

**Universidad de las Ciencias Informáticas**

**Facultad 7**



**Componente para la extracción de rasgos en  
culotes de proyectiles**

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

**Autores:** Reinier Pupo Ruiz

Juan Miguel Peña Cabrera

**Tutores:** Ing. Darien Menéndez Molina

Ing. Yoel Rivera Suárez

Ciudad de La Habana, Junio 2009

“Año del 50 Aniversario del Triunfo de la Revolución”

## DECLARACIÓN DE AUTORÍA

Declaramos que somos los únicos autores de este trabajo, y autorizamos la Universidad de las Ciencias Informáticas a hacer uso del mismo, en su beneficio, con carácter exclusivo.

Para que así conste firmamos la presente a los 30 días del mes de junio del año 2009.

Autores:

---

Reinier Pupo Ruiz

---

Juan Miguel Peña Cabrera

Tutores:

---

Ing. Darien Menéndez Molina

---

Ing. Yoel Rivera Suárez

## DATOS DE CONTACTO

### Tutores:

-Ing. Darien Menéndez Molina ([dmenendez@uci.cu](mailto:dmenendez@uci.cu))

Profesor graduado de Ingeniero en Ciencias Informáticas en la Universidad de Ciencias Informáticas. Posee la categoría de Adiestrado. Ha impartido las asignaturas de Programación II y Programación IV y Procesamiento Digital de Imágenes. En la actualidad es profesor de la Facultad 7 y se desempeña como Jefe de Proyecto dentro de Grupo de Procesamiento de Imágenes (GPI) de la Universidad de Ciencias Informáticas (UCI).

-Ing. Yoel Rivera Suárez ([yrsuarez@uci.cu](mailto:yrsuarez@uci.cu))

Profesor graduado de Ingeniero en Ciencias Informáticas en la Universidad de Ciencias Informáticas. Posee la categoría de Adiestrado. En la actualidad es profesor de la Facultad 7 en la asignatura Física y se desempeña como Jefe de Proyecto dentro de Grupo de Procesamiento de Imágenes (GPI) de la Universidad de Ciencias Informáticas (UCI).

# Agradecimientos

---

Agradezco a todos los que de una forma u otra han ayudado a que estas líneas puedan ser escritas como parte de este trabajo de diploma. A la Revolución, por brindarle la oportunidad a todos de ser un profesional. A mis padres, mi hermana, mi familia, a todos ellos por ser mí apoyo y guía, por sus consejos, por sus desvelos. A mi novia, mis amigos que me han ayudado y apoyado. A los tutores Yoel y Darien. A todos, Gracias.

Reinier

Agradezco primeramente a la Patria, a la Revolución y a Fidel por haberme permitido llegar hasta mi carrera universitaria. A mis maestros y profesores que en las diferentes educaciones me fueron formando tanto académica como moralmente, en especial a mis maestros: Noelia, Isabel, Idania, Mirta, Iria y Reynaldo. A mis padres, hermanos, abuelos, mi familia y amigos por su apoyo incondicional de cada día. A Maikel, Charro, Jorge, Yeiser, Piñeda, Dario y Joseito. A los tutores por su ayuda y preocupación. Gracias.

Juan Miguel

# Dedicatoria

---

*A mis padres que siempre me han apoyado y guiado hasta aquí.*

*A mi hermana de sangre y alma, Arianna.*

*A Ana, por su apoyo y ayuda en todo momento.*

*A mi familia.*

*A mis amigos.*

**Reinier**

*A mis padres por su apoyo y guía constante*

*A mis hermanas Yuliet, Isabel y Luritza*

*A mis abuelas Nina y María*

*A la memoria de mi abuelo Oscar, mi tío Luis Oscar y mi amigo Dairon*

*A mis tíos y tías en especial a Nereida, Joaquín, Gismy, Zulema e Idelsy*

*A mis primos*

*A mi sobrina Merlin*

*A Guelfo por ser como un padre*

*A mis amigos que siempre me inspiraron*

**Juan Miguel**

# Resumen

---

En la actualidad, las armas de fuego están presentes en gran cantidad de crímenes, por lo que con el desarrollo de la computación y el procesamiento de imágenes digitales han surgido programas computacionales que cotejan imágenes balísticas. Este es un proceso bastante novedoso e inexplorado por lo que el presente trabajo de diploma tiene como objetivo el desarrollo de un componente para extraer rasgos característicos en imágenes de culotes de proyectiles balísticos como base de un sistema de identificación balística, para optimizar el proceso de correlación, que se hace tedioso al tener que cotejar de forma manual grandes cantidades de imágenes para lograr un resultado satisfactorio.

Para el desarrollo de los algoritmos se utilizó el lenguaje de programación C# en su versión 2.0 y como metodología de desarrollo, RUP, que rige todo el proceso de ingeniería de software. El componente procesa imágenes balísticas digitalizadas para obtener las regiones de interés y los descriptores propios de los gráficos para que sean utilizados en el proceso de correlacionamiento balístico con la utilización de novedosas técnicas en el procesado imagenológico.

El desarrollo de este componente hace posible la rapidez y confiabilidad en el proceso de extraer rasgos en las imágenes balísticas, permite precisión y disminución de la carga de trabajo de los peritos que usen un sistema ABIS (Automatic Ballistic Identification System) del cual sea parte el componente propuesto.

**Palabras claves:**

ABIS, descriptores, rasgos característicos, cotejo, balística, extracción.

## ÍNDICE

<b>AGRADECIMIENTOS</b> .....	<b>I</b>
<b>DEDICATORIA</b> .....	<b>II</b>
<b>RESUMEN</b> .....	<b>III</b>
<b>INTRODUCCIÓN</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>5</b>
1.1 IMAGEN DIGITAL:.....	5
1.2 DESCRIPTORES VISUALES:.....	6
1.3 PROCESAMIENTO DIGITAL DE IMÁGENES (PDI).....	7
1.4 ESTADO DEL ARTE.....	8
1.5 PRINCIPALES HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS.....	10
CONCLUSIONES.....	16
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA</b> .....	<b>18</b>
2.1 PROBLEMA Y SITUACIÓN PROBLEMÁTICA.....	18
2.2 OBJETO DE AUTOMATIZACIÓN.....	22
2.3 PROPUESTA DE SISTEMA.....	23
2.4 MODELO DE DOMINIO.....	23
2.5 ESPECIFICACIÓN DE LOS REQUISITOS DEL SOFTWARE.....	24
2.6 DEFINICIÓN DE LOS CASOS DE USO.....	25
2.7 DIAGRAMA DE CASOS DE USO DEL SISTEMA.....	26
2.8 CASOS DE USO POR CICLO DE DESARROLLO.....	28
CONCLUSIONES.....	29
<b>CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL SISTEMA</b> .....	<b>30</b>
3.1 DESCRIPCIÓN DE LA ARQUITECTURA.....	30
3.2 MODELO DE ANÁLISIS.....	31
3.3 MODELO DE DISEÑO.....	37
CONCLUSIONES.....	48
<b>CAPÍTULO 4: IMPLEMENTACIÓN</b> .....	<b>49</b>
4.1 TÉCNICAS UTILIZADAS.....	49
4.2 IMPLEMENTACIÓN.....	65
CONCLUSIONES.....	67
<b>CONCLUSIONES GENERALES</b> .....	<b>68</b>
<b>RECOMENDACIONES</b> .....	<b>69</b>
<b>REFERENCIAS BIBLIOGRÁFICAS</b> .....	<b>70</b>

<b>BIBLIOGRAFÍA .....</b>	<b>73</b>
<b>ANEXOS .....</b>	<b>78</b>
ANEXO 1. DESCRIPCIÓN EXTENDIDA DE LOS CASOS DE USO.....	78
ANEXO 2. IMÁGENES BALÍSTICAS.....	80
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>84</b>

# Introducción

---

A lo largo de la historia de la humanidad el hombre ha creado sus propias armas, instrumento cuyo principal objetivo es el de realizar ataques o defenderse de ellos, para infligir daño a su oponente. Progresivamente estas armas se han perfeccionado hasta llegar a lo que son hoy las armas de fuego, con diversas formas y tamaños, estas poseen cada vez mayor potencia y certeza.

No resulta extraño que debido a la efectividad y fácil adquisición por parte del hombre, las armas de fuego se hayan convertido en la herramienta principal con que se llevan a cabo la mayoría de los crímenes; de ahí la necesidad de hacer estudios de criminalística aplicados a la identificación de armas, a partir del análisis de las partes del proyectil encontrados en la escena del crimen.

La balística es la ciencia encargada de estudiar el movimiento, avance y proyección de todo proyectil lanzado al espacio en general y los lanzados por armas de fuego en particular. La balística forense es un estudio criminológico aplicado a la identificación del arma a partir del análisis científico del proyectil que ha disparado (1), además, estudia los fenómenos que ocurren en el interior del arma, durante el disparo de un proyectil, su trayectoria desde el momento que abandona la boca de fuego del arma hasta su impacto y los efectos producidos en el organismo durante el recorrido, al igual que las formas y características de las armas de fuegos y sus municiones.

Uno de los principales esfuerzos dedicados hoy por parte de los científicos forenses, es la determinación de correspondencia entre proyectiles o vainas encontradas en el lugar de los hechos con algún arma hallada en la propia escena o en poder de algún sospechoso. Estas técnicas de determinación de correspondencias se basan en que científicamente está demostrado que no hay dos armas que dejen marcas idénticas en la munición empleada (2).

Estas marcas se constituyen principalmente por las lesiones de la bala cuando se desliza por el ánima del cañón, y las producidas en la vaina, por la rampa de alimentación, las paredes de la recámara, la culata de cierre, la aguja percutora, el extractor y el expulsor. A través del análisis de estas huellas dejadas por el arma en el proyectil, es posible identificar si dicha arma fue la que realizó el disparo o no. Actualmente para llevar a cabo dicha tarea se han definido un conjunto de procedimientos técnicos con alto rigor

científico, aunque no exentos de modificación por parte de la comunidad científica y el desarrollo de la técnica.

El primer intento del que se tiene constancia de resolver un crimen realizado por un arma de fuego, es a principios del siglo XIX. En el año 1835, en la ciudad de Londres no había cuerpo de policía, tan solo un pequeño grupo de "ayudantes" reclutados por Henry Fielding -juez de paz de Westminster-, se les conocía como los Bow Street-Runners, y que se dedicaban a investigar los crímenes con la práctica de métodos poco ortodoxos, e incluso alguna que otra vez no muy legales (3).

Henry Goddard, uno de estos "peculiares investigadores", al observar una bala extraída del cuerpo de la víctima de un asesinato, se percató de la existencia de una llamativa protuberancia o abultamiento en la misma. Dado que por aquella época los tiradores habitualmente hacían mediante un molde o turquesa sus propios proyectiles, el investigador pensó que si encontraba el molde encontraría al asesino. Al revisar las casas de los sospechosos, en una de estas se encontró un molde y al fabricarse un proyectil que fue comparado con el obtenido, se logró descubrir al culpable (4).

Con la creación en Nueva York, alrededor del año 1924, del primer instituto de balística forense del mundo, *Bureau of Forensic Ballistics*, por Charles E. Wite, notable investigador en este campo, el físico John H. Fisher y el químico y gran especialista en microfotografía Philipp O. Gravelle; comienza la incorporación y avance gradual de la tecnología en esta área de la ciencia forense. Se introduce el microscopio calibrado, que permitió la medición de las lesiones en el proyectil con mucha más precisión.

Se crean microscopios más potentes que el fabricado anteriormente para medir con muchísima más precisión los campos intermedios, las estrías y su orientación. La unión de dos microscopios calibrados mediante un dispositivo óptico, permitió observar dos proyectiles juntos, superponiéndolos en una sola imagen y lograr que ambas giraran de manera que se pudieran ver tanto sus coincidencias como diferencias. La inclusión de tales herramientas al proceso de identificación balística contribuyó a la creación de los procedimientos manuales básicos que se realizan hoy en día.

Al final de la década del 1920 se crearon algunos proyectos con el objetivo de mejorar estas herramientas. Tal es el caso de la compañía Kodak, que creó un sistema de cámaras para tomar fotografías de una bala en vista panorámica. Se realizaron intentos de almacenar estas fotografías pero el proceso no fue efectivo y terminó descontinuándose el proyecto. Otros instrumentos fueron inventados para la grabación y

almacenamiento de imágenes balísticas pero debido a la complejidad e incapacidad de almacenar la gran cantidad de datos que se requería, fueron desechados (5).

Con el surgimiento y creciente desarrollo de la computación, la capacidad de cálculo ha aumentado considerablemente, con lo que se hace posible el procesamiento y almacenamiento tanto de información como de imágenes balísticas digitales. De esta forma se propició la construcción de Sistemas Automatizados de Identificación Balística (ABIS, por sus siglas en idioma inglés), que permiten comparar de forma automática el grado de similitud entre una evidencia balística contra un grupo de ellas previamente almacenadas; se identifican, si existen, aquellas con un gran porcentaje de similitud. El uso de un sistema de estas características posibilita que los investigadores forenses puedan relacionar casos entre sí y dar un veredicto con mucha más eficiencia y rapidez.

En el mundo existen pocos sistemas ABIS, estos son muy costosos y por lo tanto privativos de algunos pocos países. En Cuba se han realizado algunos estudios investigativos sobre el tema. Debido a las ventajas de un ABIS y el desarrollo que ha tenido en los últimos años los estudios en este campo, se hace necesario crear un componente que sirva para formar parte de un Sistema Automatizado de Identificación Balística.

Por todo lo antes descrito, se determina que el **Problema científico** del presente trabajo la siguiente interrogante: ¿Cómo facilitar la extracción de rasgos distintivos en imágenes de culotes de proyectiles disparados por armas de fuego?

La investigación tiene como **Objeto de estudio**: la extracción de rasgos distintivos en imágenes de culotes de proyectiles disparados por armas de fuego. El estudio se centra en el **Campo de acción** definido por las imágenes digitales de culotes de proyectiles.

Para dar solución al problema planteado se define como **Objetivo general**: implementar un componente que facilite la extracción de rasgos distintivos en imágenes digitales de culotes de proyectiles disparados por armas de fuego.

Para dar cumplimiento al objetivo general, se definieron las siguientes **Tareas de la investigación**:

- Analizar sistemas existentes para el procesamiento de imágenes balísticas.

- Analizar métodos para el procesamiento de imágenes digitales en el ámbito de la comparación de imágenes balísticas.
- Investigar algoritmos empleados para caracterizar imágenes.
- Proponer algoritmos a emplear para implementar el componente.
- Implementar un componente que permita el procesado de las imágenes balísticas digitales para obtener las regiones de interés y sus descriptores.

El trabajo está estructurado en cuatro capítulos como se describe a continuación:

En el capítulo 1 se aborda todo lo relacionado con la fundamentación teórica que sustenta la presente investigación, algunas definiciones necesarias para la comprensión del presente trabajo, el estado del arte y se exponen las tendencias, técnicas, tecnologías, métodos y software que son utilizados durante la investigación y construcción de la solución propuesta.

En el capítulo 2 se expone en detalle la situación problemática y el objeto de automatización. Se describe la propuesta del sistema y el diagrama conceptual del modelo de dominio. Se exponen los requisitos funcionales y no funcionales y el diagrama de casos de uso del sistema con su descripción detallada.

En el capítulo 3 se describe la arquitectura, se expone el modelo de análisis y sus diagramas de interacción. Se muestran los diagramas de clases del diseño con sus diagramas de interacción. Se describen y se especifican clases que se implementarán.

En el capítulo 4 se exponen los pasos realizados para la implementación del sistema y se muestra el diagrama de componentes.



## 1.2 Descriptores visuales:

Los descriptores visuales describen, como su nombre indica, las características visuales de los contenidos dispuestos en imágenes o en vídeos. Describen características elementales tales como la forma, el color, la textura o el movimiento, entre otros.

Los descriptores visuales se dividen en dos grandes grupos:

1. **Descriptores de información general:** contienen descriptores de bajo nivel, proporciona una descripción acerca del color, formas y regiones, texturas y movimiento.
2. **Descriptores de información de dominio específico:** proporcionan información acerca de los objetos y eventos que aparecen en la escena. Un ejemplo muy concreto es el de reconocimiento facial.

### 1.2.1 Descriptores de información general

Los descriptores de información general consisten en descriptores que cubren distintas características visuales básicas y elementales como son: el color, textura, forma, movimiento y localización, entre otras. La descripción es generada automáticamente mediante procesamiento digital de señales.

- **Color:** Es la cualidad más básica del contenido visual. Existen varias formas para describir el color: descripción de la distribución del color, la distribución espacial del color y la relación del color entre una secuencia o conjunto de imágenes:
- **Textura:** Es otra cualidad muy importante para describir una imagen. Se diseña para caracterizar las texturas o regiones de una imagen. Observan la homogeneidad de las regiones y los histogramas de los bordes de dichas regiones.
- **Forma:** Posee una información semántica muy importante, debido a que los humanos son capaces de reconocer los objetos sólo con ver su forma. No obstante, esta información sólo puede ser extraída mediante una segmentación similar a la que realiza el sistema visual humano. En la actualidad todavía no está disponible, pero si existen una serie de algoritmos que permiten una

buena aproximación. Estos descriptores describen las regiones, contornos y formas para imágenes 2D y volúmenes 3D.

- **Movimiento:** Se define mediante algunos descriptores que describen el movimiento en las secuencias de vídeo. Este movimiento va asociado a los objetos de la secuencia o bien al movimiento propio de la cámara. Ésta última información es proporcionada por el dispositivo de captura, mientras que el resto se realiza con procesamiento digital de señales.
- **Localización:** Se utiliza para describir elementos en el dominio espacial o temporal dentro de la secuencia de vídeo.

### 1.2.2 Descriptores de información de dominio específico

Éstos descriptores, que proporcionan información sobre objetos y eventos en la escena, no son fáciles de extraer, aún más si se pretende realizar una extracción automática. Sin embargo, siempre cabe la posibilidad, aunque ardua, de realizar un procesado manual.

Un ejemplo concreto de una aplicación que intenta extraer este tipo de información de forma automática sería el de reconocimiento facial.

## 1.3 Procesamiento Digital de Imágenes (PDI)

El procesamiento digital de imágenes se puede definir de manera sencilla como un conjunto de procedimientos que se realizan sobre una escena para su almacenamiento, transmisión, reconocimiento, análisis e interpretación.

Con el devenir de los años el PDI, donde en la actualidad de forma más genérica se le ha denominado visión por computadora, ha definido más nítidamente sus campos de acciones, entre los que se pueden mencionar:

- **Mejoramiento:** Tiene que ver con el aumento del contraste, la eliminación del ruido y el filtrado espacial y de frecuencia, entre otros.

- Restauración: Estudia fenómenos tales como desenfoque, movimiento y otros tipos de degradaciones que pueden originarse en el momento de la captación de una escena.
- Reconstrucción: Técnica muy empleada en la medicina y geología para la obtención de una imagen tridimensional a través de diferentes cortes.
- Segmentación: Procedimiento que se utiliza para separar diferentes tipos de objetos, aquí subyace un campo de gran importancia denominado detección de contornos.
- Compresión de imágenes: Tiene que ver con las formas de comprimir las imágenes para su almacenamiento y transmisión.
- Reconocimiento visual de patrones: Tiene como objetivo la búsqueda de métodos que sean capaces de identificar diferentes estructuras.
- Análisis e interpretación: Constituye una técnica de alto nivel y estudia el problema relacionado con la interpretación de la imagen.

En sentido general la visión por computadora se puede definir como una disciplina que trata con el estudio de las teorías, técnicas y herramientas relacionadas con el análisis de datos sensoriales obtenidos de un cierto contexto, la generación de descripciones simbólicas de esos datos y de suministrar conclusiones o diagnósticos que pueden ser de utilidad según criterios establecidos a priori.

## **1.4 Estado del Arte**

En la actualidad existen pocos sistemas que cotejan imágenes balísticas. En Estados Unidos, el BATFE (Bureau of Alcohol, Tobacco, Firearms and Explosives) ha construido el NIBIN (National Integrated Ballistic Information Network), una base de datos de imágenes de proyectiles y cartuchos balísticos implicados en algún crimen, a la que tienen acceso más de 200 laboratorios forenses de ese país (6).

