

Universidad de las Ciencias Informáticas

Facultad 6



**Título: Software para el Procesamiento de
Imágenes de Microarray.**

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

Autor: Yenisbel Valle Garrido.

Tutor: Ing. Sergio Barrios Díaz.

Consultante: Dr. Julio Raúl Fernández Massó.

Lic. Miguel Sancho Fernández.

Julio, 2007

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los 4 días del mes de julio del año 2007.

Yenisbel Valle Garrido

Sergio Barrios Díaz

Firma del Autor

Firma del Tutor

DATOS DE CONTACTO

Tutor: Sergio Barrios Díaz, Pinar del Río (1979). Ingeniero en Ciencias Informáticas de la Universidad de las Ciencias informáticas en el 2006. Actualmente profesor de la facultad 6 en la que imparte clases de 2 asignaturas, coordinador de 4to año de la facultad, Líder del Proyecto MicroImage, fundador de la UCI. Email: sbarrios@uci.cu

Consultante: Julio Raúl Fernández Massó, Ciudad de la Habana. Ingeniero químico tecnólogo. Máster en Ciencias, Instituto Tecnológico de Bielorrusia en el 1986. Actualmente ocupa el cargo de Investigador en el Centro de Ingeniería Genética y Biotecnología. Miembro del Comité Experto Programa Neurociencias del CITMA desde el 2005. Ha impartido varios cursos de postgrados de Biología y Neurobiología Molecular, cursos de Biotecnología Universidad para Todos entre otras. Cuenta con diversas publicaciones reconocidas nacional e internacionalmente, así como resultados académicos de la Academia de Ciencias de Cuba. Email: julio.fernandez@cigb.edu.cu

Consultante: Miguel Sancho Fernández, Ciudad de la Habana (1980). Licenciado en Ciencia de la Computación, en la Facultad de Matemática y Computación de la Universidad de la Habana en el 2003. Actualmente cursa la Maestría en Ciencia de la Computación. Profesor Instructor Adjunto de la UCI, 3 años de trabajo como profesor en la educación superior. Se desempeña laboralmente como Cuadro Profesional de la UJC en la UCI en la responsabilidad de funcionario que atiende la producción, fundador de la UCI. Email: sancho@uci.cu

La sabiduría resulta de la síntesis armoniosa entre el conocimiento, fundamental en la razón, la dimensión poética de la existencia y la locura sabia del amor.

Edgar Morin

AGRADECIMIENTOS

A mis padres, mi familia y mis amigos, los de aquí y los de allá, gracias por compartir mis penas y duplicar mis alegrías.

A mis compañeros de aula y de laboratorio, los lazos de amistad hacen los nudos más fuertes; y a mis confidentes de estos cinco años, Aida y Keily, por enseñarme que mientras no se ha dado todo no se ha dado nada.

A Migue: por el apoyo sin límites, la dedicación y el amor profesado; sobretodo por la constante superación que ha implicado su persona.

Aquellos que de una forma u otra colaboraron con la realización de este trabajo: al Dr. Julio Raúl Fernández por el asesoramiento científico, a Henrik por su presencia certera y ayuda desinteresada, y a los profesores Ana Lupe, Aliesky, Liliana y Sergio cuyos conocimientos sirvieron de guía en esta larga travesía.

Agradecer también a los que me cerraron las puertas y se encargaron que no todos mis amaneceres fueran hermosos; por ustedes conocí mi voluntad de trabajo y fuerza para marcar el camino aunque se levante el polvo.

DEDICATORIA

A los que me enseñaron que uno crece asimilando lo que deja por detrás, construyendo lo que tiene por delante y proyectando lo que puede ser el porvenir; mis padres, no existe comparación alguna para lo mucho que representan en mi vida.

Para mi hermana, por el enorme compromiso que contraigo con su persona y la confianza depositada que espero no defraudar.

A mi familia, porque el amor más grande del mundo ahí lo he encontrado, por componer el armazón sobre el que se teje mi existencia.

RESUMEN

La investigación desarrollada centra su atención en mejorar la eficiencia del procesamiento y cuantificación de imágenes de *microarray* en el Centro de Ingeniería Genética y Biotecnología. Esta necesidad da paso al análisis y diseño de una aplicación de software para el procesamiento de imágenes a partir de la técnica de *microarray*, concebida particularmente para la obtención de valores de intensidades que soportan determinado nivel investigativo; objetivo fundamental desarrollado en este trabajo.

