la reutilización de código, facilita el compartimiento de datos y recursos, un uso eficaz de la memoria y espacio en disco reducido. Por lo tanto, el sistema operativo y los programas se cargan más rápido, se ejecutan rápidamente y necesitan menos espacio de disco en el equipo.

Las librerías compiladas son imposibles modificarlas desde Visual Studio, pero se puede acceder a los métodos implementados como si se tratasen de una clase más del proyecto. Si se realizan cambios en el código fuente de los algoritmos se debe volver a generar la librería y acceder a la nueva versión de la misma desde el entorno de desarrollo.

Si se realizan cambios en el código fuente de los algoritmos se debe volver a generar la librería y acceder a la nueva versión de la misma desde el entorno de desarrollo.

Para más información acerca del proceso de instalación y el MCR, puede escribir en la ventana de comandos del MATLAB *"help MCR"* o *"help mcrinstaller"*.

3.3 Pruebas.

Las pruebas son acciones en las cuales el sistema es ejecutado bajo condiciones o requisitos determinados. Los resultados son chequeados y registrados. Las pruebas verifican los resultados de la implementación del sistema. El modelo de prueba indica cómo han de ser probados aspectos determinados del producto o *software*. Hay multitud de conceptos asociados a las tareas de prueba. Clasificarlas es difícil, pues no son mutuamente disjuntas, sino muy entrelazadas. En lo que sigue se intentará una pequeña organización de acuerdo con sus características.

3.3.1 Diseño de pruebas.

Un aspecto muy importante de rendimiento en los sistemas biométricos es la precisión del proceso de verificación o identificación. En la actualidad se consideran tres parámetros que ayudan a determinar la precisión del proceso de una manera cuantitativa:

- ✓ **Tasa de falso rechazo (FRR):** es el porcentaje de usuarios autorizados que tratan de acceder al sistema y este los declara como no autorizados.
- ✓ **Tasa de falsa aceptación (FAR):** es el porcentaje de intentos de accesos de usuarios no autorizados los cuáles el sistema acepta como autorizados.
- ✓ **Tasa de igualdad de error (ERR):** es el punto en el cuál FRR y FAR son el mismo valor.

3.3.2 Pruebas de aceptación.

Las pruebas de aceptación son creadas a partir de las HU. Durante una iteración la HU seleccionada en la planificación de iteraciones conllevará una o varias pruebas de aceptación. El cliente o usuario especifica los aspectos a testear cuando una HU ha sido correctamente implementada.

Una HU puede tener más de una prueba de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento y no se considera completa hasta que no supera sus pruebas de aceptación. Es responsabilidad del cliente verificar la corrección de las pruebas y tomar decisiones acerca de las mismas.

Para las pruebas se utilizaron imágenes de la base de datos CASIA (Ver Figura 18). Cada imagen del iris se encuentra en escala de grises, con una resolución de 320x380 píxeles. Las imágenes se pueden descargar de forma gratuita (46). Los archivos se encuentran en formato JPG (Ver Anexo 1).

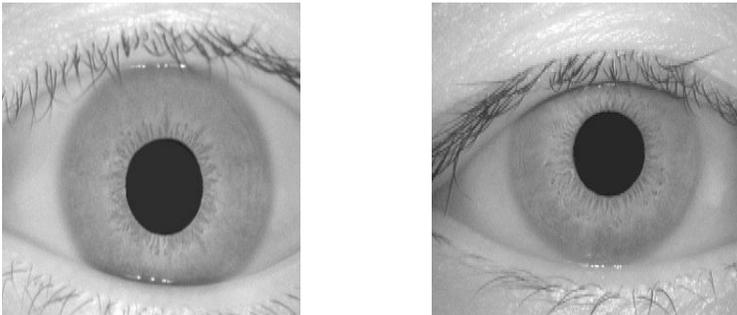


Figura 16: Imágenes de ojos fotografiados con cámara NIR, pertenecientes a la base de datos CASIA.

Para la compresión y visualización de los procesos desarrollados en el componente y tomando en cuenta que la extensión del mismo es .dll, lo cual imposibilita observar por parte del equipo de desarrollo las transformaciones que ocurren en las imágenes, se decide implementar una aplicación visual en la herramienta MatLab. Dicha aplicación para su funcionamiento hará uso del componente desarrollado. Esto facilitará en gran medida el trabajo de los desarrolladores, ya que a través de la aplicación visual, se podrán realizar los procesos de cargar imágenes, aplicar los algoritmos de segmentación y normalización, así como, la distancia de Hamming, esta última utilizada para comprobar si los vectores extraídos son iguales o no.