La corporación canadiense Forensic Technology, ha trabajado en el tema de la identificación balística automatizada por más de quince años y actualmente es líder mundial en este tema. Posee cientos de agencias en más de 40 países y territorios. IBIS, Integrated Ballistic Identification System es el producto que han desarrollado que consta de varios componentes como el IBIS-TRAX 3D, que es un sistema de identificación balística integrado para capturar imágenes y comparar las evidencias de balas y casquillos usados, BrassTRAX-3DMC, una estación de captura de imágenes de casquillos IBIS, BulletTRAX-3DMC,

una estación de captura de imágenes de balas 3D IBIS y MatchPoint+MC, una estación de análisis comparativo 3D IBIS. Todos estos componentes hacen que este producto sea muy competitivo en el mercado mundial. (7)

DRUGFIRE es otro sistema desarrollado conjuntamente entre el FBI (Federal Bureau of Investigations) y la empresa norteamericana Mnemonics Systems INC (8). Este sistema, es considerado uno de los avances más significativos en el tema de la identificación de armas de fuego. DRUGFIRE utiliza imágenes de alta resolución, y realiza una correlación para encontrar una correspondencia en la red de computadoras que unen los laboratorios de criminalística de varios países. Esto permite resolver delitos que traspasen casos locales y regionales, puede resolver crímenes a nivel nacional e incluso cruzar las fronteras internacionales (9). Este software está instalado en más de 100 laboratorios de criminalística en todo el mundo, es utilizado por más de 500 examinadores de armas de fuego y técnicos. Ha procesado más 65,000 casos en todo el mundo y de estos a resuelto más de 5.200 casos internacionales.

SCICLOPS es un nuevo producto ABIS que permite caracterizar la superficie en 3D y minimiza errores de comparación integra de forma computarizada las superficies bajo estudio. Este sistema aún está bajo evaluación final, promete ser uno de los avances más importantes para la balística forense (10).

El sistema ruso ARSENAL tiene como tarea más importante, usar sus archivos balísticos de la manera más eficiente con el fin de hacer un examen a fondo y obtener resultados más fiables en un plazo lo más breve posible. El núcleo de la solución consiste en la automatización y la informatización de las operaciones de identificación, y esta tarea no puede resolverse sin el estado actual de las tecnologías más avanzadas. El sistema posee todas las herramientas necesarias para ello. ARSENAL es un moderno sistema de gran alcance capaz de automatizar todo el proceso de examen de las balas, cartuchos y sus fragmentos: a partir de la entrada de datos y creación de base de datos electrónica. Adecua y divide la pantalla de verificación a los resultados de las comparaciones realizadas (11).

En Cuba, no existe ningún sistema ABIS que agilice el trabajo de los peritos, se han realizado estudios como el de la Lic. Mayelin Macías Norat de la Universidad de Oriente, titulado "Reconocimiento Automático de Armas" donde se explica una vía que puede ser utilizada para la extracción de rasgos característicos en culotes de vainas. El Centro de Aplicaciones de Tecnologías de Avanzada (CENATAV),

también ha realizado investigaciones sobre este tema, principalmente en el área de procesamiento de imágenes digitales (12).

Con los altos costos que tienen estos sistemas, lo recomendable es tener una solución propia, contribuyendo de esta manera al avance de Cuba en esta materia y al ahorro de recursos que podrían ser utilizados en la compra de uno de estos Sistemas Automatizados de Identificación Balística. En Cuba, con el surgimiento de la Universidad de Ciencias Informáticas (UCI), se han capacitado cientos de jóvenes, y se capacitan otros miles para emprender tareas de gran envergadura como lo es la creación de un ABIS. La mayoría de estos software a nivel mundial, vienen ligados a un hardware específico para ellos que encarecen la compra.

Con la creación de un sistema propio se ayuda a que Cuba logre la independencia tecnológica. También se incrementará los conocimientos científicos de los profesionales, con la creación de herramientas propias sin depender de los altos costos de estos sistemas a nivel mundial.

## **1.5 Principales Herramientas y Tecnologías utilizadas**

### **1.5.1 Lenguaje de Programación**

#### **1.5.1.1 Microsoft Visual C# 2.0**

C# es un lenguaje de propósito general OO (Orientado a Objetos) creado por Microsoft para la plataforma .NET. La sintaxis y estructuración de C# es muy parecida a la de C++ o Java, puesto que la intención de Microsoft es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos. Sin embargo, su sencillez y el alto nivel de productividad son comparables con los de Visual Basic.

El código escrito en C# es auto contenido, lo que significa que no necesita de ficheros adicionales al propio fuente tales como ficheros de cabecera o ficheros IDL (Interface Definition Language)<sup>3</sup>. El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para

---

<sup>3</sup> lenguaje de especificación de interfaces que se utiliza en software de computación distribuida. Ofrece la sintaxis necesaria para definir los procedimientos o métodos que se quieren invocar remotamente.

quienes se compile, lo que facilita la portabilidad del código. No se incluyen elementos poco útiles de lenguajes como C++, tales como macros, herencia múltiple o la necesidad de un operador diferente del punto (.) para acceder a miembros de espacios de nombres (::).

C# soporta todas las características propias del paradigma de programación orientada a objetos: encapsulamiento, herencia y polimorfismo, además su sintaxis utiliza el modelo de objetos de la plataforma .NET, por lo que permite transferir resultados de las funciones incorporadas en MATLAB, para el procesamiento de imágenes, para desarrollar soluciones en C#. (13). Entre otras características están:

- Compatibilidad con el diseño, la programación y la implementación de servicios con rapidez.
- Diseñadores de formularios y controles visuales para crear aplicaciones basadas en Windows muy completas.
- Lenguaje intuitivo basado en C++ (familiar para los programadores de C++ y Java, así como ofrece nuevas estructuras de lenguaje intuitivas que simplifican enormemente las tareas de programación).

Se puede integrar Visual SourceSafe (sistema de control de versiones para proyectos de equipo y proyectos individuales) al Entorno de Desarrollo Integrado (IDE en inglés) de Visual Studio .NET como proveedor de control de código fuente.

## **1.5.2 Lenguaje de Modelado**

### **1.5.2.1 UML (Unified Modeling Language) 2.0**

UML es un lenguaje para modelar que emplean los ingenieros para el diseño de software antes de pasar a su construcción, al igual que sucede con cualquier producto manufacturado o fabricado en serie. Desde el punto de vista tecnológico, este lenguaje tiene una gran cantidad de propiedades que han sido las que, realmente, han contribuido a que se considere como un estándar de facto de la industria que es en realidad.

Algunas de las propiedades de UML como lenguaje de modelado estándar son:

- Concurrencia, es un lenguaje distribuido y adecuado a las necesidades de conectividad actuales y futuras.
- Ampliamente utilizado por la industria desde su adopción por Object Management Group (OMG) (14).
- Reemplaza a decenas de notaciones empleadas con otros lenguajes.
- Modela estructuras complejas.
- Las estructuras más importantes que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clase, componentes y nodos.
- Emplea operaciones abstractas como guía para variaciones futuras, añade variables si es necesario.
- Comportamiento del sistema: casos de uso, diagramas de secuencia y de colaboraciones, que sirven para evaluar el estado de las máquinas.

Existe una gran diversidad de herramientas que permiten la realización de sus diagramas de forma automatizada, ya que realizarlos manualmente implica mucho tiempo y puede constituir una fuente importante de errores. Su uso puede ser una garantía de establecer comunicación con una gran cantidad de personas. Los diagramas de descripción de la interacción fusionan los diagramas de actividades y secuencia para permitir que los fragmentos de interacción sean fácilmente combinados con los puntos y flujos de decisión. (15)

### **1.5.3 Metodología de Desarrollo**

Todo desarrollo de software es riesgoso y difícil de controlar, pero sin una metodología de por medio, lo que se obtiene son clientes insatisfechos con el resultado y desarrolladores aún más insatisfechos. Sin embargo, muchas veces no se toma en cuenta el utilizar una metodología adecuada, sobre todo cuando

se trata de proyectos pequeños de dos o tres meses. Lo que se hace con este tipo de proyectos es separar rápidamente el aplicativo en procesos, cada proceso en funciones, y por cada función determinar un tiempo aproximado de desarrollo.

Cuando los proyectos que se van a desarrollar son de gran envergadura, generalmente toma sentido el basarse en una metodología de desarrollo, y se comienza a buscar la más apropiada para dicho caso. Lo cierto es que muchas veces no se encuentra la más adecuada y se termina por hacer o diseñar una propia metodología, algo que por supuesto no está mal, siempre y cuando cumpla con el objetivo.

Muchas veces se realiza el diseño de un software de manera rígida, con los requerimientos que el cliente solicitó, de tal manera que cuando el cliente en la etapa final, solicita un cambio se hace muy difícil realizarlo. De hacerse, altera muchas cosas que no se habían previsto, y es justo este, uno de los factores que ocasiona un atraso en el proyecto y por tanto la incomodidad del desarrollador por no cumplir con el cambio solicitado y el malestar por parte del cliente por no tomar en cuenta su pedido. Para evitar estos incidentes se debe haber llegado a un acuerdo formal con el cliente, al inicio del proyecto, de tal manera que cada cambio o modificación no perjudique al desarrollo del mismo.

### **1.5.3.1 Rational Unified Process (RUP)**

RUP (Proceso Unificado de Desarrollo, en español) es un proceso que define claramente quien, cómo, cuándo y qué debe hacerse. Como su enfoque está basado en modelos utiliza un lenguaje bien definido para tal fin, el UML. Este aporta herramientas como los casos de uso, que definen los requerimientos. Permite la ejecución iterativa del proyecto y del control de riesgos. Además, trata los riesgos importantes en las dos primeras fases, inicio y elaboración, y cualquier riesgo restante al principio de la fase de construcción, por orden de importancia. Identifica, gestiona y reduce los riesgos en las primeras fases mediante las iteraciones. En consecuencia, los riesgos no identificados o ignorados no emergen más tarde para poner en peligro el proyecto entero (16).

Las características principales del proceso son:

- Guiado por los Casos de Uso.
- Centrado en la Arquitectura.

- Iterativo e Incremental.

A través de un proyecto guiado por RUP, los requerimientos funcionales son expresados en la forma de Casos de Uso, que guían la realización de una arquitectura ejecutable de la aplicación. Además el proceso focaliza el esfuerzo del equipo en construir los elementos críticos estructuralmente y del comportamiento (Línea base de la Arquitectura) antes de construir elementos menos importantes. La mitigación de los riesgos más importantes guía la definición y confirmación del alcance en las primeras etapas del ciclo de vida. Finalmente RUP divide el ciclo de vida en iteraciones que producen versiones incrementales de los ejecutables de la aplicación.

RUP implementa las siguientes mejores prácticas asociadas al proceso de Ingeniería de Software:

- Desarrollo Iterativo.
- Manejo de los Requerimientos.
- Uso de una Arquitectura basada en componentes.
- Modelación Visual.
- Verificación Continua de la Calidad.
- Manejo de los Cambios.

De forma general RUP se divide en 4 fases el desarrollo del software, estas son:

- Inicio: En la que se define el modelo del negocio, el alcance y los límites del proyecto.
- Elaboración: En la cual se define, valida y cimienta la arquitectura.
- Construcción: Donde se desarrolla el producto.
- Transición: Es la encargada de poner el producto en manos de los usuarios.

## 1.5.4 Herramientas

### 1.5.4.1 Enterprise Architect 7.0

Enterprise Architect (EA) es una herramienta comprensible de diseño y análisis UML, cubre el desarrollo de software desde el paso de los requerimientos a través de las etapas del análisis, modelos de diseño, pruebas y mantenimiento. Es una herramienta multi-usuario, basada en Windows, diseñada para ayudar a construir software robusto y fácil de mantener. Ofrece salida de documentación flexible y de alta calidad.

El UML provee beneficios significativos para ayudar a construir modelos de sistemas de software rigurosos y donde es posible mantener la trazabilidad de manera consistente. Enterprise Architect soporta este proceso en un ambiente fácil de usar, rápido y flexible.

Provee trazabilidad completa desde el análisis de requerimientos hasta los artefactos de análisis y diseño, a través de la implementación y el despliegue. Combinados con la ubicación de recursos y tareas incorporados, los equipos de Administradores de Proyectos y Calidad están equipados con la información que ellos necesitan para ayudarles a entregar proyectos en tiempo. Las bases de Enterprise Architect están construidas sobre la especificación de UML 2.0. Usa perfiles UML para extender el dominio de modelado, mientras que la Validación del Modelo asegura integridad.

EA soporta generación e ingeniería inversa de código fuente para muchos lenguajes populares, incluyendo C++, C#, Java, Delphi, VB.Net, Visual Basic y PHP. Brinda soporte para los 13 diagramas de UML 2.0 (17).

Las principales características son:

- Interfaz de usuario intuitiva
- Ingeniería de Código Directa e Inversa
- Agregados para vincular EA a Visual Studio.NET o Eclipse
- Modelado de Base de Datos
- Soporta Control de Versiones

- Archivos Binarios de Ingeniería Inversa para Java y .NET
- Soporte para Prueba

#### **1.5.4.2 Visual Studio Team System 2008 Team Suite**

La herramienta Visual Studio Team System 2008 proporciona múltiples disciplinas a los miembros del equipo con un conjunto integrado de herramientas para la arquitectura, el diseño, desarrollo, creación de bases de datos y pruebas de aplicaciones. Los miembros del equipo pueden colaborar continuamente y utilizar un conjunto completo de herramientas y la orientación en cada etapa de la aplicación del ciclo de vida (18).

#### **1.5.4.3 Visual Source Safe 2005**

Visual SourceSafe (VSS) es la herramienta de control de versiones de Microsoft. Resulta una herramienta útil para las personas que desarrollan programas bajo el sistema operativo Windows ya que se integra fuertemente con el entorno de desarrollo integrado o IDE de Visual Studio que permite un manejo relativamente simple de versiones sobre una computadora individual y en equipos de trabajo relativamente pequeños.

#### **1.5.4.4 AQtime Profiler 5.x**

Es una herramienta creada por la compañía AutomatedQA para medir el comportamiento de un software, está diseñada para ayudar a comprender completamente cómo se comporta un programa durante su ejecución. Puede ser ejecutado como una herramienta en si o puede integrarse con Microsoft Visual Studio, Borland Developer Studio y CodeGear RAD Studio, que permite optimizar las aplicaciones sin salir del IDE<sup>4</sup> de desarrollo (19).

## **Conclusiones**

En este capítulo se han definido algunos conceptos importantes para el entendimiento del trabajo como el de imagen digital y descriptor visual. Se ha hecho un análisis del estado del arte a nivel nacional e

---

<sup>4</sup> *Integrated Development Environment*, es un programa compuesto por un conjunto de herramientas para un programador.

internacional. Fueron caracterizados los principales sistemas ABIS existentes en el mundo y se determinó que no son factibles comprarlos debido a sus altos precios en el mercado mundial. Se expusieron las principales propiedades por las cuales se han escogido las herramientas y tecnologías utilizadas en el desarrollo del sistema.

# CAPÍTULO 2: Características del Sistema

---

En este capítulo se profundiza en el objeto de estudio, se hace una descripción de los procesos del negocio centrándose en los objetivos estratégicos de la organización, el flujo de los procesos involucrados en el campo de acción, causas que originan la situación problemática y sus consecuencias. Se describen los procesos que serán objeto de automatización. Se realiza una propuesta inicial del sistema. Se expone el dominio en que se enmarca el sistema y se exponen los requisitos funcionales y no funcionales. Se definen los casos de uso del sistema, exponiéndose el diagrama de casos de uso del sistema así como la descripción detallada de cada caso de uso.

## 2.1 Problema y Situación Problemática

El estudio de las formas y características de las armas de fuego y sus municiones es muy importante para el proceso de cotejo balístico, esto permite la identificación del arma que ha disparado el proyectil a partir de un estudio científico de dos de los componentes del proyectil, la bala o plomo, y la vaina o cartucho.

En un laboratorio de criminalística se encuentran los medios necesarios para procesar las pruebas, este proceso no es más que un estudio comparativo de las vainas y balas incriminados o dubitados, cotejándolos con vainas y proyectiles testigos o indubitados obtenidos por el perito que se apoya en la o las armas sometidas a estudio.

Un perito no es más que una persona que declara ante un tribunal, que tiene la característica particular de poseer conocimientos técnicos en una ciencia, arte u oficio determinado, que le permiten emitir opiniones sobre materias de relevancia para la resolución de un juicio. (20)

En un laboratorio común de criminalística, en el área de Balística Forense, la actividad fundamental se centra en la evidencia –armas, plomos y vainas-, se hace un estudio de estas para llegar a una conclusión. El perito analiza de forma manual la evidencia recogida en el lugar del crimen, compara visualmente a través del microscopio de comparación las semejanzas o diferencias que pueden tener las vainas o balas incriminados o dubitados con los testigos o indubitados.

Se controla la concordancia o no entre el arma y el proyectil y/o vaina incriminada de características cuya no coincidencia descartan, por si solas, toda posibilidad de identidad, tales como: igualdad de calibre; número de estrías, dirección, paso y ancho de las mismas, ubicación relativa del conjunto extractor-botador, etc. La no concordancia entre las características expuestas del arma sospechosa con la vaina y/o el proyectil incriminado, descarta toda posibilidad de vinculación entre las mismas, mientras que corroborada la coincidencia de estas características, se hace necesario profundizar la investigación y se recurre al cotejo de las características microscópicas (fig. 2.1).

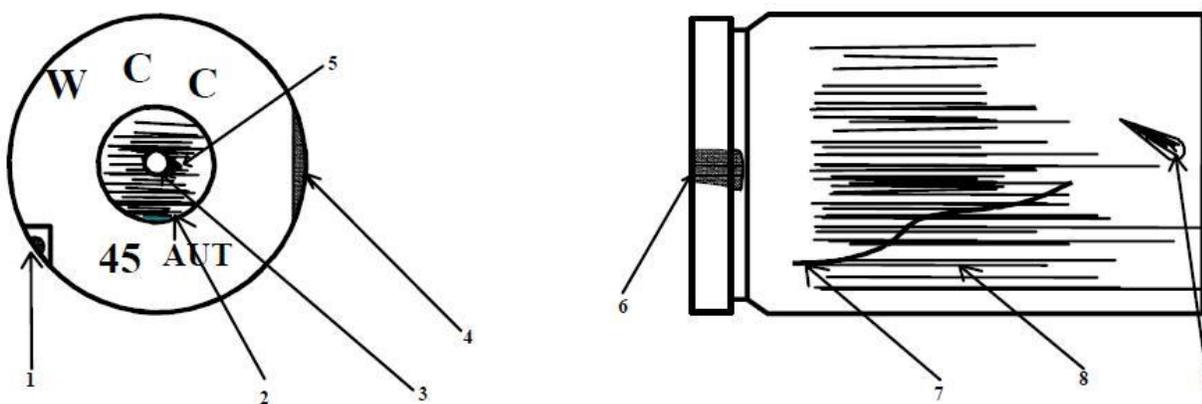


Figura 2.1 Marcas en la vaina de un proyectil disparado por un arma semiautomática.

Leyenda:

- 1- Marca del expulsador o eyector.
- 2- Marca de la parte trasera del cañón del arma (espaldón, véase fig. 2.2) en el fulminante.
- 3- Marca de la aguja percutora.
- 4- Marca del extractor.
- 5- Marca del arrastre de la aguja percutora (Firing pin drag).
- 6- Marca del extractor.

- 7- Marca del cargador.
- 8- Marca de la recámara del arma.
- 9- Marca de la abertura de expulsión.