Se realiza un análisis crítico de la metodología y tecnología a usar, se fundamentan los conocimientos biológicos necesarios, y a partir de un modelo conceptual se diseña una propuesta que contiene los métodos para procesar las imágenes de *microarray*, visualizar los resultados e información numérica necesaria, logrando que sea flexible a cambios que puedan suscitarse en el futuro.

PALABRAS CLAVES: imágenes de *microarray*, procesamiento, cuantificación.

TABLA DE CONTENIDOS

INTRODUCCIÓN	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Introducción	5
1.2 Características y evolución del <i>microarray</i>	5
1.2.1 ¿Qué es un <i>microarray</i> ?.....	5
1.2.2 Construcción y funcionamiento de un <i>microarray</i>	6
1.2.3 ¿Qué tipo de <i>microarrays</i> existen?	7
1.2.4 Aplicaciones de los <i>microarrays</i>	9
1.3 Imágenes digitales de un <i>microarray</i>	9
1.4 Tratamiento de imágenes. Procesamiento de imágenes de <i>microarray</i>	11
1.4.1 Preprocesamiento	11
1.4.2 Métodos de procesamiento	11
1.5 Programas vinculados al procesamiento y cuantificación de imágenes.....	17
1.6 Importancia del procesamiento y cuantificación de imágenes de <i>microarray</i>	18
1.7 Metodologías de desarrollo	18
1.7.1 Herramientas CASE	22
1.7.2 Artefactos generados	23
1.8 Metodología de investigación	24
1.8.1 Método utilizado para recopilar información.....	24
1.8.2 Método utilizado para seleccionar y organizar los datos.....	25
1.9 Tendencias y tecnologías actuales	25
1.9.1 Herramientas de desarrollo y <i>framework</i>	26
1.9.2 Lenguajes de programación	28
1.10 Conclusiones	29
CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA.....	30
2.1 Introducción	30
2.2 Objeto de Estudio	31
2.2.1 Flujo actual de procesos	31
2.2.2 Análisis crítico de la ejecución de procesos	32

2.3	Objeto de automatización	33
2.4	Propuesta de sistema	34
2.5	Modelo de Dominio.....	38
2.5.1	Descripción de clases	39
2.6	Sistema	40
2.6.1	Requisitos Funcionales.....	41
2.6.2	Requisitos no Funcionales.....	42
2.6.3	Actores identificados. Diagrama de Casos de Uso	44
2.6.4	Descripción textual de Casos de Uso.....	45
2.7	Conclusiones.....	56
CAPÍTULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA		58
3.1	Introducción.....	58
3.2	Clases del análisis.....	59
3.3	Realización de Casos de Uso del análisis	59
3.4	Diseño	60
3.4.1	Estilo arquitectónico.....	60
3.4.2	Subsistemas e interfaces.....	62
3.4.3	Clases del diseño	65
3.4.4	Realizaciones de Casos de Uso del diseño	65
3.4.5	Patrones de diseño: Singleton.....	66
3.5	Conclusiones.....	67
CONCLUSIONES		68
RECOMENDACIONES.....		69
REFERENCIAS BIBLIOGRÁFICAS.....		70
ANEXOS		72
GLOSARIO DE TÉRMINOS.....		88