A partir de las HU se crean las pruebas de aceptación y durante una interacción de una HU seleccionada en la planificación de iteraciones conllevará una o varias pruebas de aceptación, tantas como sean necesarias para garantizar su correcto funcionamiento. Cuando una HU ha sido correctamente implementada el usuario o el cliente describe los aspectos a testear. Es responsabilidad del cliente verificar la corrección de las pruebas y tomar decisiones sobre las mismas.

Tabla 17: Caso de prueba para la HU Transformar a escala de grises.

| Caso de prueba de aceptación | |
|---|--|
| Código de caso de prueba: CP1 | Nombre de historia de usuario: Transformar imagen a escala de grises. |
| Responsable de la prueba: Nathaly Rodríguez Guillén. | |

| |
|---|
| <p>Descripción de la prueba:</p> <p>Prueba de funcionalidad para transformar las imágenes a escala de grises.</p> |
| <p>Condición de ejecución:</p> <p>El usuario debe haber seleccionado en el menú buscar imágenes en color, en caso contrario no hace nada.</p> |
| <p>Entrada/Pasos de ejecución:</p> <p>El proceso se realiza automáticamente después de haber cargado las imágenes en color.</p> |
| <p>Resultado esperado:</p> <p>Todas las imágenes son transformadas a escala de grises.</p> <p>Muestra la imagen de prueba transformada a escala de grises.</p> |
| <p>Evaluación de la prueba: Prueba satisfactoria.</p> |

Tabla 18: Caso de prueba para la HU Segmentar imagen de iris.

| Caso de prueba de aceptación | |
|--|--|
| <p>Código de caso de prueba: CP2</p> | <p>Nombre de historia de usuario: Segmentar imagen de iris.</p> |
| <p>Responsable de la prueba: Mariena Ferrer Barrientos.</p> | |
| <p>Descripción de la prueba:</p> <p>Prueba de funcionalidad para segmentar las imágenes. Cada imagen de la base de datos y la imagen de prueba son segmentadas para extraer los bordes de cada una de las imágenes.</p> | |
| <p>Condición de ejecución: Ninguna.</p> | |
| <p>Entrada/Pasos de ejecución:</p> <p>El Usuario debe presionar el botón “Aplicar Algoritmo”.</p> | |

| |
|---|
| <p>Resultado esperado:</p> <p>Se extrae los bordes de cada imagen.</p> <p>Se muestran los resultados después de aplicarle la segmentación a la imagen de prueba. Dicho resultado es la imagen obtenida del iris.</p> |
| <p>Evaluación de la prueba: Prueba satisfactoria.</p> |

Tabla 19: Caso de prueba para la HU Normalizar imagen de iris

| Caso de prueba de aceptación | |
|---|---|
| <p>Código de caso de prueba: CP3</p> | <p>Nombre de historia de usuario: Normalizar imagen de iris.</p> |
| <p>Responsable de la prueba: Mariena Ferrer Barrientos.</p> | |
| <p>Descripción de la prueba:</p> <p>Prueba de funcionalidad para normalizar las imágenes después de haber sido segmentada.</p> | |
| <p>Condición de ejecución: Ninguna.</p> | |
| <p>Entrada/Pasos de ejecución:</p> <p>El Usuario debe presionar el botón “Aplicar Algoritmo”.</p> | |
| <p>Resultado esperado:</p> <p>Se transforma la región circular del iris en una representación rectangular de tamaño fijo.</p> | |
| <p>Evaluación de la prueba: Prueba satisfactoria.</p> | |

Tabla 20: Caso de prueba para la HU Extraer las características.

| Caso de prueba de aceptación | |
|---|---|
| <p>Código de caso de prueba: CP4</p> | <p>Nombre de historia de usuario: Extraer las características.</p> |

| |
|--|
| Responsable de la prueba: Nathaly Rodríguez Guillén. |
| Descripción de la prueba: Prueba de funcionalidad para extraer los vectores a los resultados de segmentar y normalizar. |
| Condición de ejecución: Ninguna |
| Entrada/Pasos de ejecución: El Usuario debe presionar el botón “Aplicar Algoritmo”. Cada vector extraído por la segmentación y normalización son procesados seguidamente por el algoritmo Daugman. |
| Resultado esperado: Se extraen los vectores con los principales componentes de las imágenes. Se muestra como resultado del proceso 3 imágenes, la primera es el iris segmentado, la segunda y la tercera imágenes del iris normalizado. |
| Evaluación de la prueba: Prueba satisfactoria. |