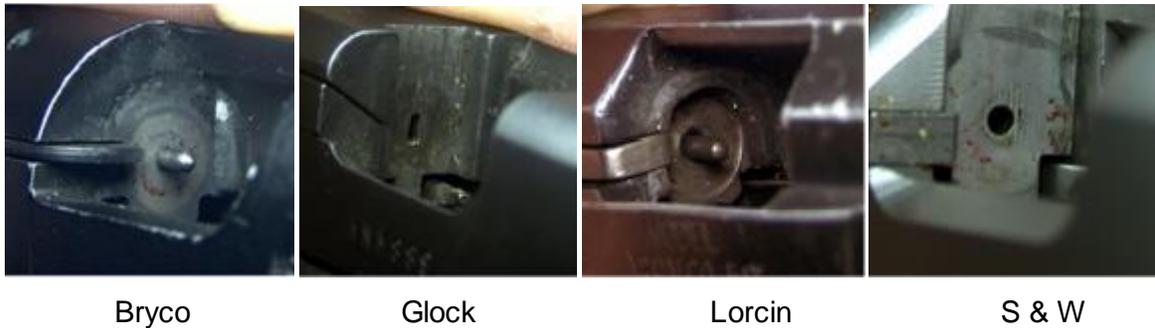


Figura 2.2 Diferentes tipos de caras opuestas al culote de la vaina, espaldones (Breech Face).

Una de las formas de determinar la identidad de un arma es efectuar un estudio comparativo entre las vainas dubitadas e indubitadas o testigos (fig. 2.3). Se utiliza preferentemente el microscopio comparador (fig. 2.4) mediante el cual se efectuará el cotejo de las líneas o rayas identificadoras que hayan dejado estampadas en la vaina, piezas tales como: la aguja de percusión, la uña extractora, el botador y el espaldón. De existir alguna coincidencia surgirá la categórica conclusión de un común origen, es decir que ambas vainas (dubitada e indubitada) fueron servidas por la misma arma.



Figura 2.3 Vista de cotejo de vainas realizada por el perito.

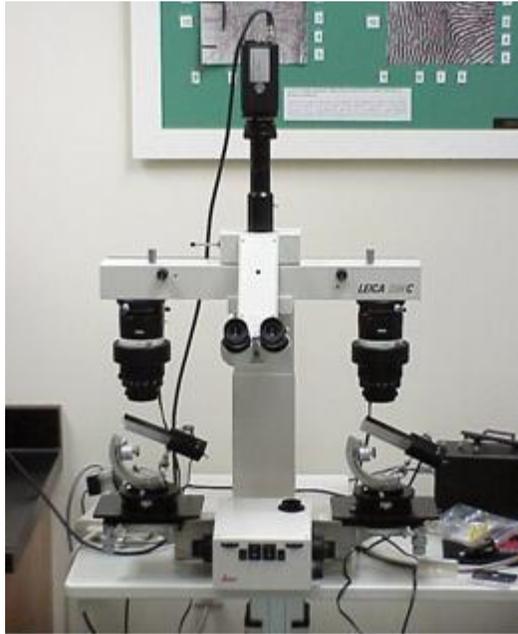


Figura 2.4 Microscopio de comparación LEICA.

De manera general, los principales procesos que se ejecutan con respecto a la identificación de rasgos característicos en culotes de vainas de proyectiles balísticos son: recolección de evidencia (vainas); identificación de características generales (calibre, tipo de marca de la aguja percutora, etc.); selección según características generales de las vainas, cuál o cuáles son candidatas para cotejar, en caso de tener un arma candidata que podría haber participado en el crimen, se obtiene en el laboratorio la vaina testigo (disparada y recolectada por el perito) a través de las técnicas existentes; se compara con el auxilio del microscopio de comparación balística, de forma visual, cual de todas las vainas testigos es la que pertenece a la misma arma en estudio.

Este proceso es muy tedioso y propenso a errores debido a la gran carga de trabajo del perito, que tiene que observar gran número de candidatos, para obtener un veredicto, además de la variable error presente en todo momento debido al pesado trabajo. Todo esto trae consigo una demora considerable en el proceso de cotejo balístico para evitar errores, ya que la decisión final no puede ser errónea porque se trata de crímenes, donde el culpable debe ser castigado, no un inocente.

En la recolección de la evidencia, el perito se presenta en el lugar del crimen para recolectar cualquier material que haya sido dejado por el criminal y analizar los parámetros necesarios para la investigación, este preservará la integridad de la evidencia.

Una vez que haya sido trasladada la evidencia, se procederá a la identificación de características generales de la vaina como: calibre, tipo de marca de la aguja percutora (cuadrada, circular, en los costados), posible marca del arma que la disparó entre otras.

En caso de que haya sido confiscada un arma sospechosa de haberse realizado el crimen con ella, se obtendrá la muestra del proyectil correspondiente y se compararán el proyectil incriminado y la muestra a través del microscopio, se verá entonces si las características son lo bastante similares para afirmar que provienen de la misma arma y de esta manera dar solución al crimen.

En caso de tener proyectiles que hayan estado involucrados en crímenes anteriores o hayan sido registrados por fuerza de la ley en el momento de vender un arma, en caso de que esto último este permitido, se comenzará un proceso de comparación uno por uno, vaina testigo con vaina incriminada, hasta obtener un porcentaje bastante alto de coincidencia o no obtener ningún resultado satisfactorio.

Por toda la problemática existente en este flujo de trabajo, principalmente la demora, y a raíz del proceso de informatización de la sociedad cubana, se decide crear un componente que agilice la obtención de los rasgos característicos en los culotes de las vainas de proyectiles, con el objetivo de optimizar este proceso de la balística forense.

## **2.2 Objeto de automatización**

A continuación se describe el proceso del negocio que será automatizado por el componente propuesto.

*-Extracción de rasgos característicos de la marca de la aguja percutora y del fulminante en imágenes digitales de culotes de vainas de proyectiles.*

Es el proceso mediante el cual se procesa la imagen digital, a través de las técnicas de procesamiento digital de imágenes, para obtener las características descriptivas de la marca, como firma digital para una posterior correlación con otras firmas y obtener de esta manera el porcentaje de similitud.

## 2.3 Propuesta de Sistema

Se propone la creación de un componente para extraer rasgos característicos de la marca de la aguja percutora en imágenes balísticas digitales. Para llevar a fin tal propósito se desarrollará una librería que pueda ser utilizada como componente de un Sistema Automático de Identificación Balística y agilizar el proceso de identificación pericial en crímenes cometidos por armas de fuego. Ésta permitirá obtener de manera automática las regiones de interés para su procesamiento y la obtención de los rasgos distintivos.

## 2.4 Modelo de Dominio

Los procesos que se enmarcan en el flujo de trabajo actual, no complementan la realización de un modelo de negocio, por lo que se ha decidido crear un modelo de dominio donde se recogen los principales conceptos presentes en el área de estudio.

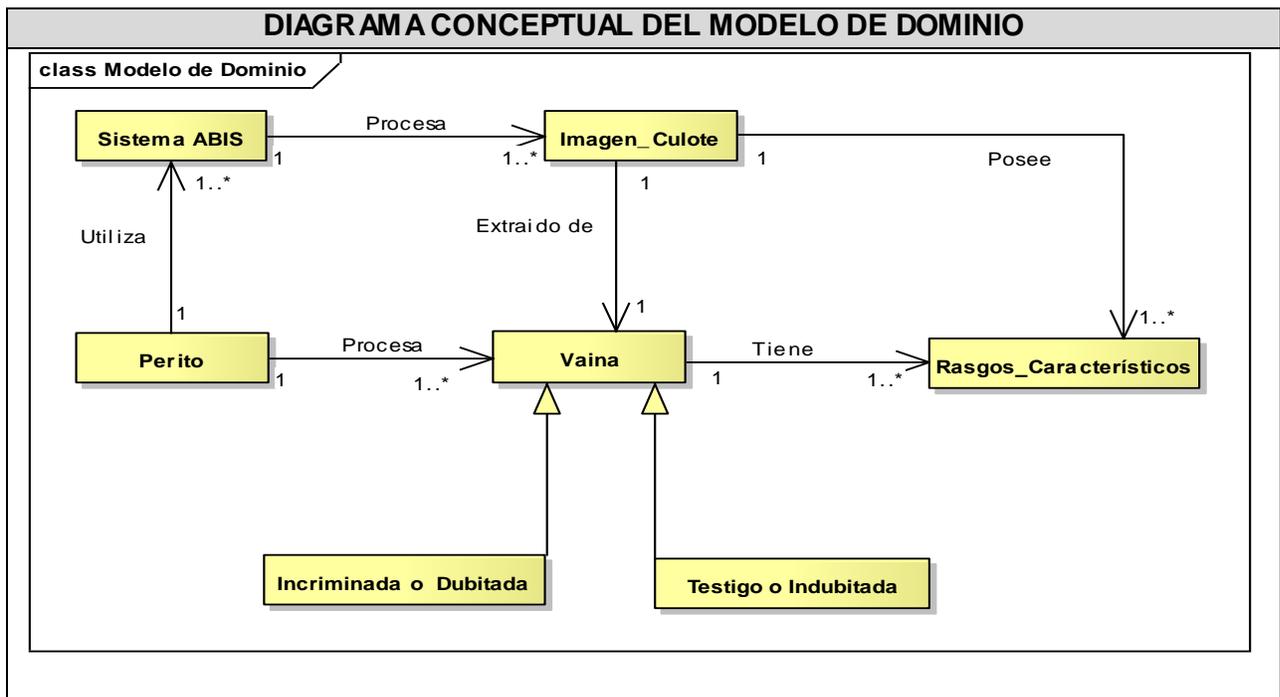


Figura 2.5 Diagrama Conceptual del Modelo de Dominio. Se muestran los principales objetos o conceptos enmarcados en el proceso en estudio.

**Sistema ABIS:** Software que utilizará las funciones del componente propuesto.

**Perito:** Una persona que declara ante un tribunal, que tiene la característica particular de poseer conocimientos técnicos en una ciencia, arte u oficio determinado, los cuales les permiten emitir opiniones sobre materias de relevancia para la resolución de un juicio. Es quien interactúa con el Sistema ABIS para después cotejar visualmente las vainas que han sido recomendadas por el sistema con mayor similitud a la vaina en estudio.

**Imagen Culote:** Imagen balística digital (ver fig. 1.1) que será procesada por el componente a petición del sistema ABIS que maneja el Perito.

**Vaina:** Objeto de metal que compone un proyectil, donde se encuentran varias marcas identificativas (ver fig. 2.1) del arma que ha disparado el proyectil. Ésta puede ser incriminada o dubitada y testigo o indubitada, la primera es la que se obtiene en la escena del crimen y la segunda la que se obtiene en el laboratorio para cotejar con la incriminada en caso de que se haya encontrado un arma sospechosa.

**Rasgos Característicos:** Propiedades que definen la firma digital de las marcas en estudio.

## 2.5 Especificación de los Requisitos del Software

### 2.5.1 Requerimientos Funcionales

Los Requerimientos Funcionales (RF) son condiciones o capacidades que el sistema debe cumplir, suficientemente buenas como para llegar a un acuerdo entre los clientes (incluyendo usuarios) sobre qué debe y qué no debe hacer el sistema. (16)

**RF1.** Obtener Región de Interés.

**RF1.1.** La imagen a procesar debe tener la calidad y formato correctos.

**RF2.** Obtener Rasgos Característicos.

**RF2.1.** Se necesitan obtener las regiones de interés.

## **2.5.2 Requerimientos No Funcionales**

Los Requerimientos No Funcionales (RNF) especifican propiedades del sistema, como restricciones del entorno o de la implementación, rendimiento, dependencias de la plataforma, facilidad de mantenimiento, extensibilidad y fiabilidad (16). Se piensa en este tipo de requerimiento después que se conocen los RF, para determinar cómo ha de comportarse el sistema y que cualidades debe tener. Se identificaron los siguientes RNF:

### 1. Rendimiento.

1.1 Se debe procesar la imagen para obtener las regiones de interés en no más de dos segundos.

1.2 Se debe obtener los rasgos de las marcas en no más de dos segundos.

### 2. Usabilidad

2.1 Proveer descripción detallada de las funcionalidades del componente.

2.2 Interfaz de comunicación sencilla.

### 3. Portabilidad

3.1 El componente debe poder ser utilizado sobre diferentes plataformas.

3.2 El componente debe correr en versión de Windows XP o superior.

### 4. Hardware

4.1 Se debe tener 256 Mb de memoria RAM o superior y tarjeta de video de 64 Mb o superior.

## **2.6 Definición de los casos de uso**

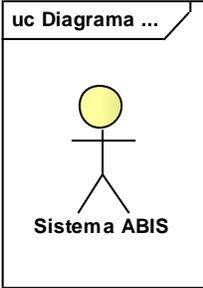
Los Casos de Uso no son parte del diseño (el cómo), sino parte del análisis (qué). De forma que al ser parte del análisis ayudan a describir qué es lo que el sistema debe hacer. Los Casos de Uso son qué hace el sistema desde el punto de vista del usuario. Es decir, describen un uso del sistema y cómo este interactúa con el usuario (21).

### 2.6.1 Definición de los actores

Un actor es una idealización de una persona externa, de un proceso, o de un ente que interactúa con un sistema, un subsistema, o una clase. Un actor caracteriza las interacciones que los usuarios exteriores pueden tener con el sistema. En tiempo de ejecución, un usuario físico puede estar limitado a los actores múltiples dentro del sistema. Diferentes usuarios pueden estar ligados al mismo actor y por lo tanto pueden representar casos múltiples de la misma definición de actor.

Cada actor participa en uno o más casos de uso. Interactúa con el caso de uso (y por lo tanto con el sistema o la clase que posee el caso de uso), intercambiando mensajes. La implementación interna de un actor no es relevante en el caso de uso; un actor puede ser caracterizado suficientemente por un conjunto de atributos que definen su estado.

Los actores pueden ser definidos en jerarquías de generalización, en las cuales una descripción abstracta del actor es compartida y aumentada por una o más descripciones específicas del actor. Un actor puede ser un ser humano, otro sistema informático, o un cierto proceso ejecutable. Se dibuja a un actor como una persona pequeña con trazos lineales y el nombre debajo de él (15).

Actor del Sistema	Justificación
	Es el sistema con quien interactúa el componente, es el que inicia todos los casos de usos.

### 2.7 Diagrama de casos de uso del sistema

En la figura 2.6 se muestra el diagrama de casos de uso del sistema para el componente propuesto.

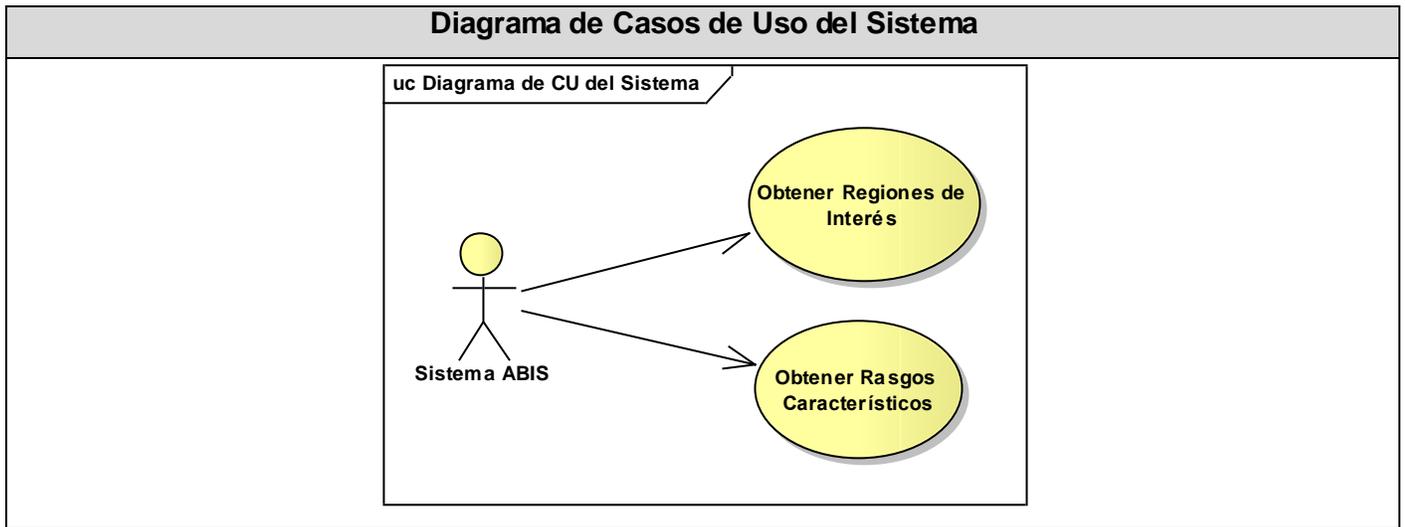


Fig. 2.6 Diagrama de casos de uso del sistema.

### 2.7.1 Listado de casos de uso

<b>CU-1</b>	Obtener Regiones de Interés
<b>Actor</b>	Sistema ABIS
<b>Descripción</b>	El caso de uso inicia cuando el Sistema ABIS hace la petición de que se obtengan las regiones de interés para obtener sus rasgos característicos, el sistema procesa la imagen balística digital con diferentes técnicas de procesamiento digital de imágenes para obtener las regiones que son de interés (las que tienen las marcas que permitan identificar el arma que ha disparado el proyectil). El sistema retorna los círculos que delimitan las regiones de interés.
<b>Referencia</b>	RF1
<b>Prioridad</b>	Crítico

<b>CU-2</b>	Obtener Rasgos Característicos
<b>Actor</b>	Sistema ABIS
<b>Descripción</b>	El caso de uso inicia cuando el Sistema ABIS pide obtener los rasgos

	característicos de la marca de la aguja percutora y de la región comprendida entre la marca de la aguja percutora y el borde del fulminante (Breech Face Mark) en una imagen balística digital, el componente requerirá las regiones de interés para a partir de estas, separar la imagen en regiones y procesarlas con las técnicas de procesamiento digital de imágenes, devuelve los rasgos característicos de las regiones de las marcas de la imagen digital.
<b>Referencia</b>	RF2
<b>Prioridad</b>	Crítico

## 2.8 Casos de uso por ciclo de desarrollo

### Primer ciclo de desarrollo.

Cód	Nombre de caso de uso	Paquete	Justificación de la selección.
CU1	Obtener Regiones de Interés	–	La región obtenida es el punto de partida para obtener las características de las marcas.

### Segundo ciclo de desarrollo.

Cód	Nombre de caso de uso	Paquete	Justificación de la selección.
CU2	Obtener Rasgos Característicos	–	Es el caso de uso más importante, sin él, el sistema no tendría sentido porque no habría nada con que relacionar, es el que se encarga de extraer los descriptores.

En el Anexo 1 se muestran las descripciones detalladas de los casos de uso del sistema. Se muestra el flujo de eventos que ocurren entre el actor y el sistema.

## **Conclusiones**

En este capítulo se han definido los principales procesos que actualmente se pueden observar en un laboratorio de criminalística con respecto al cotejo de las vainas. Se expusieron el objeto de automatización y el dominio del campo de acción. Se mostraron los requisitos funcionales y no funcionales que tiene el sistema. Se mostró el diagrama de casos de uso con las descripciones detalladas de estos últimos.

# CAPÍTULO 3: Análisis y Diseño del Sistema

---

En este capítulo se hace referencia al patrón de arquitectura que se decidió utilizar para la realización de la solución propuesta. Se definen las clases del análisis y del diseño del software. Se realizan los diagramas de secuencia en el análisis y el diseño para responder de manera eficiente a los requerimientos funcionales. Por último se describen las clases Interfaz, Controladoras y Entidades presentes.

## 3.1 Descripción de la Arquitectura

La arquitectura de software, está relacionada con el diseño y la implementación de estructuras de software de alto nivel. Es el resultado de ensamblar un cierto número de elementos arquitectónicos de forma adecuada para satisfacer la mayor funcionalidad y requerimientos de desempeño de un sistema, así como requerimientos no funcionales, como la confiabilidad, escalabilidad, portabilidad, y disponibilidad.

En el componente propuesto se utiliza el patrón Pipes & Filters (Tuberías y Filtros), donde cada componente tiene un conjunto de entradas y uno de salida. Un componente lee un flujo de datos en la entrada y produce un flujo de datos diferente en su salida. Esto es logrado aplicando una transformación local al flujo de entrada mientras este se lee, de tal forma que el flujo de salida empieza antes que se consume todo el flujo de entrada. Por lo tanto a los componentes se les denomina filtros. Los conectores sirven como conductos para transmitir las salidas de un filtro a las entradas de otro, por lo cual se les denomina tuberías.