ÍNDICE DE FIGURAS

FIGURA 1 MUESTRA DE IMAGEN DE <i>MICROARRAY</i> EN UN SOPORTE DE VIDRIO.-----	6
FIGURA 2 REPRESENTACIÓN DE EXPERIMENTO BÁSICO DE <i>MICROARRAY</i> .-----	7
FIGURA 3 LA REGIÓN QUE ESTÁ DENTRO DEL CÍRCULO ROJO REPRESENTA LA MÁSCARA DEL <i>SPOT</i> (<i>SPOT MASK</i>) Y LA REGIÓN VERDE SE UTILIZAN PARA CALCULAR EL FONDO LOCAL (<i>BACKGROUND LOCAL</i>), POR DIFERENTE SOFTWARE. VERDE: USADO EN <i>QUANTARRAY</i> ; AZUL: USADO EN <i>SCANANALYZE</i> Y EL ROSADO SE USA EN <i>ARRAYVISION</i> .-----	16
FIGURA 4 REPRESENTACIÓN DEL FLUJO DE TRABAJO DE MODELAMIENTO DE NEGOCIO Y REQUERIMIENTOS DURANTE EL CICLO DE RUP.-----	30
FIGURA 5 LA FIGURA REPRESENTA LOS CINCO COMPONENTES FUNDAMENTALES PARA EL DISEÑO, HIBRIDACIÓN, CUANTIFICACIÓN Y ANÁLISIS DE LOS <i>MICROARRAY</i> .-----	32
FIGURA 6 IMAGEN DE <i>MICROARRAY</i> OBTENIDA EN EL CIGB.-----	35
FIGURA 7 REGIÓN SELECCIONADA DE LA IMAGEN ORIGINAL.-----	36
FIGURA 8 REPRESENTACIÓN DE LA SECUENCIA A SEGUIR PARA ANALIZAR EL HISTOGRAMA CORRESPONDIENTE FIGURA 6.-----	37
FIGURA 9 IMAGEN GRILLADA, RESULTADO DEL PRIMER MÉTODO DE PROCESAMIENTO.-----	37
FIGURA 10 NÓTESE QUE SE ELIMINA RUIDO EN ALGUNO DE LOS <i>SPOTS</i> DE LA IMAGEN DE LA DERECHA. AQUELLOS QUE QUEDAN SIN RELLENAR SON DEMASIADO TENUES Y SERÁN DESCARTADOS POSTERIORMENTE POR EL SOFTWARE IDENTIFICÁNDOLOS COMO MOLÉCULAS AUSENTES.-----	38
FIGURA 11 DIAGRAMA QUE REPRESENTA LOS CONCEPTOS O CLASES FUNDAMENTALES IDENTIFICADAS EN EL DOMINIO DEL PROBLEMA.-----	39
FIGURA 12 DIAGRAMA DE CASOS DE USO DEL SISTEMA.-----	45
FIGURA 13 REPRESENTACIÓN DEL FLUJO DE TRABAJO DE ANÁLISIS Y DISEÑO DURANTE EL CICLO DE RUP.-----	58
FIGURA 14 ARQUITECTURA EN 3 CAPAS PARA EL PROCESAMIENTO DE IMÁGENES DE <i>MICROARRAY</i> .-----	62
FIGURA 15 REPRESENTACIÓN DE SUBSISTEMAS E INTERFACES.-----	63
FIGURA 16 REPRESENTACIÓN DEL PATRÓN « <i>SINGLETON</i> » EN LA CLASE IMAGEN.-----	66

ÍNDICE DE TABLAS

TABLA 1 DIFERENTES SOFTWARE PARA EL PROCESAMIENTO Y CUANTIFICACIÓN DE IMÁGENES EN EL MUNDO.---	17
TABLA 2 BREVE DESCRIPCIÓN DEL ACTOR DEL SISTEMA IDENTIFICADO. -----	44
TABLA 3 CU-CARGAR IMAGEN. -----	46
TABLA 4 CU-SELECCIONAR REGIÓN DE INTERÉS. -----	47
TABLA 5 CU-PROCESAR IMAGEN. -----	48
TABLA 6 CU-REALIZAR GRILLADO.-----	49
TABLA 7 CU-ANALIZAR HISTOGRAMA.-----	50
TABLA 8 CU-REALIZAR SEGMENTACIÓN. -----	51
TABLA 9 CU- SEGMENTAR POR UMBRALIZACIÓN.-----	52
TABLA 10 CU-EXTRAER INTENSIDAD. -----	54
TABLA 11 CU-MOSTRAR IMAGEN CUANTIFICADA. -----	55
TABLA 12 CU-TABULAR VALORES.-----	56

INTRODUCCIÓN

El desarrollo vertiginoso de la era post-genómica dependerá en gran medida de lo eficiente que sea la comunidad científica mundial en desarrollar y validar herramientas biológicas relacionadas con el estudio del genoma.

El genoma de los seres vivos es el conjunto de genes que se encuentran distribuidos en cromosomas. Los genes, a su vez, son secuencias de ADN que contienen toda la información necesaria para sintetizar las proteínas, moléculas esenciales para la vida que realizan prácticamente todas las funciones celulares.

Cuando un gen se «activa» para dar lugar a su proteína correspondiente, ese gen se expresa en esa célula. Es conocido que anomalías en la expresión de los genes pueden llevar a disfunciones celulares, provocando graves enfermedades como el cáncer, entre otras. Por lo tanto, la identificación de los genes desregulados es un paso importante para conocer las bases moleculares de muchas enfermedades de carácter genético.