Tabla 21: Caso de prueba para la HU Exportar vector característico.

| Caso de prueba de aceptación | |
|--|--|
| Código de caso de prueba: CP5 | Nombre de historia de usuario: Exportar el vector característico. |
| Responsable de la prueba: Mariena Ferrer Barrientos. | |
| Descripción de la prueba: Prueba de funcionalidad para guardar los vectores de características de las imágenes. | |
| Condición de ejecución: Ninguna | |
| Entrada/Pasos de ejecución: El usuario presiona el botón “Exportar Vector” luego de haber extraído los vectores de características | |

| |
|--|
| mediante los algoritmos propuestos. |
| <p>Resultado esperado:</p> <p>Se muestra una ventana donde el usuario debe seleccionar el directorio donde desea guardar los vectores de las imágenes.</p> <p>Se crean 2 archivos de extensión .mat, uno con el vector de la imagen de prueba y otro con los vectores de cada imagen de la base de datos de entrenamiento en la ubicación seleccionada.</p> |
| <p>Evaluación de la prueba: Prueba satisfactoria.</p> |

Tabla 22: Caso de prueba para la HU Calcular distancia de Hamming.

| Caso de prueba de aceptación | |
|--|---|
| Código de caso de prueba: CP6 | Nombre de historia de usuario: Calcular la distancia de Hamming. |
| Responsable de la prueba: Nathaly Rodríguez Guillén. | |
| Descripción de la prueba: | |
| Prueba de funcionalidad para demostrar que los vectores extraídos son adecuados para el reconocimiento del iris. | |
| Condición de ejecución: Ninguna | |
| Entrada/Pasos de ejecución: | |
| Se debe presionar el botón “Buscar iris a comparar” para mostrar si el iris fue generado del mismo iris guardado o por otro. | |
| Resultado esperado: | |
| Se calcula la distancia de Hamming de cada iris de la base de datos al iris de prueba. | |
| Demuestra si es el mismo iris o si no se corresponde con el iris. | |

Evaluación de la prueba: Prueba satisfactoria.

3.3.3 Etapa de verificación.

Para la realización de la verificación se emplearon 900 plantillas de iris. En la primera etapa del sistema se obtuvo un alto porcentaje de reconocimiento, sin embargo, en algunas pruebas el umbral (*Threshold*) fue superado por 15 plantillas de diferentes personas, es decir, los valores de coordenadas, ángulos y distancias entre iris fueron semejantes para más de una persona. Por lo tanto, en esas pruebas el sistema entregaba a la salida más de una plantilla reconocida, esto ocasionaba también, que se aumentara el porcentaje de falsa aceptación. Por esta razón, se realizó una segunda etapa que consiste en verificar las plantillas resultantes, en esta etapa se mejoró el porcentaje de falsa aceptación, aunque no estaba bien del todo y en esta segunda etapa el umbral fue superado por 8 plantillas. Teniendo en cuenta el resultado de la segunda etapa se decide realizar una tercera con las plantillas que dieron semejantes. Con esta prueba se eliminan las plantillas similares y únicamente se acepta como verdadera una sola plantilla.

3.3.3.1 Resultados obtenidos

Tabla 23: Resultados de las pruebas.

| Etapas | Falsa Aceptación (FAR) | % | Falso Rechazo (FRR) | % |
|--------|------------------------|------|---------------------|------|
| 1 | 15 | 1,67 | 0 | 0 |
| 2 | 8 | 0,89 | 0 | 0 |
| 3 | 1 | 0,11 | 1 | 0,11 |