Este patrón tiene algunas restricciones, como que los filtros deben ser entidades independientes: en particular no deben compartir estados con otros filtros. Los filtros no conocen la identidad del filtro de donde proviene el flujo que reciben como entrada y tampoco la identidad del filtro a donde llega el flujo de salida. Este patrón de arquitectura es particularmente efectivo a la hora de descomponer el problema en pasos independientes, reutilizar filtros, facilitar el mantenimiento, independencia y ejecución concurrente de filtros.

## 3.2 Modelo de análisis

Un modelo de análisis ofrece una especificación más precisa de los requisitos que la que se tiene como resultado de la captura de requisitos, incluyendo al modelo de casos de uso. Este modelo se describe con la utilización del lenguaje de los desarrolladores, y puede por tanto introducir un mayor formalismo y ser utilizado para razonar sobre los funcionamientos internos del sistema.

Se estructuran los requisitos de un modo que facilita su comprensión, su preparación, su modificación, y en general, su mantenimiento. Se puede considerar como una primera aproximación al modelo de diseño (aunque es un modelo por si mismo), y es por tanto una entrada fundamental cuando se da forma al sistema en el diseño y la implementación.

### 3.2.1 Clase del análisis

Una clase del análisis representa una abstracción de una o varias clases y/o subsistemas del diseño del sistema. Ésta se centra en el tratamiento de los requisitos funcionales y pospone los no funcionales, denominándolos requisitos especiales, hasta llegar a las actividades de diseño e implementación subsiguientes. Su comportamiento se define mediante responsabilidades en un nivel más alto y menos formal.

Una clase del análisis participa en relaciones, aunque esas relaciones son más conceptuales que sus contrapartidas de diseño e implementación. Las clases del análisis siempre encajan en uno de tres estereotipos básicos: de interfaz, de control o de entidad (véase Fig. 3.1). Cada estereotipo indica una semántica específica (descrita brevemente), lo cual constituye un método potente y consistente de identificar y describir las clases del análisis y contribuye a la creación de un modelo de objetos y una arquitectura robusta.

Clase	Descripción
 Clase Interfaz	Se utilizan para modelar la interacción entre el sistema y sus actores (usuarios y sistemas externos). Esta interacción a menudo implica recibir (y presentar) información y peticiones de (y hacia

	los usuarios) y los sistemas externos.
 <p>Clase Control</p>	Representan coordinación, secuencia, transacciones, y control de otros objetos y se usan con frecuencia para encapsular el control de un caso de uso en concreto. Se utilizan para representar derivaciones y cálculos complejos, como la lógica del negocio que no pueden asociarse con ninguna información concreta, de larga duración, almacenada por el sistema.
 <p>Clase Entidad</p>	Se utilizan para modelar información que posee una vida larga y que es a menudo persistente. Modelan la información y el comportamiento asociado de algún fenómeno o concepto, como una persona, un objeto o suceso del mundo real.

Fig. 3.1 Tipos de clases del análisis con una breve descripción.

### 3.2.1.1 Diagrama de clases Obtener Región de Interés

En la figura 3.2 se muestra el diagrama de clases del análisis para el caso de uso Obtener Regiones de Interés, se pueden apreciar las clases entidades Imagen y Circulo con una relación de composición ya que una imagen está compuesta por círculos. La clase control Procesamiento es la que maneja todos los pasos que se van a ejecutar para lograr el objetivo propuesto que es obtener las regiones de interés para su procesamiento.

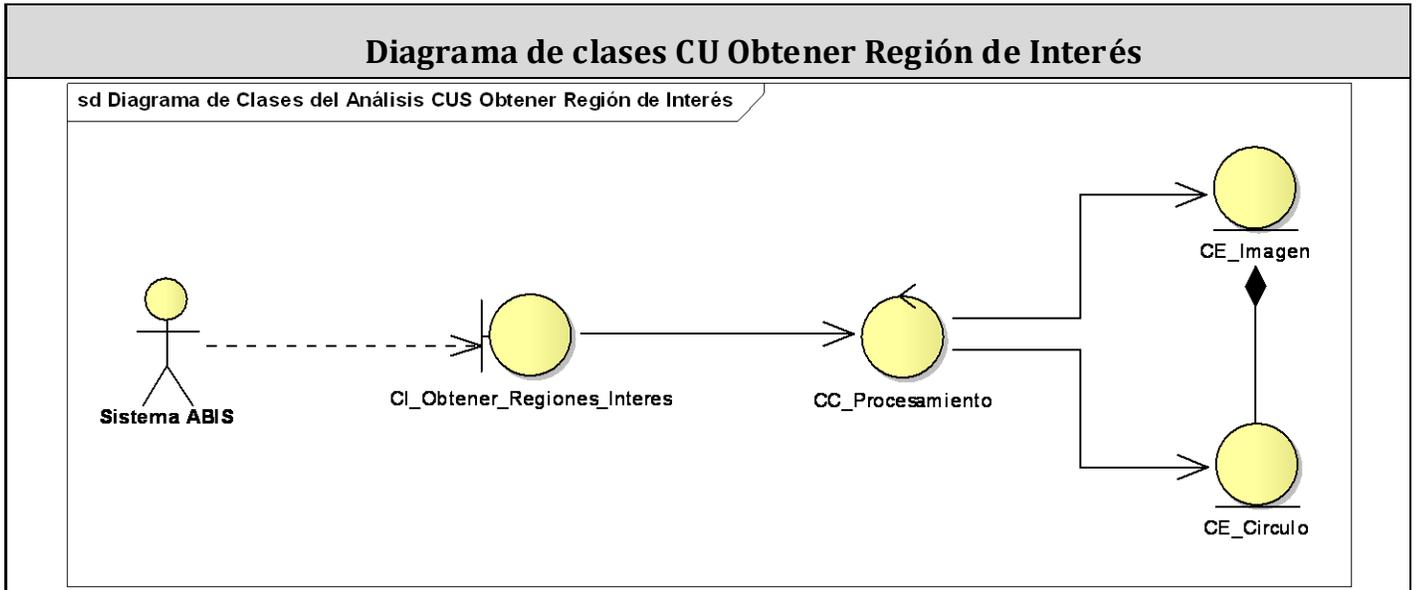


Fig. 3.2 Diagrama de clases del análisis para el caso de uso Obtener Regiones de Interés.

### 3.2.1.2 Diagrama de clases Obtener Rasgos Característicos

En la Fig. 3.3 se muestra el diagrama de clases del análisis para el caso de uso Obtener Rasgos Característicos. Se muestra la clase interfaz Obtener\_Rasgos\_Imagen que es la que se encargará de comunicarse con la clase control Procesamiento que manejará todas las rutinas para trabajar con las clases entidades Rasgos, Imagen y Círculo. Éstas últimas representan la información de larga vida con que se va a interactuar, tienen relaciones de composición ya que la imagen se compone por círculos y también por rasgos característicos.

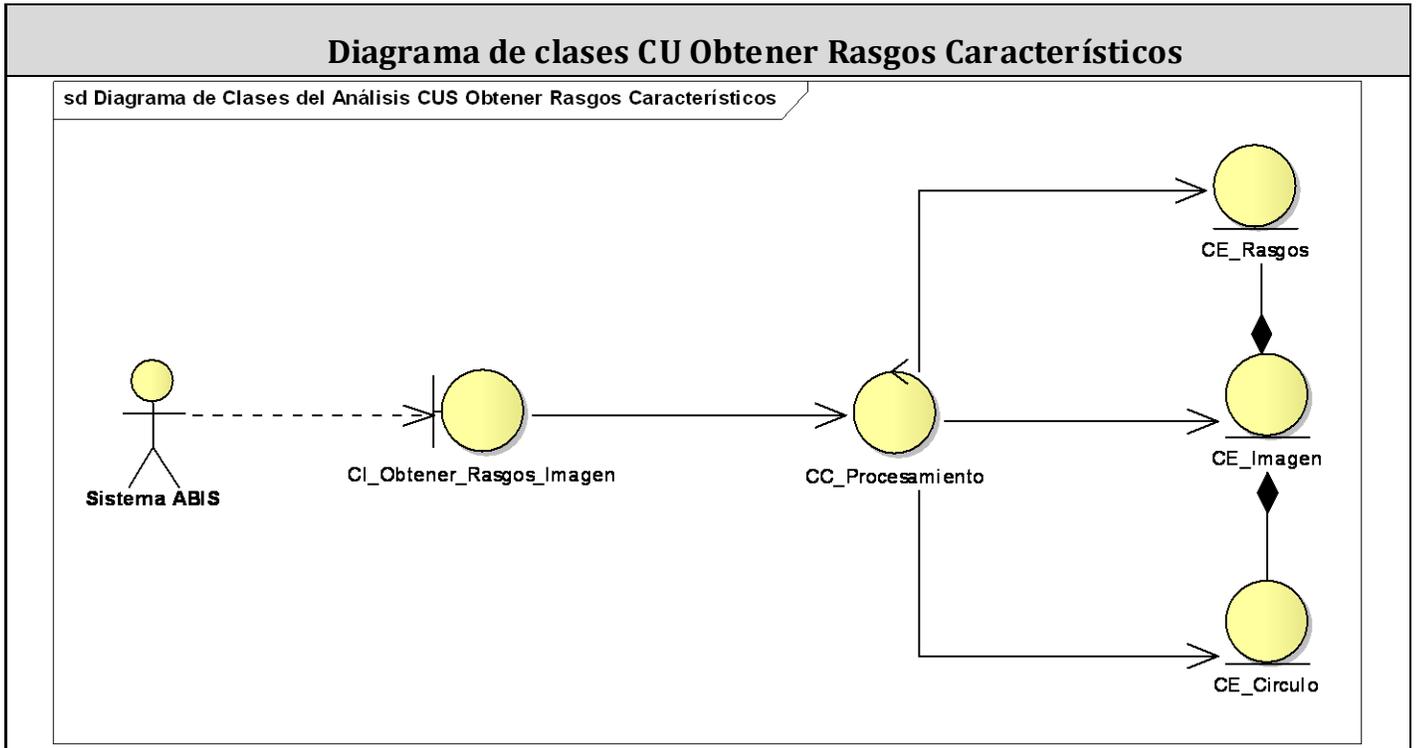


Fig. 3.3 Diagrama de clases del análisis Obtener Rasgos Característicos.

### 3.2.2 Diagramas de interacción

La secuencia de acciones en un caso de uso comienza cuando un actor invoca al caso de uso mediante el envío de algún tipo de mensaje al sistema. Si se considera el “interior” del sistema, un objeto de interfaz recibirá este mensaje del actor. El objeto de interfaz enviará a su vez un mensaje a algún otro objeto, y de esta forma los objetos implicados interactuarán para llevar a cabo el caso de uso. En el análisis se prefiere mostrar lo antes descrito con diagramas de colaboración ya que el objetivo fundamental es identificar requisitos y responsabilidades sobre los objetos.

En los diagramas de colaboración se muestran las relaciones entre objetos, se crean enlaces entre ellos y se añaden mensajes a esos enlaces. El nombre del enlace debe denotar el propósito del objeto que invoca en la interacción con el objeto invocado.

### 3.2.2.1 Diagrama de colaboración Obtener Regiones de Interés

En la figura 3.4 se muestra el diagrama de colaboración para el caso de uso Obtener Regiones de Interés. Se observan los principales mensajes intercambiados entre las clases para realizar el caso de uso en cuestión, se obtiene como resultado una lista de regiones (círculos) que servirán en la realización del próximo caso de uso.

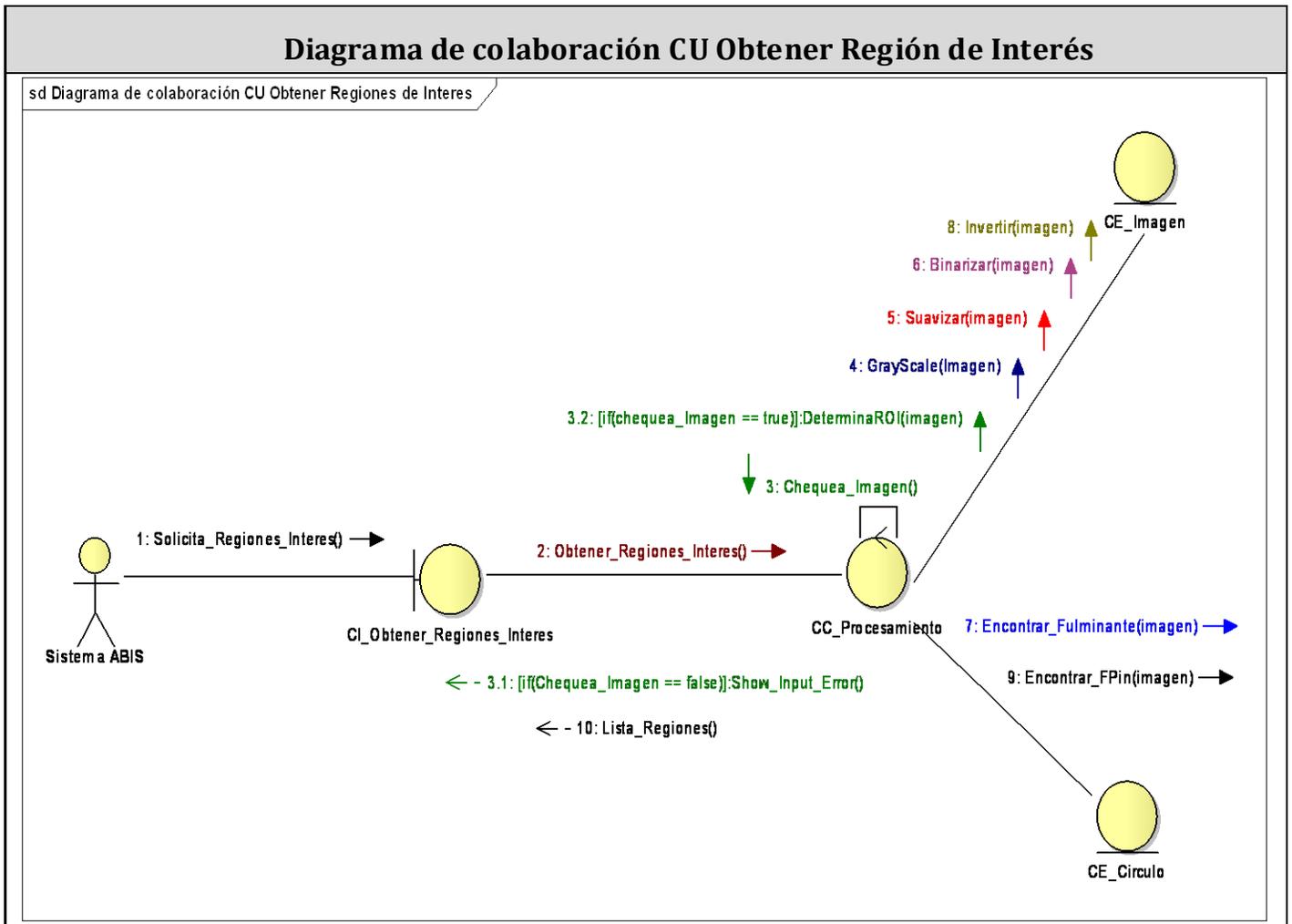


Fig. 3.4 Diagrama de colaboración para el caso de uso Obtener Regiones de Interés.

### 3.2.2.2 Diagrama de colaboración Obtener Rasgos Característicos

En la figura 3.5 se muestra el diagrama de colaboración para el caso de uso Obtener Rasgos Característicos. Se observan los principales mensajes intercambiados entre las clases para realizar el caso de uso en cuestión, se obtiene como resultado un vector característico con los rasgos distintivos de la marca de la aguja percutora y del espaldón en el fulminante de la Vaina.

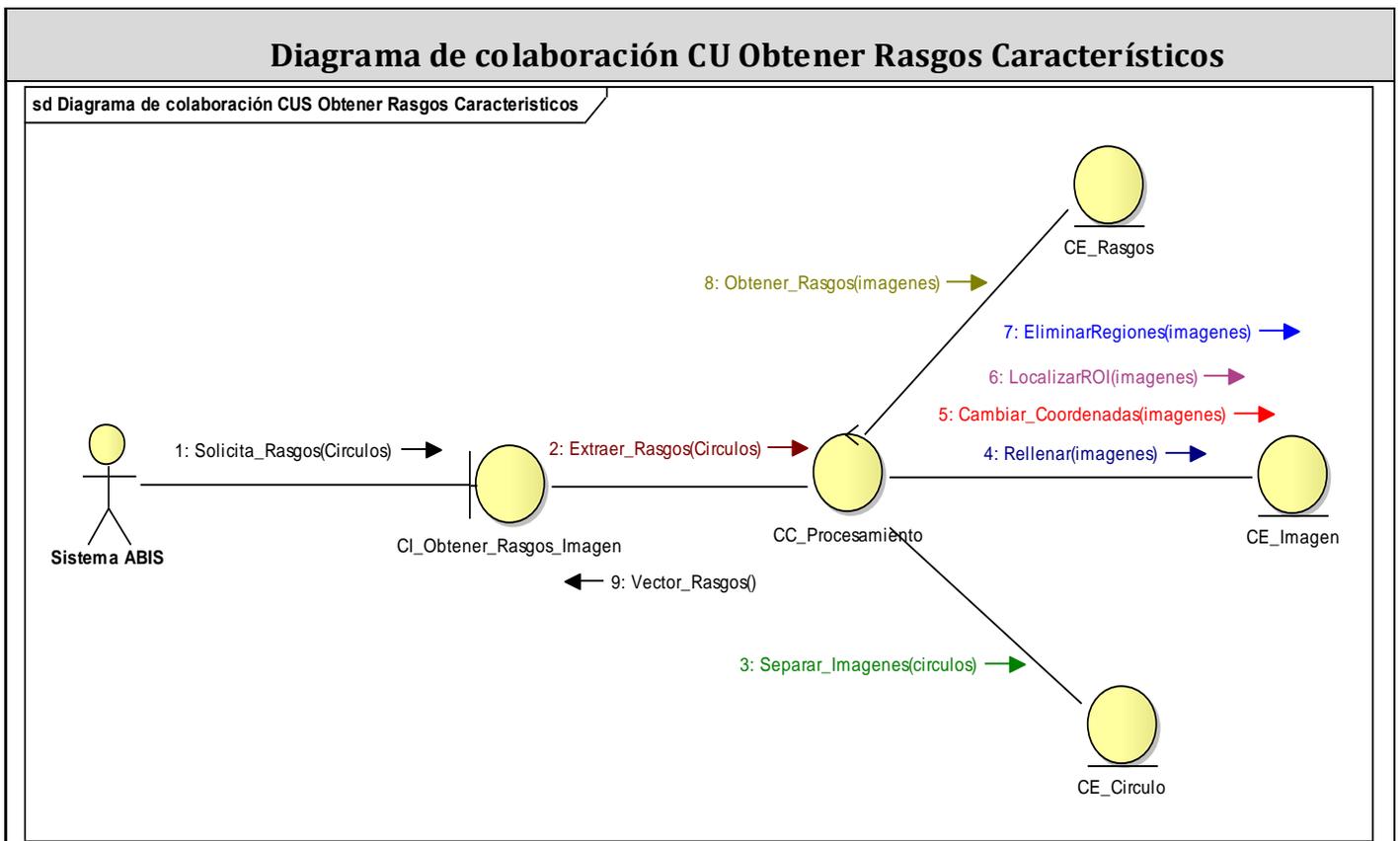


Fig. 3.5 Diagrama de colaboración para el caso de uso Obtener Rasgos Característicos.

## **3.3 Modelo de Diseño**

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas en el entorno de la implementación, tienen impacto en el sistema a considerar. Además, el modelo de diseño sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como una entrada fundamental de las actividades de implementación.