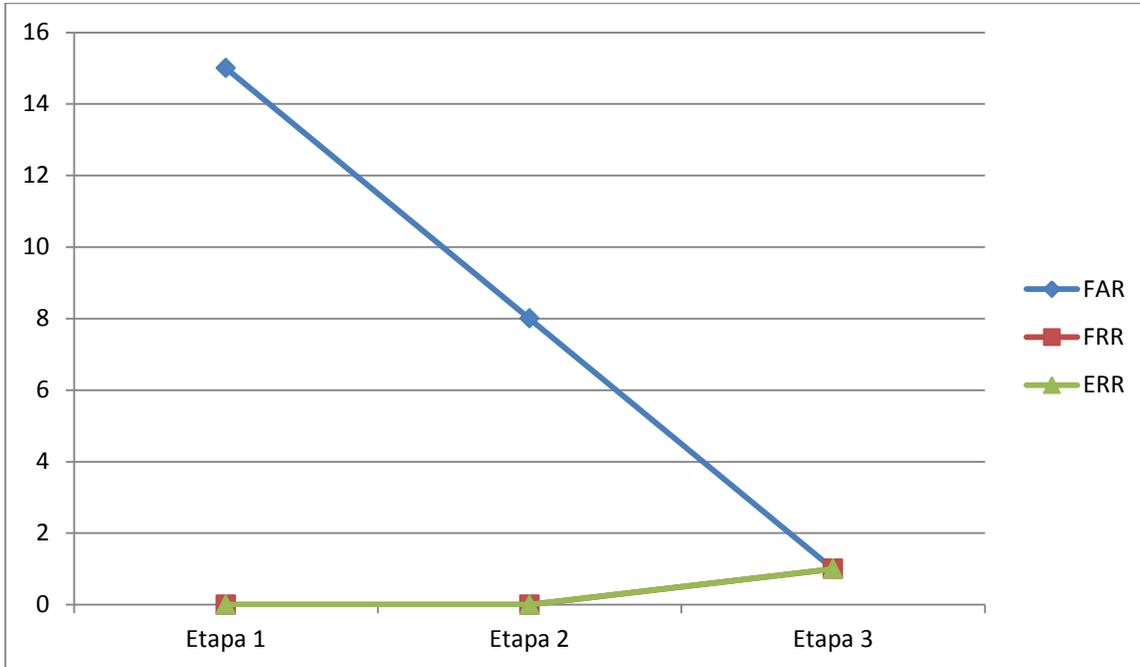


Figura 17: Gráfica sobre los resultados de las pruebas.

3.4 Conclusiones parciales

Luego de terminadas las iteraciones llevadas a cabo para la construcción del componente y la realización de las pruebas a cada una de las HU, se concluye que:

El desglose de las historias de usuario en tareas de la ingeniería fue una buena práctica que mostró a los programadores las funcionalidades específicas a implementar.

La utilización de estándares de programación permitió que los desarrolladores se sintieran más cómodos al implementar y siguieran una guía común para un mejor entendimiento del código.

El empleo de las funcionalidades ya implementadas por MatLab, suavizó el trabajo del equipo de desarrollo, ya que se redujeron los cálculos y operaciones matemáticas.

Las pruebas de aceptación concluyeron de manera exitosa demostrando la satisfacción del cliente con la solución.

Conclusiones

El desarrollo del presente trabajo de diploma permitió realizar un inmerso estudio sobre los procesos que se realizan para la comparación de iris. Una vez concluida la investigación se obtuvo como resultado un componente que automatiza el proceso de comparación de iris que podrá ser integrado al sistema de verificación dándole cumplimiento al objetivo de la investigación; específicamente a los objetivos específicos de la siguiente manera:

- ✓ Se realizó un estudio detallado de las tendencias actuales de los algoritmos para la comparación de iris.
- ✓ Luego del estudio de las tecnologías, metodologías y herramientas que existen actualmente y que podían ser utilizadas en el desarrollo de la solución se escogieron las más convenientes de acuerdo a las características del componente.
- ✓ Se realizó un análisis del sistema obteniendo como resultado los requisitos que debía cumplir el mismo y luego se realizó un diseño detallado del mismo.
- ✓ Se implementó un algoritmo para la comparación de iris obteniendo como resultado un componente que automatizó el proceso de comparación a partir de las imágenes que se obtienen como resultado del proceso de extracción de características.
- ✓ Se le realizaron pruebas al componente con el fin de validar la solución desarrollada y verificar el correcto funcionamiento del componente.

Recomendaciones

Como resultado de la investigación y el componente realizado, surgen nuevas ideas para trabajos futuros, las mismas se enuncian a continuación a modo de recomendaciones:

- ✓ Investigar nuevos métodos de extracción o combinaciones de ellos que permitan un mejor rendimiento.
- ✓ Integrar este componente a futuros sistemas de identificación que se desarrollen en el departamento de biometría del CISED.
- ✓ La extensión del componente a imágenes en tres dimensiones.