### **3.3.1 Clase del diseño**

Una clase de diseño es una abstracción sin costuras de una clase o construcción similar en la implementación del sistema. Esta abstracción es sin costuras en el siguiente sentido:

- El lenguaje especificado para especificar una clase del diseño es lo mismo que el lenguaje de programación. Consecuentemente, las operaciones, parámetros, atributos, tipos y demás son especificados con la utilización de la sintaxis del lenguaje de programación elegido.
- Las relaciones de aquellas clases del diseño implicadas con otras clases, a menudo tienen un significado directo cuando la clase es implementada.
- Los métodos (o lo que es lo mismo, las realizaciones de operaciones) de una clase del diseño, tienen correspondencia directa con el correspondiente método en la implementación de las clases (esto es, en el código).

Después de haber visto algunas de las principales características que tienen las clases del diseño, se verán algunos aspectos sobre el diagrama de clases del diseño y se mostrará el diagrama del sistema propuesto.

#### **3.3.1.1 Diagrama de clases del diseño**

Una clase de diseño y sus objetos, y de ese modo también los subsistemas que contienen las clases de diseño, a menudo participan en varias realizaciones de casos de uso. También puede darse el caso de algunas operaciones, atributos y asociaciones sobre una clase específica que son relevantes solo para una realización de caso de uso. Esto es importante para coordinar todos los requisitos que diferentes

realizaciones de casos de uso imponen a una clase, a sus objetos y a los subsistemas que contiene. En la figura 3.6 se muestra el diagrama de clases propuesto para la posterior implementación del componente.

La clase Procesamiento es la encargada de interactuar con el Sistema ABIS y brindarle a este las funcionalidades del componente. Posee los métodos necesarios para cumplir con los requisitos y además interactúa con otras clases de apoyo para el procesamiento de las imágenes.

La clase Círculo es la que guarda los datos necesarios de las regiones que serán procesadas, ella contiene datos importantes como centro, radio y área, que son de vital importancia para procesar la imagen y obtener los rasgos característicos.

La clase GLCM (Gray Level Cooccurrence Matrix) es la encargada de realizar las operaciones necesarias para procesar la imagen, para obtener su matriz de coocurrencia, que describe la relación espacial entre los píxeles de la imagen para aplicar varias transformaciones y fórmulas matemáticas que se explicarán más adelante para obtener los descriptores de textura de la imagen.

La clase Polar es la encargada de cambiar una imagen a coordenadas polares, esto es necesario para poder eliminar lo más posible las regiones que no son de interés en la imagen.

## Diagrama de clases del diseño

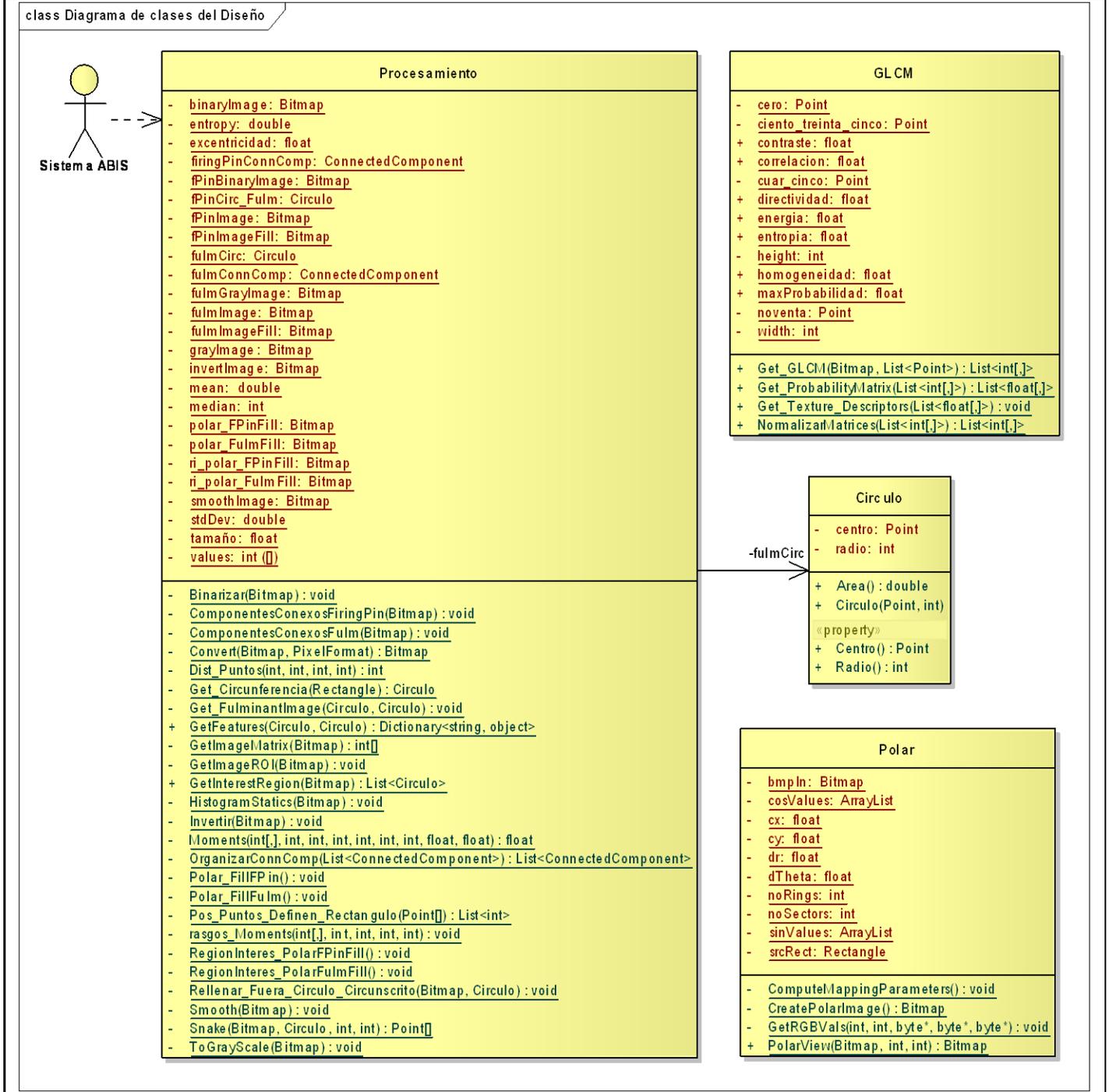


Fig. 3.6 Diagrama de clases del diseño para el sistema propuesto.

### 3.3.2 Diagramas de secuencia del diseño

La secuencia de acciones en un caso de uso comienza cuando un actor invoca el caso de uso mediante el envío de algún tipo de mensaje al sistema. Si se considera el “interior”, se tendrá algún objeto del diseño que recibe el mensaje del actor. Después el objeto de diseño llama a algún otro objeto, y de esta manera los objetos implicados interactúan para realizar y llevar a cabo el caso de uso. En el diseño es preferible representar esto con diagramas de secuencia ya que el centro de atención principal es el encontrar secuencias de interacciones detalladas y ordenadas en el tiempo.

#### 3.3.2.1 Diagrama de secuencia Obtener Región de Interés

En la Fig. 3.7 se muestra el diagrama de secuencia para el caso de uso Obtener Regiones de Interés. Se aprecia la interacción en el tiempo entre los objetos y los mensajes enviados entre estos. La clase Procesamiento hace la función de controladora e interfaz al mismo tiempo ya que no se tiene una interfaz física por ser una DLL<sup>5</sup>. El usuario interactúa con alguna funcionalidad de la biblioteca que es la que se encarga de realizar las operaciones necesarias para satisfacer el pedido.

---

<sup>5</sup> DLL es la sigla de **Dynamic Linking Library (Bibliotecas de Enlace Dinámico)**, término con el que se refiere a los archivos con código ejecutable que se cargan bajo demanda del programa por parte del sistema operativo.

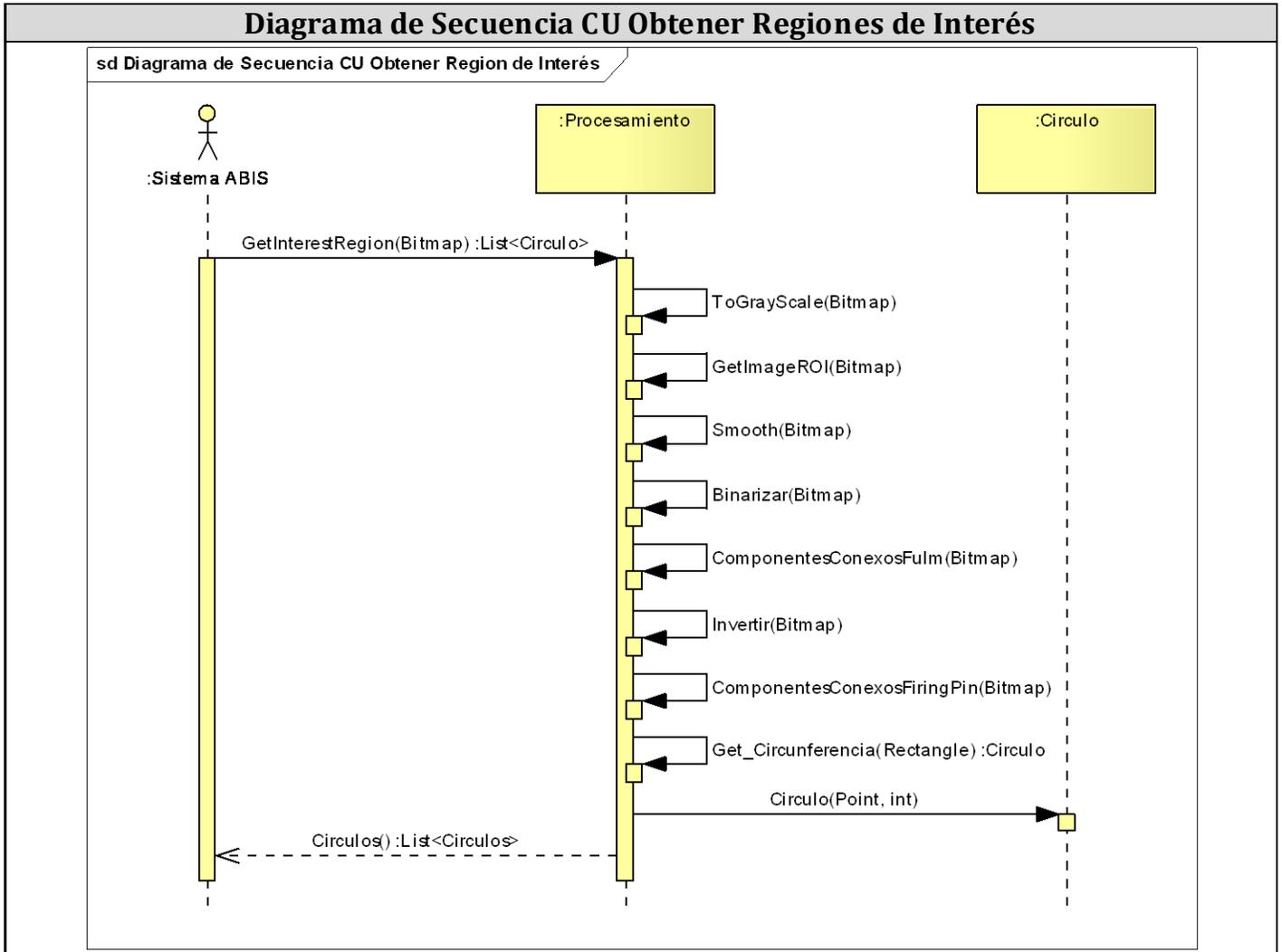


Fig. 3.7 Diagrama de secuencia CUS Obtener Regiones Interés.

### 3.3.2.2 Diagrama de Secuencia Obtener Rasgos Característicos

En la figura 3.8 se muestra el diagrama de secuencia para el caso de uso Obtener Rasgos Característicos.

## Diagrama de Secuencia CU Obtener Rasgos Característicos

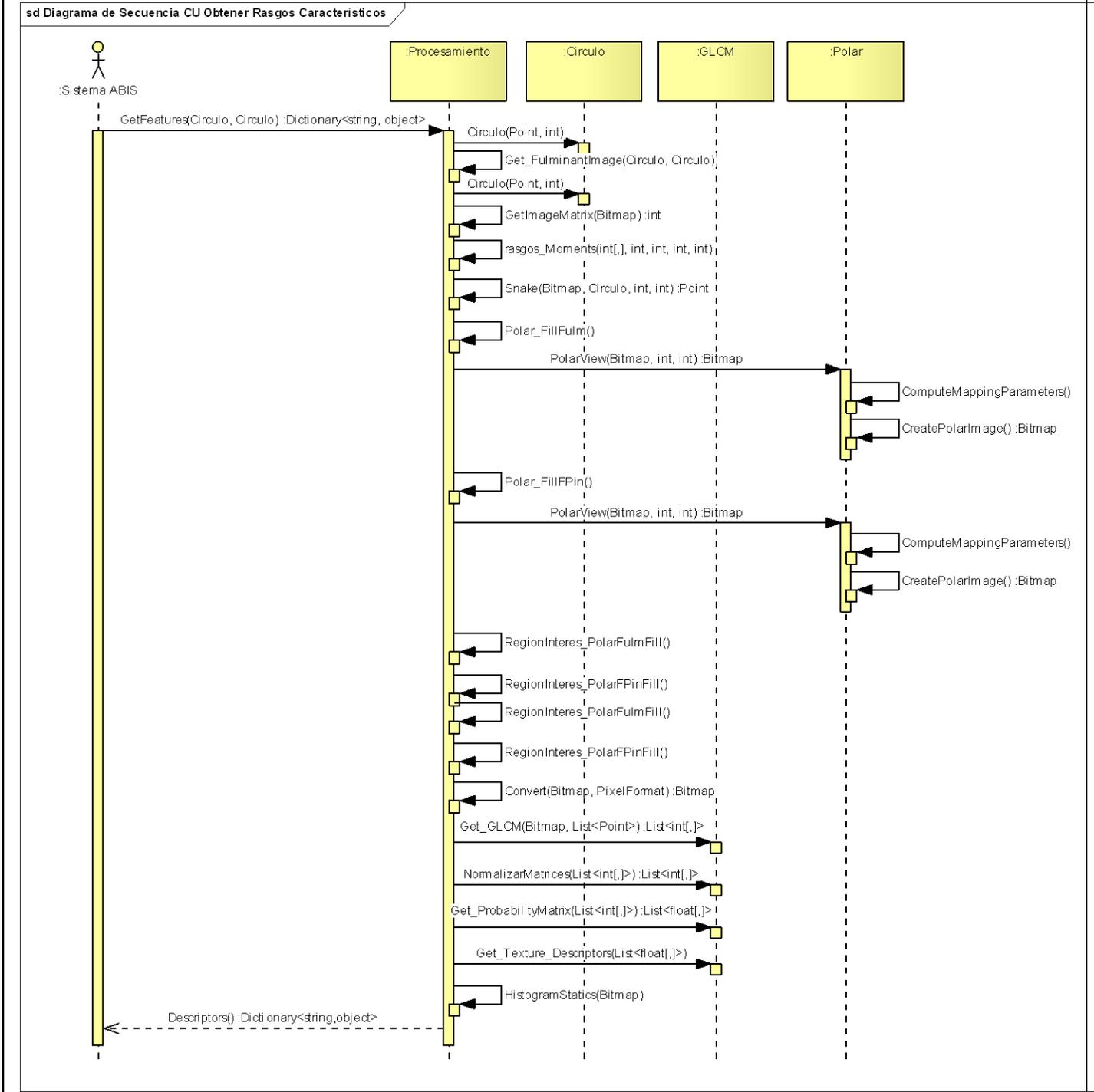


Fig. 3.8 Diagrama de secuencia para el caso de uso Obtener Rasgos Característicos.

### 3.3.3 Descripción de las clases.

En este epígrafe se expone una descripción más detallada de las clases presentes en el diagrama, se especifica el tipo de clase, sus atributos y métodos y cuáles son sus funcionalidades.

#### 3.3.3.1 Clase Procesamiento

<b>Nombre</b>	Procesamiento	
<b>Tipo de Clase</b>	Control	
<b>Atributo</b>	<b>Tipo</b>	
binaryImage	Bitmap	
firingPinConnComp	ConnectedComponent	
fPinCirc_Fulm	Circulo	
fPinImage	Bitmap	
fPinImageFill	Bitmap	
fulmCirc	Circulo	
fulmConnComp	ConnectedComponent	
fulmImage	Bitmap	
fulmImageFill	Bitmap	
grayImage	Bitmap	
invertImage	Bitmap	
fulmGrayImage	Bitmap	
fPinBinaryImage	Bitmap	
polar_FPinFill	Bitmap	
polar_FulmFill	Bitmap	
ri_polar_FPinFill	Bitmap	
ri_polar_FulmFill	Bitmap	
excentricidad	float	
smoothImage	Bitmap	

tamaño	float
values	Int[,]
entropy	double
mean	double
median	int
stdDev	double
<b>Para cada Responsabilidad</b>	
<b>Nombre</b>	Binarizar(smoothIma: Bitmap)
<b>Descripción</b>	Binariza la imagen en escala de grises
<b>Nombre</b>	ComponentesConexosFiringPin (pInvertImage: Bitmap)
<b>Descripción</b>	Halla los componentes conexos de la imagen invertida y guarda el de la aguja percutora.
<b>Nombre</b>	ComponentesConexosFulm( pBinaryImage: Bitmap)
<b>Descripción</b>	Halla los componentes conexos de la imagen binarizada y guarda el del fulminante.
<b>Nombre</b>	Convert( binIImg: Bitmap, pxFormat: PixelFormat)
<b>Descripción</b>	Convierte de un formato de pixel de la imagen a otro formato especificado.
<b>Nombre</b>	Dist_Puntos( p1: Point, centro: Point)
<b>Descripción</b>	Calcula la distancia entre dos puntos.
<b>Nombre</b>	Get_Circunferencia( _rect: Rectangle)
<b>Descripción</b>	Devuelve la circunferencia que contiene la marca dentro de dicho Rectángulo
<b>Nombre</b>	Get_FulminantImage( fPing: Circulo, Fulm: Circulo)
<b>Descripción</b>	Obtiene la imagen exacta del fulminante.
<b>Nombre</b>	GetFeatures( fPing: Circulo, Fulm: Circulo)
<b>Descripción</b>	Obtiene los descriptores de las marcas de la aguja percutora y del fulminante.
<b>Nombre</b>	GetImageMatrix( image: Bitmap)
<b>Descripción</b>	Obtiene la matriz asociada a una imagen binaria.

<b>Nombre</b>	GetImageROI( image: Bitmap)
<b>Descripción</b>	Obtiene una imagen centrada en el fulminante.
<b>Nombre</b>	GetInterestRegion( image: Bitmap)
<b>Descripción</b>	Extrae los círculos que comprenden las zonas de la aguja percutora y del fulminante.
<b>Nombre</b>	Invertir( pBinaryImage: Bitmap)
<b>Descripción</b>	Invierte la imagen binarizada.
<b>Nombre</b>	OrganizarConnComp( listConnComp: List<ConnectedComponent>)
<b>Descripción</b>	Organiza una lista de componentes conexas de mayor a menor según el área del rectángulo.
<b>Nombre</b>	Polar_FillFPin()
<b>Descripción</b>	Transforma la imagen de la marca de la aguja percutora rellena de negro de coordenadas cartesianas a coordenadas polares.
<b>Nombre</b>	Polar_FillFulm()
<b>Descripción</b>	Transforma la imagen del fulminante rellena de negro de coordenadas cartesianas a coordenadas polares.
<b>Nombre</b>	RegionInteres_PolarFPinFill()
<b>Descripción</b>	Determina la región de Interés de la imagen de la marca de la aguja percutora en coordenadas polares.
<b>Nombre</b>	RegionInteres_PolarFulmFill()
<b>Descripción</b>	Determina la región de Interés de la imagen del fulminante en coordenadas polares.
<b>Nombre</b>	Smooth(curlImage: Bitmap)
<b>Descripción</b>	Suaviza la imagen.
<b>Nombre</b>	ToGrayScale( Image: Bitmap )
<b>Descripción</b>	Convierte la imagen a escala de Grises.
<b>Nombre</b>	HistogramStatics(image: Bitmap)
<b>Descripción</b>	Calcula datos estadísticos del histograma de una imagen.
<b>Nombre</b>	Moments( imageMatrix: int[,], p: int, q: int, n1: int, m1: int, n2: int, m2:

	int, x_: float, y_: float)
<b>Descripción</b>	Obtiene los momentos a partir de la matriz de la imagen binarizada
<b>Nombre</b>	Pos_Puntos_Definen_Rectangulo(puntos: Point[])
<b>Descripción</b>	Devuelve una lista de enteros con la posición de los puntos que determinan el rectángulo de la marca.
<b>Nombre</b>	rasgos_Moments( imageMatrix: int[,], n1: int, m1: int, n2: int, m2: int)
<b>Descripción</b>	Calcula descriptores morfológicos a partir de los momentos.
<b>Nombre</b>	Rellenar_Fuera_Circulo_Circunscrito( image: Bitmap, circ: Circulo)
<b>Descripción</b>	Rellena la parte de afuera del círculo que está inscrito en el rectángulo.
<b>Nombre</b>	Snake( image: Bitmap, fPin_circulo: Circulo, cant_puntos: int, cant_iteraciones: int)
<b>Descripción</b>	Determina el contorno de la marca de la aguja percutora con la utilización de la técnica Snake.

### 3.3.3.2 Clase Círculo

<b>Nombre</b>	Circulo
<b>Tipo de Clase</b>	Entidad
<b>Atributo</b>	<b>Tipo</b>
centro	Point
radio	int
<b>Para cada Responsabilidad</b>	
<b>Nombre</b>	Area ()
<b>Descripción</b>	Calcula el área del círculo.
<b>Nombre</b>	Centro
<b>Descripción</b>	Retorna o modifica el valor del centro del círculo.
<b>Nombre</b>	Circulo (_centro: Point, _radio: int)
<b>Descripción</b>	Constructor de la clase.
<b>Nombre</b>	Radio
<b>Descripción</b>	Retorna o modifica el valor del radio del círculo.

### 3.3.3.3 Clase GLCM

<b>Nombre</b>	GLCM
<b>Tipo de Clase</b>	Controladora
<b>Atributo</b>	<b>Tipo</b>
cero	Point
ciento_treinta_cinco	Point
contraste	float
correlacion	float
cuar_cinco	Point
directividad	float
energia	float
entropia	float
height	int
homogeneidad	float
imageMatrix	Int
maxProbabilidad	float
noventa	Point
width	int
<b>Para cada Responsabilidad</b>	
<b>Nombre</b>	Get_GLCM(image: Bitmap, offset: List<Point>)
<b>Descripción</b>	Obtiene las matrices de coocurrencia para una imagen en las cuatro direcciones, 0°,45°,90° y 135°.
<b>Nombre</b>	NormalizarMatrices(glcms: List<int[,]>)
<b>Descripción</b>	Normaliza las matrices de coocurrencia.
<b>Nombre</b>	Get_ProbabilityMatrix(glcms: List<int[,]>)
<b>Descripción</b>	Calcula las probabilidades de las matrices de coocurrencia.
<b>Nombre</b>	Get_Texture_Descriptors(ProbGLCMS: List<float[,]>)
<b>Descripción</b>	Calcula los descriptores de textura.

### 3.3.3.4 Clase Polar

<b>Nombre</b>	Polar	
<b>Tipo de Clase</b>	Controladora	
<b>Atributo</b>	<b>Tipo</b>	
bmpIn	Bitmap	
cosValues	ArrayList	
cx	float	
cy	float	
dr	float	
dTheta	float	
noRings	int	
noSectors	int	
sinValues	ArrayList	
srcRect	Rectangle	
<b>Para cada Responsabilidad</b>		
<b>Nombre</b>	PolarView(image: Bitmap, pNoSectors: int, pNoRings: int)	
<b>Descripción</b>	Constructor de la clase.	
<b>Nombre</b>	ComputeMappingParameters()	
<b>Descripción</b>	Hace los cálculos de las variables necesarias para la conversión.	
<b>Nombre</b>	CreatePolarImage()	
<b>Descripción</b>	Crea la imagen en coordenadas polares.	
<b>Nombre</b>	GetRGBVals(x: int, y: int, [inout] red: byte, [inout] green: byte, [inout]	
<b>Descripción</b>	Obtiene los valores de los píxeles red, green y blue (RGB).	

## Conclusiones

En este capítulo se ha realizado la descripción y justificación de la arquitectura utilizada. Se expuso el modelo de análisis, se mostraron los diagramas de clases y colaboraciones para cada realización de los casos de uso. Se mostró el modelo de diseño, se detalló el diagrama de clases del diseño junto con los diagramas de secuencia para cada caso de uso. Se hizo una descripción de las clases utilizadas y se especificaron los atributos y operaciones de cada una.

# CAPÍTULO 4: Implementación

---

En este capítulo se explica lo esencial sobre las técnicas utilizadas para implementar el sistema propuesto, se exponen las características principales del flujo de trabajo implementación y el diagrama de componentes.

## 4.1 Técnicas utilizadas

A través de las técnicas de PDI (véase epígrafe 1.3) que han surgido a lo largo de la historia, es posible localizar zonas y objetos que están dentro de las imágenes, estas pueden ser caracterizadas a través de los descriptores (véase epígrafe 1.2). En este apartado se exponen los pasos y técnicas empleadas para lograr el objetivo de caracterizar la imagen digital del culote de un proyectil disparado por un arma de fuego.

### 4.1.1 Procesamiento de imágenes balísticas

Para procesar las imágenes balísticas se han definido una serie de pasos que procesan la imagen balística para obtener sus regiones de interés, y de estas regiones sus descriptores. Es bueno destacar que los pasos que se siguen son para obtener las regiones y características de culotes de vainas de fuego central (centerfire). Estas son las que la aguja percutora hace contacto en la parte central del culote, en estos tipos de vainas se delimita muy bien la frontera entre el borde exterior y el fulminante, que es el anillo central del culote. El otro tipo de vaina es la de fuego anular, en la que la carga explosiva se encuentra en el borde exterior y por lo tanto la aguja percutora deja su marca en el anillo externo del culote de la vaina, estas son conocidas en la literatura como vainas de tipo “rimfire” (22).

### 4.1.2 Obtención de las regiones de interés

Para comenzar con el pre-procesado de la imagen, lo primero que se realiza es llevar a escala de grises el gráfico a tratar (Fig. 4.1). Después de la imagen estar en escala de grises, se hace una aproximación al área en donde se encuentra el fulminante, y por ende, la marca de la aguja percutora. Esto se realiza recortando la imagen, que queda de la manera que se muestra en la Fig. 4.2.



Fig. 4.1 Imagen en escala de grises.

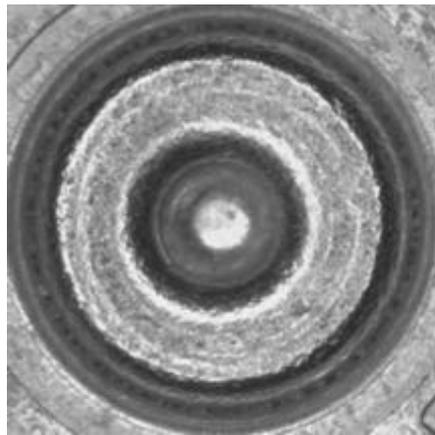


Fig. 4.2 Imagen centrada en el fulminante, donde se encuentran las regiones de interés.

La imagen en escala de grises es suavizada con el uso de una matriz de convolución de 9x9 (23) . La imagen queda como se muestra en la Fig. 4.3.

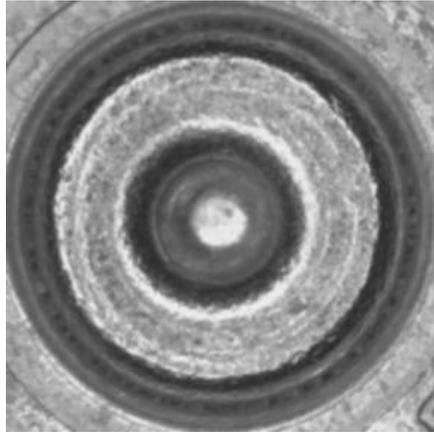


Fig. 4.3 Imagen suavizada.

Después que la imagen se suaviza para eliminar ruido, se binariza (Fig. 4.4) para delimitar bien las regiones del fulminante y de la marca de la aguja percutora.



Fig. 4.4 Imagen binarizada.

A la imagen se le aplica el algoritmo para definir las regiones conexas para precisar cuál es el rectángulo que define el fulminante (Fig. 4.5) y a partir de este obtener el círculo que delimita el fulminante.

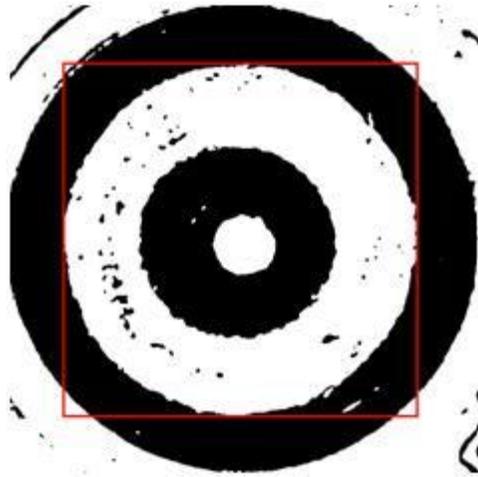


Fig. 4.5 Imagen que muestra el rectángulo que comprende el fulminante después de haber aplicado el algoritmo para encontrar las regiones conexas en la imagen.

Después que se obtiene donde está ubicada el área del fulminante se invierte la imagen (Fig. 4.6) para con la aplicación de las regiones conexas delimitar donde es que se encuentra la marca de la aguja percutora (Fig. 4.7).



Fig. 4.6 Imagen binarizada invertida para localizar la marca de la aguja percutora.

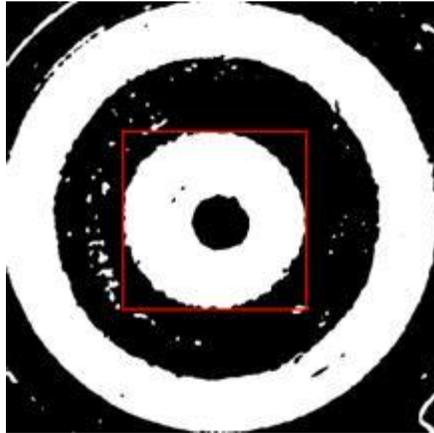


Fig. 4.7 Imagen que muestra la marca de la aguja percutora encerrada por un rectángulo después de haber aplicado las regiones conexas.

Hasta aquí se ha obtenido los círculos que encierran el fulminante y la marca de la aguja percutora, la Fig. 4.8 muestra una captura de pantalla de una aplicación que le pide al componente las regiones de interés de la imagen que se quiere procesar.



Fig. 4.8 Captura de pantalla de una aplicación que solicitó los servicios del componente para obtener las regiones de interés.

### 4.1.3 Obtención de los rasgos característicos

Se ha expuesto cómo se pueden obtener las regiones de interés de la imagen, este apartado muestra como esas regiones son procesadas para obtener los descriptores del fulminante y de la marca de la aguja percutora. Primero, con las regiones que se obtuvieron, se separan las imágenes del fulminante y de la aguja percutora para procesarlas por separado y obtener los rasgos de cada una de ellas (Figs. 4.9 y 4.10).

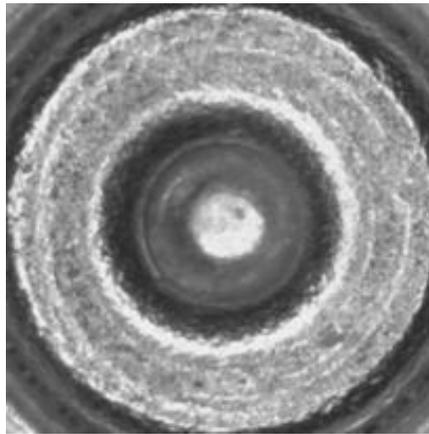


Fig. 4.9 Imagen del fulminante, nótese que la imagen está enmarcada exactamente en el fulminante.

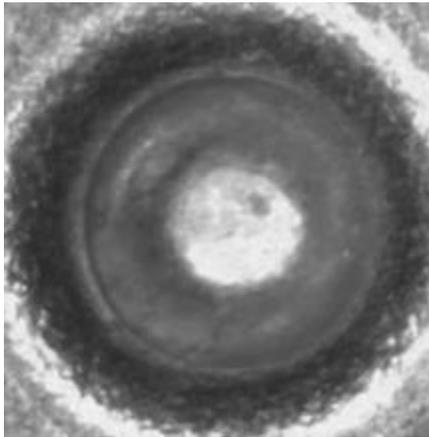


Fig. 4.10 Imagen de la marca de la aguja percutora.

A través de la técnica denominada Snake, se determina el contorno de la marca de la aguja percutora, esto permitirá después obtener descriptores del contorno de la marca. El Snake parte de un contorno inicial (círculo de la marca de la aguja percutora) y obtiene los puntos del contorno del objeto en estudio.

Las imágenes obtenidas en el paso anterior, son rellenadas de negro en las regiones que no son de interés para el procesado, para esto se tienen en cuenta las regiones obtenidas en la primera funcionalidad (ver epígrafe anterior) y el contorno obtenido del Snake. Se procesa la imagen pixel a pixel y según la distancia del que esté en análisis al centro del círculo, se pone o no en negro el pixel en cuestión. El contorno de la marca de la aguja percutora forma un polígono que es rellenado de negro por dentro o por fuera en dependencia de si está en proceso la marca el fulminante o la marca de la aguja percutora. Las imágenes resultantes se muestran en las figuras 4.11 y 4.12.

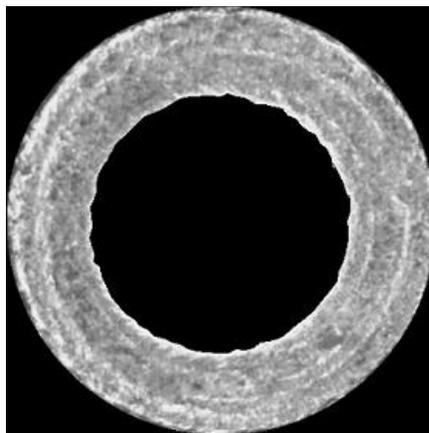


Fig. 4.11 Imagen del fulminante con las regiones que no son de interés rellenas de negro.

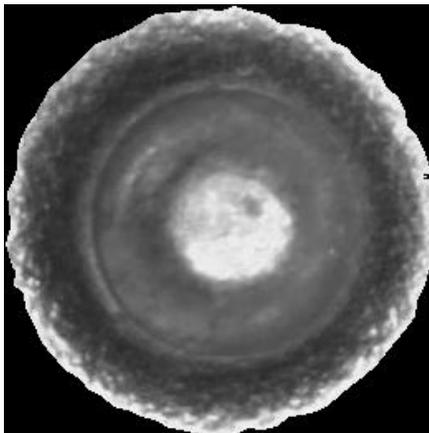


Fig. 4.12 Imagen de la marca de la aguja percutora con las regiones que no son de interés rellenas de negro.

Para un mejor procesado de las imágenes, lo mejor es eliminar lo más posible las regiones de negro, ya que no aportan información de utilidad para que la imagen sea caracterizada. Esto se logra al convertir las imágenes de coordenadas cartesianas a polares (24). Igualmente espaciados, se dibujan círculos concéntricos centrados en el centro de la imagen, y una serie de sectores, igualmente espaciados. Píxeles en los puntos de intersección de estos círculos y líneas radiales se trazan en una malla rectangular, y la imagen resultante es una vista polar. En un mapeo de coordenadas log-polares, los radios de los círculos concéntricos varían en una escala logarítmica (25).

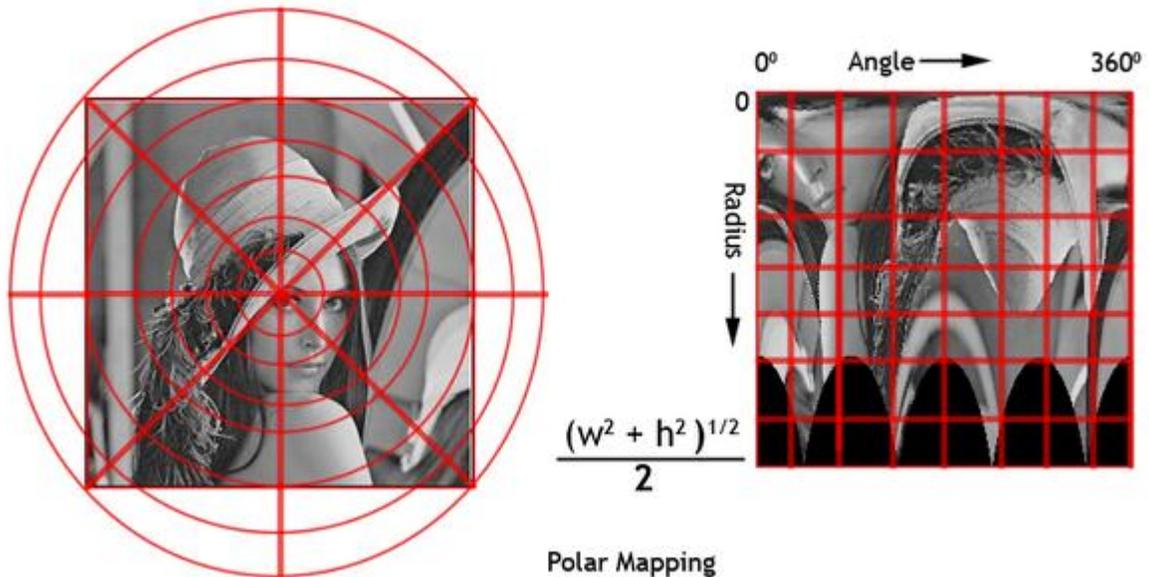


Fig. 4.14 Geometría básica de la conversión de coordenadas cartesianas a polar.

En las figuras 4.14 y 4.15 se muestra cómo quedan las imágenes en coordenadas polares, las regiones de negro quedan unidas para que sea posible eliminar gran parte de estas.

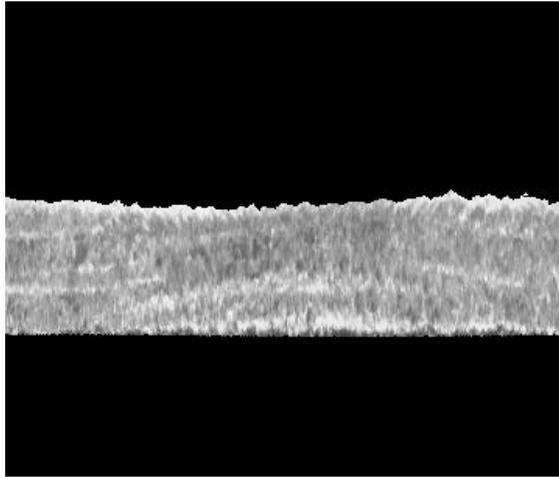


Fig. 4.14 Imagen del fulminante en coordenadas polares.



Fig. 4.15 Imagen de la marca de la aguja percutora en coordenadas polares.

Después de las imágenes estar en coordenadas polares, es necesario eliminar la región en negro que no es de interés. Para esto se binarizan las imágenes y con el auxilio de las regiones conexas, se eliminan lo más posible las regiones de negro. Las figuras 4.16 y 4.17 muestran las imágenes después de aplicarle la transformación y reducción.

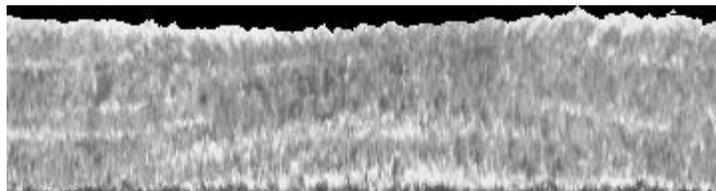


Fig. 4.16 Imagen del fulminante lista para obtener sus rasgos característicos.