Referencias Bibliográficas

1. **Tejedor Gómez, Jesus.** *Análisis comparativo de algoritmos en segmentación de iris.* 2009.
2. **Lopez Carvajal, Jaime y Gonzales Patiño, Javier.** *Estado del arte: Reconocimiento automático del iris humano.* s.l. : Scientia Et Technica, 2005. págs. 77-81. Vol. Vol XI num 29.
3. **González Jarrosay, Anet y Barnet Díaz, Eider.** *Componente para la extracción de características en imágenes faciales.* s.l. : Universidad de las Ciencias Informáticas, 2012.
4. **Schiffman, Harvey.** *La Percepción Sensorial.* Limusa Wiley. 2011. pág. p. 244. ISBN 968-18-5307-5.
5. **Persaud, Moore.** *Embriología clínica.* [ed.] ISBN 84-8174-725-4. s.l. : Elsevier España S.A, 2004.
6. *Individual Biometrics: Iris Scan.* s.l. : National Center for State Courts, 2006. <http://ctl.ncsc.dni.us/biomet%20web/BMIris.html>.
7. **Smith, Kelly.** *Iris Patent Question.* 9 June 2005.
8. IrisGuard. [En línea] [Citado el: 6 de 6 de 2013.] <http://www.irisguard.com/>.
9. ACNUR. Agencia de la ONU para Refugiados. [En línea] 2001. [Citado el: 6 de 6 de 2013.] <http://www.acnur.org/t3/>.
10. BIOID be recognized. [En línea] 2002. [Citado el: 9 de 6 de 2013.] <http://www.bioid.com/>.
11. UK Border Agency. [En línea] 2012. [Citado el: 9 de 6 de 2013.] <http://www.ukba.homeoffice.gov.uk/customs-travel/Enteringtheuk/usingiris/>.
12. *Afghan girl.* **McCurry, Steve.** s.l. : Along Afghanistan's War-torn Frontier, 1985, National Geographic.
13. **McCurry, Steve.** National Geographic. [En línea] 4 de 2002. [Citado el: 6 de 6 de 2013.] <http://ngm.nationalgeographic.com/2002/04/afghan-girl/original-story-text>.
14. CENATAV. Centro de Aplicaciones de Tecnologías de Avanzada. [En línea] [Citado el: 6 de 6 de 2013.] <http://www.cenatav.co.cu/>.

15. **Ortiz López, Jaime.** *Sistema de detección de vida vía software en imágenes de iris utilizando criterios de calidad.* [ed.] ESCUELA POLITECNICA SUPERIOR. s.l. : Universidad autónoma de Madrid, octubre 2011.
16. *How iris recognition works.* **Daugman, J.** s.l. : Proceedings of 2002 International Conference on Image Processing, 2002, Vol. Vol. 1.
17. **Wildes, R., y otros.** *A system for automated iris recognition.* 1994. págs. 121-128. Vol. Proceedings IEEE Workshop on Applications of Computer Vision.
18. **Kong, W. y Zhang, D.** *Accurate iris segmentation based on novel reflection and eyelash detection model.* [ed.] Video and Speech Processing. 2001. Vol. Proceedings of 2001 International Symposium on Intelligent Multimedia.
19. *Person identification technique using human iris recognition.* **Tisse, C., y otros.** [ed.] International Conference on Vision Interface. Canada : s.n., 2002.
20. *Iris recognition using circular symmetric filters.* **Ma, L., Wang, Y. y Tan, T.** [ed.] Chinese Academy of Sciences. s.l. : National Laboratory of Pattern Recognition, Institute of Automation, 2002.
21. *Recognition of Human Iris Patterns for Biometric Identification.* **Masek, Libor.** The University of Western Australia : s.n., 2003.
22. *Authentication for secure environments based on iris scanning technology.* **Sanderson, S. y Erbetta, J.** 2000. IEE Colloquium on Visual Biometrics.
23. *Relations between the statistics of natural images and the response properties of cortical cells.* **Field, D.** 1987, Journal of the Optical Society of America.
24. **Oppenheim, A. y Lim., J.** *The importance of phase in signals.* 1981. págs. 529-541. Proceedings of the IEEE 69.
25. **Lee, T.** *Image representation using 2D gabor wavelets.* 1996. Vol. Vol. 18. IEEE Transactions of Pattern Analysis and Machine Intelligence.
26. *Ráfagas de Hamming.* **Echaiz, Mg. Javier.** Vol. Organización de Computadoras.
27. *Trasmisión de Datos.* **Briceno, José E.**

28. **Buzón, José Campo.** *Evaluación de conformidad con ISO/IEC 19784-1 para algoritmos de identificación biométrica.* [ed.] Dpt. de tecnología electrónica Escuela Politécnica Superior. Madrid : Universidad Carlos III de Madrid, 2009.
29. BioAPI Consortium. *Version 2.0 of the BioAPI Specification — International Version.* [En línea] Site hosted by Daon. Content partially provided by OSS Nokalva. [Citado el: 14 de febrero de 2013.] http://www.bioapi.org/Version_2.0_Description.asp.
30. *INTERNATIONAL STANDARD ISO/IEC.* Vols. 6, Iris image data. ISO/IEC 19794-6:2011(E).
31. **NCITS), InterNational Committee for Information Technology Standards (formerly, [ed.].** *ANSI/INCITS 358-2002.* 2002. pág. 120. Vols. Information technology - BioAPI Specification.
32. **Carrillo Pérez, Isaías, Pérez González, Rodrigo y Rodríguez Martín, Aurelio David.** *Metodología de desarrollo del software.* 2006.
33. **Díaz Flores, Mirian Milagros.** *Documento Comparativo RUP vs XP.* s.l. : Escuela de Ingeniería de Sistemas.
34. **Hernandez Orallo, Enrique.** *El Lenguaje Unificado de Modelado (UML).*
35. **Booch, G., Rumbaugh, J. y Jacobson, I.** *El Lenguaje Unificado de Modelado.* s.l. : Addison Wesley, 1999.
36. **Miss, Alexander.** *Herramienta Case. Visual Paradigm para UML.* s.l. : Instituto Tecnológico de Villahermosa.
37. **Agudo Ruano, David y Niños Aznar., Pablo.** *Herramientas Case. Análisis y especificación de sistemas multimedia.* 2011/2012.
38. **Caldas, Francisco Jose.** Matlab. [En línea] [Citado el: 19 de enero de 2013.] <http://gemini.udistrital.edu.co/comunidad/estudiantes/ocala/matlabTut/acerca.php>.
39. Docstoc. [En línea] [Citado el: 19 de enero de 2013.] <http://www.docstoc.com/docs/2147683/Tutorial-de-MatLab>.
40. *Extreme Programming Explained.* **Beck, Kent.** 2da Edición, 2004. ISBN-10: 0321278658.

41. **Letelier, Patricio y Penadés, M^a Carmen.** *Métodologías ágiles para el desarrollo de software: eXtreme Programming (XP)*. Universidad Politécnica de Valencia : s.n.
42. **Quiroga, Juan Pablo.** *Requerimientos funcionales y No funcionales*. s.l. : Dpto. de Ingeniería de Sistemas y Computación.
43. *Implementación de un sistema para la gestión de eventos y servicios literarios en la Universidad de las Informáticas.* **Martínez Borrego, Kerlins.** s.l. : UCILITERARIA, 2010.
44. Masek, Libor; www.csse.uwa.edu.au/~pk/studentprojects/libor/sourcecode.html
45. **Escribano, G.F.** *Extreme Programming*. UK ©2002, 2002. s.l. : London : Springer-Verlag London. ISBN:3-540-44024-0.
46. Chinese Academy of Sciences – Institute of Automation Version 1.0. [En línea] 2003. [Citado el: 20 de 1 de 2013.] <http://www.sinobiometrics.com>.

Glosario de Términos

Cámara anterior: es una estructura del ojo que está limitada por delante por la córnea y por detrás por el iris. Este espacio se encuentra relleno de un líquido que se llama humor acuoso, el cual circula libremente y aporta los elementos necesarios para el metabolismo de las estructuras anteriores del ojo que no reciben aportes de la sangre como la córnea.

Cámara posterior: es un espacio situado detrás del iris, bañado por un fluido que se llama humor acuoso, el cual produce en esta zona los llamados procesos ciliares.

CASE: Herramientas de Ingeniería de Software Asistida por Ordenador.

CENATAV: Centro de Aplicaciones de Tecnologías de Avanzadas.

CISED: Centro de Identificación y de Seguridad Digital.

CRC: Tarjetas Clase-Responsabilidad-Colaboración.

.dll: Bibliotecas de Enlace Dinámico.

ERR: Tasa de igualdad de error.

FAR: Tasa de Falsa Aceptación.

Fotograma: a cada una de las imágenes individuales captadas por cámaras de video.

FRR: Tasa de Falso Rechazo.

HU: Historias de usuarios.

IDE: Entorno Integrado de Desarrollo.

Iris: es la membrana coloreada y circular del ojo que se separa de la cámara anterior de la cámara posterior. Posee una apertura de tamaño variable que comunica las dos cámaras: la pupila.

MatLab: *Matrix Laboratory*.

Píxel: es la menor unidad homogénea que forma parte de una imagen digital, ya sea esta una fotografía, un fotograma de video o un gráfico.

Retina: es un tejido sensible a la luz situado en la superficie interior del ojo. Es similar a una tecla donde se proyectan las imágenes.