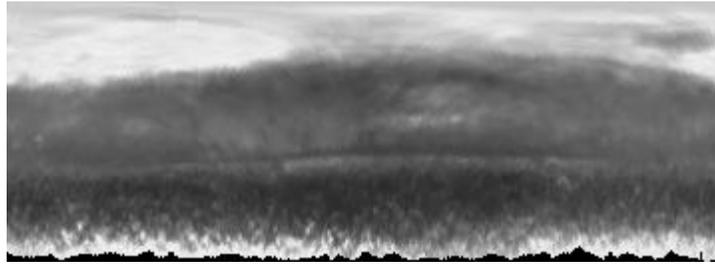


Fig. 4.17 Imagen de la marca de la aguja percutora lista para obtener sus descriptores.

Para la obtención de los descriptores de textura se analiza el histograma y la matriz de coocurrencia o histograma de segundo orden, conocida en la literatura como Gray Level Coocurrence Matrix (GLCM) (26). El histograma es un medidor de la frecuencia con que aparece en una imagen un determinado nivel de luminosidad. Con esta información se pueden extraer algunos datos estadísticos como: media, mediana, desviación estándar y entropía.

Media (en inglés mean) es el promedio de los niveles de luminosidad, está definido por la siguiente expresión:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

Donde  $n$  es el número de elementos de la muestra (256) y  $x$  el valor en cada posición de la muestra.

La desviación estándar es la media de las diferencias de todos los píxeles sobre el promedio. Se suman todas las diferencias elevadas al cuadrado (para que cuente lo mismo la desviación positiva que la negativa) y después se calcula la raíz cuadrada de la media. Es un indicativo de si la imagen está poca o muy contrastada. La desviación estándar está definida por la ecuación:

$$s = \left( \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \right)^{\frac{1}{2}}$$

Si se considera una variable discreta  $X$  cuyas observaciones en una tabla estadística han sido ordenadas de menor a mayor. Se llama mediana (median) al primer valor de la variable que deja por debajo de sí al 50 % de las observaciones. Por tanto, si  $n$  es el número de observaciones, la mediana corresponderá a la observación  $[n/2]+1$  (27).

La entropía es una medida estadística de la aleatoriedad que puede ser utilizada para caracterizar la textura de una imagen. La entropía se define como:

$$-\sum(p * \log(p))$$

Donde p contiene la frecuencia de aparición de un nivel de color en el histograma.

Con la matriz de coocurrencia se pueden obtener otros datos que describen la textura de la imagen (28). Para crear la matriz, se calcula con qué frecuencia un pixel con intensidad i, se relaciona con otro pixel de intensidad j en una relación espacial específica. En la matriz de coocurrencia obtenida, cada elemento (i, j) es la suma del número de veces que el pixel con valor i tuvo relación con el pixel j en el espacio determinado. Una imagen con una intensidad de color de 256 genera una GLCM de 256x256, esto es muy costoso en cuanto a recursos de la computadora, por lo que la imagen se escala a 8 niveles de grises para que la GLCM resultante sea más pequeña (8x8). El número de niveles de grises de la imagen, será el tamaño de la matriz de coocurrencia. La matriz es normalizada y después se expresa la matriz como probabilidad, esto está definido por el número de veces que un evento ocurre dividido por el número total de posibles eventos (26).

De la GLCM se pueden derivar datos estadísticos como son:

- **Contraste:** proporciona información acerca de las variaciones bruscas de color en la imagen. La extracción de este descriptor está dada por la siguiente expresión matemática:

$$\sum_{i,j} |i - j|^2 p(i, j)$$

Donde i y j es la fila y la columna de la matriz de probabilidades y p (i, j) es la probabilidad en la intersección de i con j.

- **Correlación:** Es una medida de cómo está relacionado un pixel con su vecino en toda la imagen. Es una medida de la probabilidad que mide la relación entre las diferentes intensidades de los colores. Está dada por las ecuaciones:

$$C = \sum_{i=0}^8 \sum_{j=0}^8 \frac{(i - \mu_x)(j - \mu_y)}{\sigma_x \sigma_y} p(i, j)$$

$$\mu_x = \sum_{i=0}^8 \sum_{j=0}^8 i p(i, j), \quad \mu_y = \sum_{i=0}^8 \sum_{j=0}^8 j p(i, j), \quad (\mu_x = \mu_y)$$

$$\sigma_x = \sqrt{\sum_{i=0}^8 \sum_{j=0}^8 (i - \mu_x)^2 p(i, j)}, \quad \sigma_y = \sqrt{\sum_{i=0}^8 \sum_{j=0}^8 (j - \mu_y)^2 p(i, j)}, \quad (\sigma_x = \sigma_y)$$

Nótese que la sumatoria es hasta 8 porque los niveles de grises se escalaron hasta esa cantidad de niveles.  $x$  &  $y$  son filas y columnas y  $p$  la probabilidad.

- **Energía o Uniformidad:** También es conocido como Segundo Momento Angular (ASM en inglés) y está dada por la ecuación:

$$\sum_{i,j} p(i, j)^2$$

Donde  $p$  es la probabilidad en la intersección (fila, columna)  $(i, j)$ .

- **Homogeneidad:** El descriptor de homogeneidad local proporciona información sobre la regularidad local de la textura. La descripción matemática de este descriptor está dada por la siguiente ecuación:

$$HL = \sum_i \sum_j \frac{c_{ij}}{1 + (i - j)^2}$$

- **Entropía:** Ya se vio la entropía en los descriptores obtenidos a partir del histograma de la imagen, esta también mide la aleatoriedad pero cambia en que la fórmula se aplica sobre una matriz y no sobre un arreglo unidimensional como lo es el histograma. La ecuación vendría dada por:

$$-\sum_i \sum_j c_{ij} \log c_{ij}$$

- **Máxima Probabilidad:** El descriptor de máxima probabilidad suministra información sobre la respuesta más fuerte de un par de píxeles en la matriz de coocurrencia. La descripción matemática de este descriptor es la siguiente:

$$mp = \max_{ij} (c_{ij})$$

Se puede inferir que la máxima probabilidad no es más que el mayor porcentaje de probabilidad de la matriz, esto es, la combinación de píxeles que más ocurre.

- **Directividad:** El descriptor de Directividad extrae información sobre patrones de dirección en la textura. Definido matemáticamente con la expresión:

$$Dir = \sum_i c_{ii}$$

Esto es la sumatoria de todos los valores de la diagonal de la matriz de probabilidades.

Para la obtención de los descriptores de contorno se parte de los puntos que aporta el Snake. Con estos puntos se obtiene:

- **Descriptores de Fourier:**

Una buena forma de caracterizar un objeto es a través de los descriptores de Fourier debido a las bondades que ofrecen los mismos. Sea un contorno digital cerrado, si dicho contorno se recorre a favor de las manecillas del reloj su forma se puede describir por las coordenadas  $x(t)$ ,  $y(t)$ , para  $t = 0, 1, \dots, N-1$ . De forma más general, cada par de coordenadas se puede tratar como un número complejo de forma tal que:  $f(t) = x(t) + jy(t)$ , para  $t = 0, 1, 2, \dots, N-1$  donde se ha tomado el eje  $x$  como el eje real y el eje  $y$  como el eje imaginario de una serie de números complejos. Obsérvese que esta notación ha convertido un problema bidimensional en uno unidimensional.

Tal función puede ser representada a través de la transformada unidimensional discreta de Fourier, la cual viene dada por la ecuación:

$$F(u) = \frac{1}{N} \sum_{n=0}^{N-1} f(n) \exp \left[ -j2\pi \left( \frac{nu}{N} \right) \right] \quad [1]$$

A los coeficientes complejos  $F(u)$  se les denominan *descriptores de Fourier* del contorno. La transformada inversa de Fourier restaura completamente el contorno  $f(n)$ , es decir:

$$f(n) = \frac{1}{N} \sum_{u=0}^{N-1} F(u) \exp \left[ j2\pi \left( \frac{nu}{N} \right) \right] \quad [2]$$

La representación acorde a los descriptores de Fourier dada por las expresiones [1] y [2] tiene algunas propiedades muy interesantes. Por ejemplo, el coeficiente  $F(0)$  representa el centro de gravedad de la curva. Los coeficientes de Fourier  $F(u)$  tienen índices pequeños para variaciones de formas suaves, mientras que sus índices son mayores para variaciones rápidas.

- Una traslación en las coordenadas de la curva, por ejemplo:

$$f_t(n) = f(n) + d_0, \quad d_0 = x_0 + jy_0$$

afecta solamente al coeficiente  $F(0)$  de la representación, es decir:

$$F_t(0) = F(0) + d_0$$

- Una rotación de las coordenadas de la curva por el ángulo  $\theta$ , o sea:

$$f_r(n) = f(n) e^{i\theta}$$

produce un corrimiento de fase de los coeficientes de la transformada igual a la cantidad:

$$F_r(u) = F(u) e^{i\theta}$$

- Una operación de multiplicación por un factor  $w$  con respecto a un sistema de coordenadas que tiene el origen en el centro de gravedad de la curva da por resultado una multiplicación de los coeficientes de Fourier igual a la cantidad:

$$f_s(n) = wf(n)$$

$$F_s(u) = wF(u)$$

- Por último, un cambio en el punto de comienzo de la curva, es decir:

$$f_t(n) = f(n_1 - n_0)$$

produce una *modulación* de los coeficientes de Fourier, o sea:

$$F_t(u) = F(u) e^{-i2\pi n_0 u/N}$$

Otro aspecto a destacar en los descriptores de Fourier es lo siguiente: Supóngase que en vez de escoger todos los  $F(u)$  solamente se utilizan los primeros  $M$  coeficientes, lo cual es equivalente a hacer  $F(u) = 0$  para  $u > M-1$  en la expresión (2). Esto da por resultado la aproximación siguiente:

$$\hat{f}(n) = \frac{1}{N} \sum_{n=0}^{M-1} F(u) \exp\left[\frac{j2\pi u n}{N}\right]$$

Como se puede observar en la expresión, aunque sólo se utilizan  $M$  términos para obtener cada componente de  $\hat{f}(n)$ ,  $M$  sigue en el intervalo de 0 a  $N-1$ , es decir; lo único que ha ocurrido es que ahora no se utilizan tantos términos en la reconstrucción del contorno. Como es lógico, cuanto más pequeño sea  $M$  la resolución del algoritmo será menor y por tanto se perderán más detalles del contorno. No obstante, en dependencia de la aplicación, en algunos casos se pueden utilizar unos cuantos descriptores de Fourier para captar la esencia de un contorno y a partir de éstos obtener información para el reconocimiento.



$$\mu_{pq} = \sum_i \sum_j (i - \bar{x})^p (j - \bar{y})^q \quad [7]$$

El momento  $m_{00}$  representa el área del objeto binario.

A partir de la expresión [5] se puede definir de otra forma la **excentricidad**. En efecto, ésta viene dada ahora acorde a la siguiente expresión,

$$\epsilon = \frac{(\mu_{02} - \mu_{20})^2 + 4\mu_{11}}{A} \quad [8]$$

donde  $A$  es el área del objeto.

También dada la expresión [5] es posible definir un descriptor de **tamaño** para una región, el cual se puede expresar como sigue,

$$S = \mu_{02} + \mu_{20} \quad [9]$$

## 4.2 Implementación

En la implementación se comienza con el resultado del diseño y se implementa el sistema en términos de componentes: ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares.

El principal propósito de la implementación es desarrollar la arquitectura y el sistema como un todo. Los propósitos de la implementación son: planificar las integraciones de sistema necesarias en cada iteración, a través de un enfoque incremental; implementar las clases y subsistemas encontrados durante el diseño. En particular, las clases se implementan como componentes de fichero que contienen código fuente. Probar los componentes individualmente, y a continuación integrarlos compilándolos y enlazándolos en uno o más ejecutables, antes de ser enviados para ser integrados y llevar a cabo las comprobaciones de sistema.

## 4.2.1 Diagrama de componentes

Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño, algunos estereotipos estándar de componentes son los siguientes:

- <<executable>> es un programa que puede ser ejecutado en un nodo.
- <<file>> es un fichero que contiene código fuente o datos.
- <<library>> es una librería estática o dinámica.
- <<table>> es una tabla de una base de datos.
- <<document>> es un documento.

En la figura 4.19 se muestra el diagrama de componentes para el sistema propuesto, ExtracRasgos.dll es el componente para extraer los rasgos característicos en culotes de proyectiles, esta librería usa ImageProcessing.dll, una librería con filtros y algoritmos para el trabajo con imágenes, la librería openCV que es utilizada para aplicar la técnica del Snake y la librería Fourier que tiene las funcionalidades necesarias para la obtención de los descriptores de Fourier.

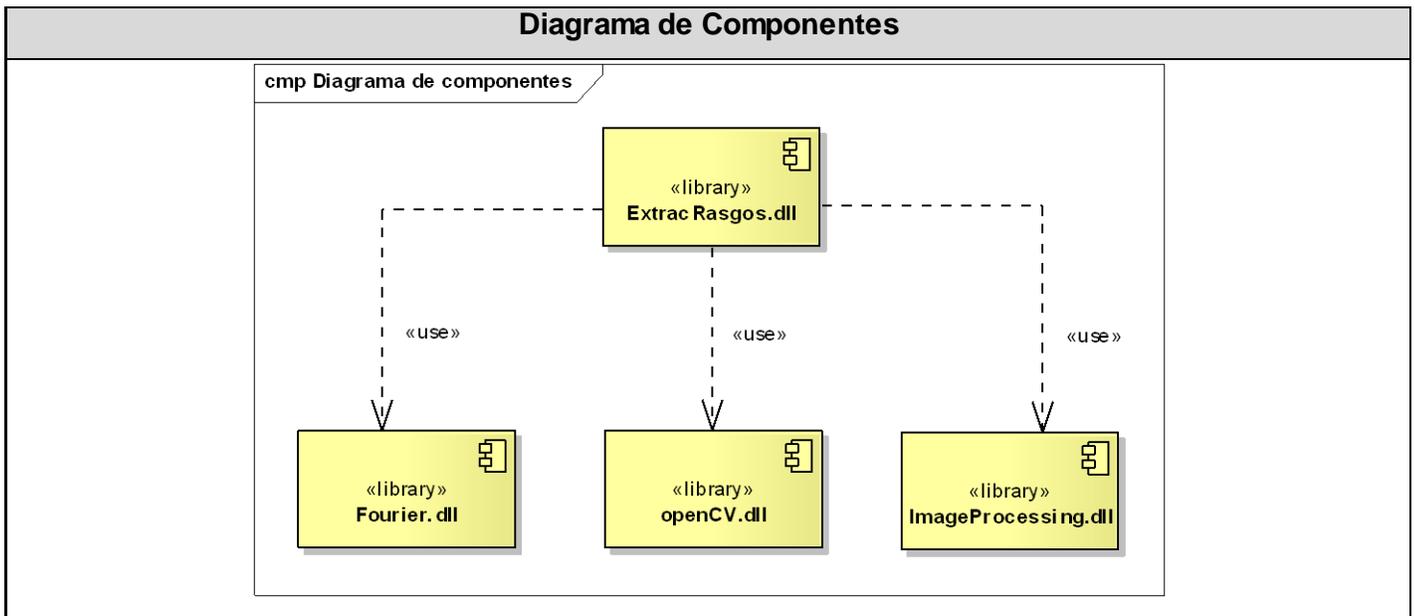


Fig. 4.18 Diagrama de componentes.

## **Conclusiones**

En este capítulo se han planteado los principales pasos realizados para lograr el objetivo de implementar un componente que permita la extracción de rasgos característicos en culotes de proyectiles. Además se expusieron los descriptores que se han obtenido y el diagrama de componentes.

# Conclusiones Generales

---

Durante la investigación, se implementó un componente que permite la extracción de los rasgos característicos en imágenes digitales de culotes de proyectiles, se centró el procesamiento en los proyectiles donde las vainas son de fuego central. Se precisan automáticamente las regiones de interés y se recolectan las características de la marca de la aguja percutora y el fulminante por separado, para poder lograr una mayor efectividad en un posterior correlacionamiento entre características de imágenes balísticas, proceso este último que es parte esencial de un sistema ABIS.

Se analizaron los sistemas existentes a nivel mundial que procesan las imágenes balísticas, demostrándose que por sus altas prestaciones poseen un alto precio en el mercado.

Se estudiaron algunos métodos y algoritmos sobre la comparación y procesamiento de imágenes balísticas e imágenes digitales en general. Con el empleo de técnicas como la binarización, la determinación de las regiones conexas, el Snake, la vista polar se logró darle solución al problema planteado. El algoritmo propuesto para implementar el componente se detalló en el capítulo 4 y se ha comprobado su efectividad a través de diferentes pruebas con imágenes de culotes de proyectiles.

Para la implementación del sistema, debido a sus particularidades, se definió una arquitectura basada en el patrón *Pipes & Filters* que permite descomponer el problema en pasos independientes, reutilizar filtros, y facilita el mantenimiento e independencia.

El componente permite la extracción de 11 y 14 descriptores del fulminante y de la marca de la aguja percutora respectivamente. Algunos de estos que en pruebas de correlacionamiento muestren un alto índice de acierto, pueden ser utilizados para el cotejo de imágenes balísticas digitales.

# Recomendaciones

---

Los autores del presente trabajo de diploma recomiendan:

- Continuar el estudio de métodos y algoritmos para caracterizar imágenes balísticas.
- Continuar la optimización del componente para lograr un mejor tiempo de procesamiento y extracción de las características.
- Utilizar el componente propuesto como base para un Sistema Automático de Identificación Balística.
- Estudiar otros tipos de marcas que sirvan para el cotejo balístico e implementar otras funcionalidades para su procesado y caracterización.
- Estudiar las vainas de tipo fuego anular (rimfire) para ampliar el procesamiento a este tipo de vaina.

# Referencias Bibliográficas

---

1. **Valdez, Dr. Hugo Nazario.** *Orange.utb.edu*. [En línea] [Citado el: 2 de Noviembre de 2008.] [orange.utb.edu/hvaldez1/PPP/BALISTICA%20FORENSE.ppt](http://orange.utb.edu/hvaldez1/PPP/BALISTICA%20FORENSE.ppt).
2. **www.lasarmas.com.** *Criminalistic.org*. [En línea] 26 de enero de 2007. [Citado el: 2009 de enero de 28.] [http://criminalistic.org/index2.php?option=com\\_content&do\\_pdf=1&id=203](http://criminalistic.org/index2.php?option=com_content&do_pdf=1&id=203).
3. **Ruiz, Mauricio.** *Criminalística Forense. hpchile.cl*. [En línea] 22 de febrero de 2006. [Citado el: 17 de abril de 2009.] <http://www.hpchile.cl>.
4. **Latino Seguridad.** *Latino Seguridad*. [En línea] 2008. [Citado el: 2 de Noviembre de 2008.] <http://www.latinoseguridad.com/LatinoSeguridad/Reps/BalFor.shtml>.
5. **Tulleners, Frederic A.** *Technical evaluation: Feasibility of a Ballistics imaging databases for all new handgun sales*. Sacramento, CA, E.U : s.n., 2001.
6. **Kopel, David B. y Burnett, H. Sterling.** *Ballistic Imaging: Not Ready for Prime Time*. 2003.
7. **Technology, Forensic.** *Forensic Technology*. [Online] 2008. [Cited: noviembre 5, 2008.] <http://www.fti-ibis.com>.
8. **eNotes.com Inc.** *enotes.com*. [En línea] 2009. [Citado el: 9 de abril de 2009.] <http://www.enotes.com/forensic-science/drugfire>.
9. **Florida Department of Law Enforcement.** *Florida Department of Law Enforcement*. [Online] 2008. [Cited: noviembre 8, 2008.] [http://www.fdle.state.fl.us/Publications/tech\\_success\\_stories/drugfire.htm](http://www.fdle.state.fl.us/Publications/tech_success_stories/drugfire.htm).
10. **Cardini, Dr. Fernando.** *TECNICAS DE INVESTIGACION CRIMINAL*. [En línea] 12 de mayo de 2003. [Citado el: 8 de noviembre de 2008.] [www.tecnicasinvcriminal.com.ar](http://www.tecnicasinvcriminal.com.ar).
11. **Papillon Systems.** *Papillon Systems*. [En línea] 2009. [Citado el: 10 de abril de 2009.] <http://www.papillon.ru/eng/7/?PHPSESSID=e4b224b8f29a1d513f9e1439179da848>.
12. **CENATAV.** *CENATAV*. [En línea] 2004. [Citado el: 10 de abril de 2009.] <http://www.cenatav.co.cu>.

13. **Clikear**. *Clikear.com*. [En línea] 2 de abril de 2007. [Citado el: 10 de abril de 2009.] <http://www.clikear.com/manuales/csharp/c10.asp>.
14. **Object Management Group, Inc.** *Object Management Group*. [En línea] 2009. [Citado el: 17 de abril de 2009.] <http://www.omg.org>.
15. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady**. *El Lenguaje Unificado de Modelado, Manual de referencia*. s.l. : Addison Wesley, 1998.
16. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James**. *El Proceso Unificado de Desarrollo de Software*. Madrid : PEARSON EDUCACION, S.A, 2000.
17. **Sparx Systems Pty Ltd.** *SPARX Systems*. [En línea] 2007. [Citado el: 9 de abril de 2009.] <http://www.sparxsystems.com.ar>.
18. **Microsoft**. *MSDN*. [En línea] 2009. [Citado el: 9 de abril de 2009.] <http://msdn.microsoft.com/es-es/vstudio/products/default.aspx>.
19. **AutomatedQA**. *AutomatedQA- Tools for Software Testing and Quality Assurance. AutomatedQA*. [En línea] 2009. [Citado el: 20 de abril de 2009.] <http://www.automatedqa.com/products/aqtime/index.asp>.
20. **HispaNetwork Publicidad y Servicios, S.L.** *Glosario.net*. [En línea] HispaNetwork Publicidad y Servicios, S.L., 2003-2009. [Citado el: 21 de enero de 2009.] <http://www.glosario.net>.
21. **Gracia, Joaquin**. *IngenieroSoftware*. [En línea] 2003. [Citado el: 10 de abril de 2009.] <http://www.ingenierosoftware.com>.
22. **Committee to Assess the Feasibility, Accuracy, and Technical Capability of a National Ballistics Database, National Research Council**. *Ballistic Imaging*. [ed.] Daniel L. Cork, John E. Rolph y Eugene S. Meieran. Washington : THE NATIONAL ACADEMIES PRESS, 2008. pág. 345. 0-309-11725-9.
23. **Paz Gutiérrez, Jenaro Carlos**. *Generación de Imágenes para Web con GD+*. Juárez : s.n., 2007. pág. 440. 978-968-7845-913.

24. **Geradts, Zeno, Bijhold, Jurrien y Hermsen, Rob.** *Pattern recognition in a database of cartridge cases*. Department Forensic Information, Netherlands Forensic Science Laboratory of the Ministry of Justice.
25. **T, Harsha, S, Amarnath y Mahesh Reddy, S.** CodeProject: Polar View of an Image. *CodeProject*. [En línea] 24 de Febrero de 2009. [Citado el: 15 de abril de 2009.] <http://www.codeproject.com/KB/graphics/PolarView.aspx>.
26. **Presutti, Miriam.** *La matriz de co-ocurrencia en la clasificación multiespectral: Tutorial para la enseñanza de medidas texturales en cursos de grado universitario*. Departamento de Ambiente y Recursos Naturales, Universidad Nacional de La Plata, Facultad de Ciencias Agrarias y Forestales. 2004. pág. 9.
27. **Universidad de Málaga.** *Biostatística, Dpto. Medicina Preventiva y Salud Pública*. [En línea] [Citado el: 16 de abril de 2009.] <http://www.bioestadistica.uma.es/>.
28. **Jiménez Jaramillo, Alvaro.** *Análisis del comportamiento de descriptores de la matriz de coocurrencia en la extracción de características de textura en imágenes digitales*. Universidad Nacional de Colombia Sede Manizales, Facultad de Ingeniería y Arquitectura. Manizales : s.n., 2002. pág. 46, Trabajo Final para optar al título de Ingeniero Electrónico.
29. **Universidad de Coruña.** Departamento de electrónica y sistemas. [En línea] 2009. [Citado el: 12 de abril de 2009.] [http://www.des.udc.es/~adriana/TercerCiclo/Cursolmagen/curso/web/Filtrado\\_Espacial.html](http://www.des.udc.es/~adriana/TercerCiclo/Cursolmagen/curso/web/Filtrado_Espacial.html).

# Bibliografía

---

1. **Apple Inc.** Apple .Mac. mac.com. [En línea] 2008. [Citado el: 4 de Febrero de 2009.] <http://homepage.mac.com/imaz/iblog/C612772037/E20050907222635/Media/Algunos%20Tipos%20de%20Arquitecturas.pdf>.
2. **AutomatedQA.** AutomatedQA- Tools for Software Testing and Quality Assurance. AutomatedQA. [En línea] 2009. [Citado el: 20 de abril de 2009.] <http://www.automatedqa.com/products/aqtime/index.asp>.
3. **Cardini, Dr. Fernando.** TECNICAS DE INVESTIGACION CRIMINAL. [En línea] 12 de mayo de 2003. [Citado el: 8 de noviembre de 2008.] [www.tecnicasinvcriminal.com.ar](http://www.tecnicasinvcriminal.com.ar).
4. **CENATAV.** CENATAV. [En línea] 2004. [Citado el: 10 de abril de 2009.] <http://www.cenatav.co.cu>.
5. **Clikear.** Clikear.com. [En línea] 2 de abril de 2007. [Citado el: 10 de abril de 2009.] <http://www.clikear.com/manuales/csharp/c10.asp>.
6. **Committee on Identifying the Needs of the Forensic Science Community; Committee on Science, Technology, and Law Policy and Global Affairs; Committee on Applied and Theoretical Statistics, Division on Engineering and Physical Sciences.** Strengthening Forensic Science in the United States: A Path Forward. [ed.] SARA D. MADDUX y Donald Kennedy. Washington, D.C. : National Academies Press, 2009. pág. 254. 0-309-13131-6.
7. **Committee to Assess the Feasibility, Accuracy, and Technical Capability of a National Ballistics Database, National Research Council.** Ballistic Imaging. [ed.] Daniel L. Cork, John E. Rolph y Eugene S. Meieran. Washington : THE NATIONAL ACADEMIES PRESS, 2008. pág. 345. 0-309-11725-9.
8. **Daniele, Ing. Marcela.** Departamento de Computación. Universidad Nacional de Río Cuarto. [En línea] 2 de Noviembre de 2006. [Citado el: 4 de Marzo de 2009.] <http://dc.exa.unrc.edu.ar/nuevodc/materias/ingenieria/Material/1163109537/1163109558/PipesyFilters%202.pdf>.

9. **eNotes.com Inc.** enotes.com. [En línea] 2009. [Citado el: 9 de abril de 2009.]  
<http://www.enotes.com/forensic-science/drugfire>.
10. **F. Vecchia, Dominic, y otros.** Statical Engineering Division. NIST: National Institute of Standards and Technology. [En línea] 20 de julio de 2001. [Citado el: 7 de febrero de 2009.]  
<http://www.itl.nist.gov/div898/pubs/ar/ar1999/node31.html>.
11. **Fiscalía General de la República de Colombia.** Balística Forense. 2005. ISBN 958-97156-9-9.
12. **Florida Departament of Law Enforcement.** Florida Departament of Law Enforcement. [En línea] 2008. [Citado el: 8 de noviembre de 2008.]  
[http://www.fdle.state.fl.us/Publications/tech\\_success\\_stories/drugfire.htm](http://www.fdle.state.fl.us/Publications/tech_success_stories/drugfire.htm).
13. **Geradts, Zeno, Bijhold, Jurrien y Hermsen, Rob.** Pattern recognition in a database of cartridge cases. Department Forensic Information, Netherlands Forensic Science Laboratory of the Ministry of Justice.
14. **Geradts, Zeno, y otros.** Image matching algorithms for breech face marks and firing pins in a database of spent cartridge cases of firearms. Netherlands Forensic Institute of the Ministry of Justice, Queen's University of Belfast. pág. 8.
15. **Gonzáles García, Elizabeth, y otros.** Descriptores de Fourier para identificación y posicionamiento de objetos en entornos 3D. Ciudad Real : s.n., 2004. Evento: Jornadas de Automática.
16. **Gracia, Joaquin.** IngenieroSoftware. [En línea] 2003. [Citado el: 10 de abril de 2009.]  
<http://www.ingenierosoftware.com>.
17. **HispaNetwork Publicidad y Servicios, S.L.** Glosario.net. [En línea] HispaNetwork Publicidad y Servicios, S.L., 2003-2009. [Citado el: 21 de enero de 2009.] <http://www.glosario.net>.
18. **Jacobson, Ivar, Booch, Grady y Rumbaugh, James.** El Proceso Unificado de Desarrollo de Software. Madrid : PEARSON EDUCACION, S.A, 2000.
19. **Jiménez Jaramillo, Alvaro.** Análisis del comportamiento de descriptores de la matriz de coocurrencia en la extracción de características de textura en imágenes digitales. Universidad Nacional de Colombia

Sede Manizales, Facultad de Ingeniería y Arquitectura. Manizales : s.n., 2002. pág. 46, Trabajo Final para optar al título de Ingeniero Electrónico.

20. **Kopel, David B. y Burnett, H. Sterling.** Ballistic Imaging: Not Ready for Prime Time. 2003.

21. **Latino Seguridad.** Latino Seguridad. [En línea] 2008. [Citado el: 2 de Noviembre de 2008.] <http://www.latinoseguridad.com/LatinoSeguridad/Reps/BalFor.shtml>.

22. **Macias Norat, Mayelin.** Reconocimiento Automático de armas. Universidad de Oriente. Santiago de Cuba : s.n. pág. 14.

23. **Mery, Domingo.** Extracción de Características. Departamento de Ciencia de la Computación, Universidad Católica de Chile. Santiago de Chile : s.n., 2006. pág. 22.

24. **Microsoft.** MSDN. [En línea] 2009. [Citado el: 9 de abril de 2009.] <http://msdn.microsoft.com/es-es/vstudio/products/default.aspx>.

25. **National Academy of Sciences.** The National Academy Press. [En línea] 2009. [Citado el: 9 de abril de 2009.] <http://www.nap.edu/>.

26. **Object Management Group, Inc.** Object Management Group. [En línea] 2009. [Citado el: 17 de abril de 2009.] <http://www.omg.org>.

27. **Papillon Systems.** Papillon Systems. [En línea] 2009. [Citado el: 10 de abril de 2009.] <http://www.papillon.ru/eng/7/?PHPSESSID=e4b224b8f29a1d513f9e1439179da848>.

28. **Paz Gutiérrez, Jenaro Carlos.** Generación de Imágenes para Web con GDI+. Juárez : s.n., 2007. pág. 440. 978-968-7845-913.

29. **Porta, Paulo.** quesabesde.com. quesabesde.com. [En línea] Exartia Media SL, 28 de febrero de 2005. [Citado el: 16 de abril de 2009.] <http://www.quesabesde.com>.

30. **Presutti, Miriam.** La matriz de co-ocurrencia en la clasificación multiespectral: Tutorial para la enseñanza de medidas texturales en cursos de grado universitario. Departamento de Ambiente y

Recursos Naturales, Universidad Nacional de La Plata, Facultad de Ciencias Agrarias y Forestales. 2004. pág. 9.

31. **Rodríguez, José Luis.** dZoom: El Histograma, el Mejor Amigo del Fotógrafo. dZoom. [En línea] 14 de noviembre de 2007. [Citado el: 16 de abril de 2009.] <http://www.dzoom.org.es/noticia-1708.html>.

32. **Ruiz, Mauricio.** Criminalística Forense. hpchile.cl. [En línea] 22 de febrero de 2006. [Citado el: 17 de abril de 2009.] <http://www.hpchile.cl>.

33. **Rumbaugh, James, Jacobson, Ivar y Booch, Grady.** El Lenguaje Unificado de Modelado, Manual de referencia. s.l. : Addison Wesley, 1998.

34. **S. Nixon, Mark y S. Aguado, Alberto.** Feature Extraction and Image Processing. Primera edición. s.l. : Planta Tree, 2002. 0 7506 5078 8.

35. **Scott Doyle, Jeffrey.** FirearmsID. [En línea] National Academy of Sciences, 2009. [Citado el: 9 de Abril de 2009.] <http://www.firearmsid.com/>.

36. **Sparx Systems Pty Ltd.** SPARX Systems. [En línea] 2007. [Citado el: 9 de abril de 2009.] <http://www.sparxsystems.com.ar>.

37. **T.V, Vorburger, y otros.** Surface Topography Analysis for a Feasibility Assessment of a National Ballistics Imaging Database. National Institute of Standards and Technology; Intelligent Automation, Inc. 2007. pág. 173.

38. **T, Harsha, S, Amarnath y Mahesh Reddy, S.** CodeProject: Polar View of an Image. CodeProject. [En línea] 24 de Febrero de 2009. [Citado el: 15 de abril de 2009.] <http://www.codeproject.com/KB/graphics/PolarView.aspx>.

39. **Technology, Forensic.** Forensic Technology. [En línea] 2008. [Citado el: 5 de noviembre de 2008.] <http://www.fti-ibis.com>.

40. **Tulleners, Frederic A.** Technical evaluation: Feasibility of a Ballistics imaging databases for all new handgun sales. Sacramento, CA, E.U : s.n., 2001.

41. **Universidad de Coruña.** Departamento de electrónica y sistemas. [En línea] 2009. [Citado el: 12 de abril de 2009.]  
[http://www.des.udc.es/~adriana/TercerCiclo/Cursolmagen/curso/web/Filtrado\\_Espacial.html](http://www.des.udc.es/~adriana/TercerCiclo/Cursolmagen/curso/web/Filtrado_Espacial.html).
42. **Universidad de Málaga.** Biostatística, Dpto. Medicina Preventiva y Salud Pública. [En línea] [Citado el: 16 de abril de 2009.] <http://www.bioestadistica.uma.es/>.
43. **University of California.** Firearms Microstamping Feasible But Variable, Study Finds. ScienceDaily. [En línea] 16 de mayo de 2008. [Citado el: 23 de enero de 2009.]  
<http://www.sciencedaily.com/releases/2008/05/080514092333.htm>.
44. **Valdez, Dr. Hugo Nazario.** Orange.utb.edu. [En línea] [Citado el: 2 de Noviembre de 2008.]  
[orange.utb.edu/hvaldez1/PPP/BALISTICA%20FORENSE.ppt](http://orange.utb.edu/hvaldez1/PPP/BALISTICA%20FORENSE.ppt).
45. **www.lasarmas.com.** Criminalistic.org. [En línea] 26 de enero de 2007. [Citado el: 2009 de enero de 28.] [http://criminalistic.org/index2.php?option=com\\_content&do\\_pdf=1&id=203](http://criminalistic.org/index2.php?option=com_content&do_pdf=1&id=203).
46. **Yi Tou, Jing, Haur Tay, Yong y Yee Lau, Phooi.** One-dimensional Grey-level Co-occurrence Matrices for Texture Classification. Universiti Tunku Abdul Rahman (UTAR), Malaysia, Instituto de Telecomunicacoes, Portugal. IEEE. 2008. pág. 6. 978-1-4244-2328-6.
47. **Zárate Rea, Héctor.** Aplicación de las Matrices en el Procesamiento de Imágenes. División de ciencias Básicas, Universidad Nacional Autónoma de México, Facultad de ingeniería. pág. 15, Trabajo Final de Asignatura Algebra Lineal.

# Anexos

---

## Anexo 1. Descripción extendida de los casos de uso

<b>CU1</b>	Obtener Regiones de Interés	
<b>Propósito</b>	Localizar en la imagen donde es que se encuentran la marcas que se quieren procesar.	
<b>Actores</b>	Sistema ABIS (Inicia)	
<b>Resumen:</b>	El Sistema ABIS solicita la obtención de las regiones de interés, obteniéndose las coordenadas donde están las regiones.	
<b>Referencias</b>	RF 1	
<b>Flujo normal de los eventos</b>		
	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	1- Solicita obtención de las regiones de interés de una imagen.	2- Chequea que la imagen no está corrupta.
		3- Define en que región aproximada se encuentra el fulminante que es quien contiene las marcas.
		4- Convierte la imagen a escala de grises.
		5- Elimina ruido de la imagen.
		6- Binariza la imagen.
		7- Localiza la región en donde está el fulminante.
		8- Invierte la imagen.
		9- Localiza la región en donde está la marca de la aguja percutora.
		10- Devuelve las dos regiones.
	11- Obtiene las regiones de interés.	
<b>Flujo alternativo Paso 2</b>		
	<b>Acción del actor</b>	<b>Respuesta del sistema</b>

	3- En caso de que la imagen no tenga el formato o tamaño requerido, lanza una excepción que informa el error.
4- Observa mensaje de error.	

<b>CU2</b>	Obtener Rasgos Característicos.	
<b>Propósito</b>	Obtener los descriptores de la marca de la aguja percutora y del fulminante.	
<b>Actores</b>	Sistema ABIS (Inicia)	
<b>Resumen:</b>	El Sistema ABIS solicita la obtención de los rasgos característicos de las marcas en estudio, el componente procesa la imagen para obtener a cual o cuales se le obtendrán los descriptores, se extraen los rasgos de la marca de la aguja percutora y de la marca del espaldón, retornando el vector de características de la imagen.	
<b>Referencias</b>	RF2	
<b>Flujo normal de los eventos</b>		
	<b>Acción del actor</b>	<b>Respuesta del sistema</b>
	1- Solicita la obtención de los rasgos característicos.	2- Separa en imágenes apartes las dos marcas.
		3- Rellena con color negro las regiones de las imágenes que no son de interés.
		4- Cambia a coordenadas polares las imágenes.
		5- Localiza las regiones negras.
		6- Elimina las regiones que no son de interés.
		7- Procesa las imágenes para obtener descriptores.
		8- Devuelve el vector característico.
	9- Obtiene vector característico.	

## Anexo 2. Imágenes balísticas.

### Fuego Central (Centerfire)







**Fuego Anular (Rimfire)**



# Glosario de términos

---

Término	Definición
Aguja percutora	Pieza de cualquier arma que martilla por acción del mecanismo del disparo, el fulminante, produciendo la inflamación de los gases la ejecución del disparo.
Ánima	En las piezas de artillería y en toda arma de fuego, en general, el hueco del cañón.
Binarizar	La binarización de una imagen digital consiste en convertir la imagen digital en una imagen en blanco y negro, de tal manera que se preserven las propiedades esenciales de la imagen.
Calibre	Diámetro interior de las armas de fuego, diámetro del proyectil.
Cartucho o vaina de fuego anular.	Tipo de cartucho que poseen los proyectiles donde la carga explosiva que al contactar con la aguja percutora produce el disparo del plomo se encuentra en el anillo exterior del culote del cartucho.
Cartucho o vaina de fuego central	Tipo de cartucho que poseen los proyectiles donde la carga explosiva que al contactar con la aguja percutora produce el disparo del plomo se encuentra en el centro del culote del cartucho.
Concéntrico	Objeto o figura que tiene el mismo centro que otro objeto o figura.
Cotejo	Comparación de una cosa con otra u otras.
Culote	Parte posterior de las vainillas de los proyectiles.
Dubitado	Documento o cualquier otro indicio, que es susceptible de ser estudiado e investigado, es el dudoso.
Encapsulamiento	Ocultamiento del estado, es decir, de los datos miembros de un objeto de manera que sólo se puede cambiar mediante las operaciones definidas para ese objeto.

Espaldón	Valla artificial, de altura y cuerpo correspondientes, para resistir y detener el impulso de un tiro o rechazo.
Estría	Raya, surco o hendidura que suelen tener algunos cuerpos.
Indubitado	Es el documento u otro indicio que se sabe cierta su procedencia.
Matriz de coocurrencia	Se utiliza para caracterizar la información contenida en la textura de una imagen, o en general la ocurrencia de cierto evento de una variable aleatoria
Vaina	Cilindro de metal donde se encuentra la pólvora, el fulminante y es el soporte para el plomo del proyectil balístico.