RP: Reconocimiento de Patrones.

RUP: Proceso Racional Unificado.

Sinusoidal: Curva que representa gráficamente la función trigonométrica seno.

UCI: Universidad de las Ciencias Informáticas.

UML: Lenguaje Unificado de Modelado.

VP: *Visual Paradigm*.

XP: Programación Extrema.

Anexo

Anexo 1

La base de datos CASIA (45) cuenta con un conjunto de imágenes de iris tomadas entre Abril de 1992 y Abril de 1994. La base de datos fue elaborada por *Chinese Academy of Sciences – Institute of Automation* en colaboración con el grupo de habla, visión y robótica del departamento de ingeniería de la Universidad de China.

Las imágenes de cada individuo fueron tomadas en diferentes sesiones. Las tomas incluyen variaciones de iluminación y distancia. Todas las imágenes fueron tomadas contra un fondo homogéneo oscuro con el individuo en posición frontal.

Los archivos se encuentran en formato JPG, con una resolución de 320x380 píxeles.

La base de datos CASIA (45) se puede descargar de forma gratuita de la página de internet de <http://www.sinobiometrics.com/>.

Anexo 2

La escala de grises es empleada en las imágenes digitales en las que el valor de cada píxel posee un valor equivalente a una graduación de gris. Estas imágenes están representadas por sombras de grises, que van desde el negro más oscuro variando gradualmente en intensidad de grises hasta llegar al blanco.

Cuando se convierte una foto en escala de grises se puede representar un conjunto de colores en un tono de gris, o incluso poner cada color en una intensidad. Las escalas de grises son diferentes de las fotografías en blanco y negro, en las que los colores se codifican en blanco o en negro; la escala de grises ofrece una gama de tonalidades de gris entre ambos.

Las imágenes en escala de grises, emplean 8 bits para representar cada píxel lo que sólo permite una escala con 256 intensidades (o escalas de gris); es decir, 2 valores posibles para cada bit (0 y 1) elevado a 8 bits que se emplean para representar cada píxel, nos da 256 tonos de color diferentes que pueden representarse en una imagen en escala de grises.

Anexo 3

El usuario debe seleccionar el tipo de imágenes que serán procesadas para realizarles la comparación.

Anexo 4

El usuario debe seleccionar el tipo de imágenes que serán procesadas para comparar.



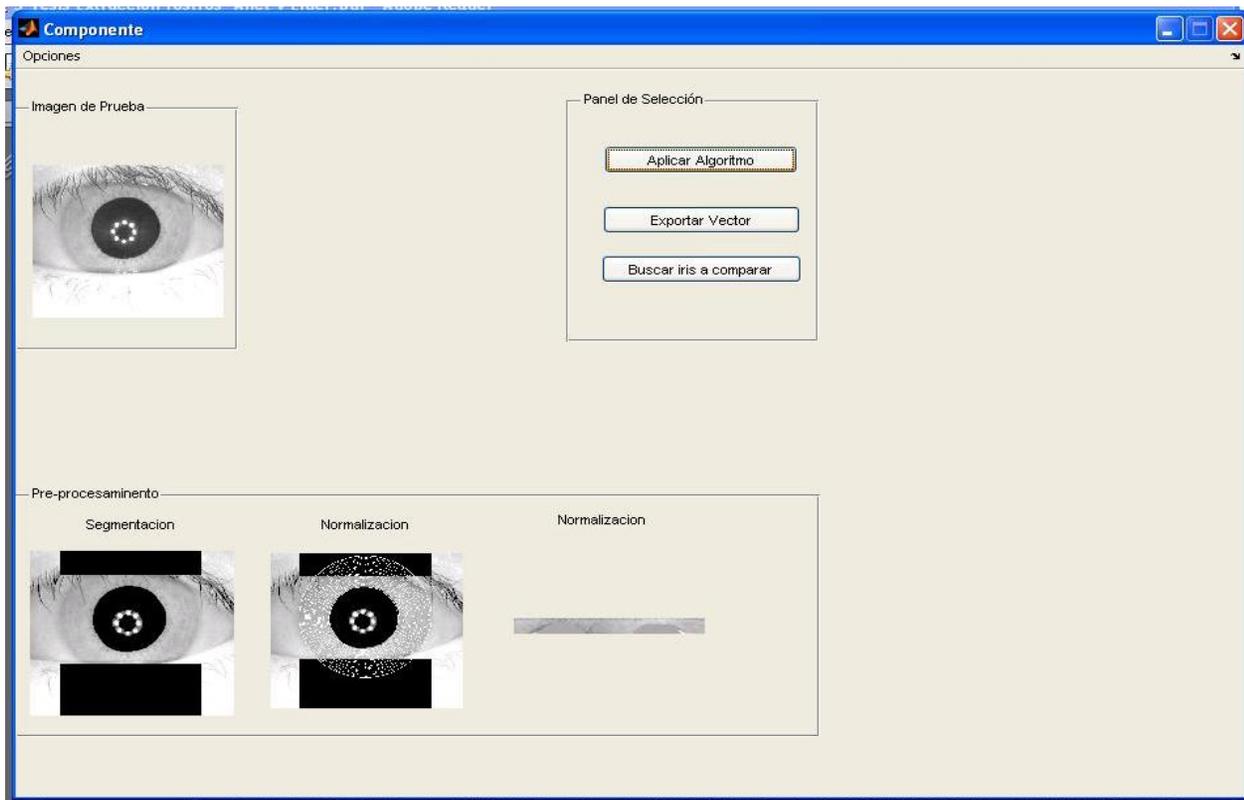
Anexo 5

El usuario debe indicar la ruta o dirección de la base de datos CASIA.



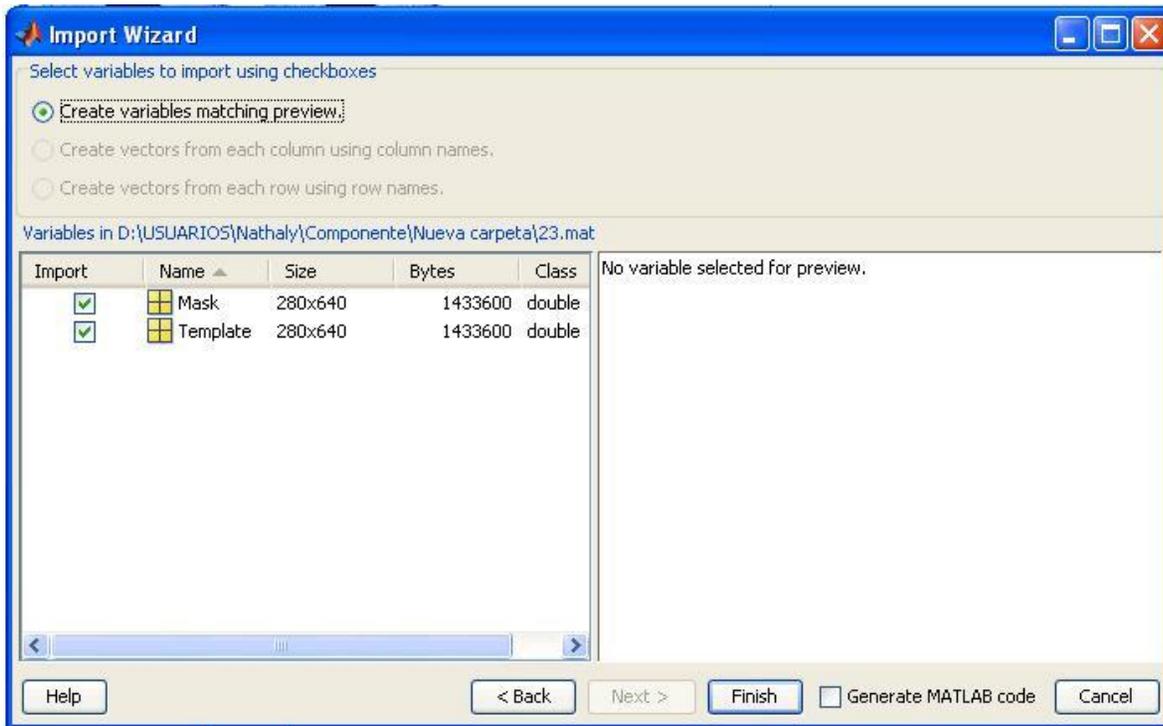
Anexo 6

El usuario deberá aplicar los algoritmos de pre-procesado.



Anexo 7

El usuario deberá después de aplicado el algoritmo de segmentación y normalización, exportar su vector característico. Los vectores característicos pueden ser guardados para su posterior análisis en ficheros .mat. Un vector contiene la máscara y la plantilla a comparar.



Anexo 8

Al presionar el botón Buscar iris a comparar se calcula la distancia de Hamming de cada vector de la base de datos a la imagen de pruebas. Y si el vector fue generado por la misma imagen de iris, te devuelve un mensaje de confirmación.