Estudiar la relación gen-enfermedad no está basado en analizar un gen único y sus efectos, sino en analizar el comportamiento de miles de genes de forma simultánea. Estos sistemas, denominados genéricamente como matrices, *microarrays* o *biochips*, están cambiando la forma de plantear los problemas y extraer conclusiones de los experimentos que ofrecen una foto compleja del conjunto del genoma.

Para el avance científico en el siglo XXI, los *microarray* son una tecnología versátil y de amplias posibilidades, que han proporcionado velocidad al desarrollo de la biología molecular y la ciencia en general. La cuantificación de la expresión génica, a través de imágenes de *microarray*, se introduce como una técnica de alto rendimiento para trabajos en el área de la genómica funcional.

Con el propósito de desarrollar estudios para la detección de enfermedades con implicación genética en la población cubana, el Centro de Ingeniería Genética y Biotecnología (CIGB), dispone para analizar los resultados de dichos estudios del software *ArrayVision*, inicialmente utilizado en el desarrollo de proyectos de investigación.

Al incrementarse el número de estudios desarrollados y en consecuencia crecer de modo proporcional la cantidad de imágenes necesarias a procesar, el software *ArrayVision* no resultó eficiente ante los nuevos requerimientos, evidenciándose en:

- El análisis de la región seleccionada, resulta poco asequible para el investigador por estar descrito de modo complejo.
- Los resultados obtenidos deben ser organizados manualmente.

Referente a la alineación de *spots*, el software exhibe una cierta distorsión geométrica, acarreado errores en los valores de intensidad asociados a los *spots*.

La situación problemática planteada, requiere de una solución bioinformática, la cual está condicionada por el problema científico: ¿Cómo mejorar el procesamiento y cuantificación de imágenes de *microarray* en el CIGB?

A pesar de las dificultades que presenta el software *ArrayVision*, la familiarización con las funcionalidades y métodos matemáticos con que cuenta así como otras aplicaciones afines, han servido para la reutilización de prestaciones que se puedan adecuar a la propuesta de diseño del software, de sello nacional, que beneficie inicialmente los intereses de dicha institución, logrando en lo posible a la vez una genericidad, que permita satisfacer las necesidades de cualquier otro laboratorio del país, de modo que se facilite su posterior despliegue. Todo lo cual contribuirá al estudio de nuestra dotación genética para prevenir enfermedades de bases moleculares.

Con este fin, el diseño de la aplicación de software debe estar integrado funcionalmente por métodos, procesos y técnicas que permitan:

- Destacar características de interés de la imagen.
- Localizar *spots* con alta precisión (se tiene en cuenta cuántos *spots* hay, cómo están agrupados, etc.)
- Identificar que *píxeles* corresponden a un *spot* y cuales a su fondo (usar histograma, método de segmentación, según el umbral hallado).
- Cuantificar imágenes, extracción de intensidades asociadas a cada *spot* de la imagen.
- Obtener valores como la media, mediana y desviación estándar de las intensidades de los *píxeles* que forman el *spot*.

Gracias al desarrollo de nuevas técnicas y al perfeccionamiento de aquellas con que se contaban, ha sido posible el estudio de las interacciones entre múltiples proteínas de un organismo, lo que constituye una nueva visión global frente a la visión reduccionista de estudiar las proteínas de forma aislada.

Además del software *ArrayVision* (1), existen numerosos sistemas que posibilitan el estudio de imágenes de *microarray*, incluyendo el análisis, procesamiento y cuantificación de las mismas, ejemplo de estos son *SpotFinder* (2), *ScanAlyze* (3) y *QuantArray* (4). Sin embargo, ninguno de ellos se ajusta a las necesidades particulares del centro, con el inconveniente adicional de ser propietarios.

La nueva propuesta de software para el análisis y cuantificación de imágenes implica aportes importantes en el desarrollo de la medicina en el país; tales como, mejorar el proceso de identificación y detección de las irregularidades de los genes de forma rápida y confiable, crear un software propio de mayor eficiencia que *ArrayVision* en lo que respecta al procesamiento y cuantificación de imágenes actualmente desarrolladas en el CIGB, y por último, que se utilice como marco referencial para el estudio de la función de los genes, lográndose una mejor comprensión de diferencias genéticas individuales y su incidencia en el desarrollo de las enfermedades, específicamente en el territorio nacional.

El desarrollo de este trabajo se ha realizado en conjunto con la dirección de Investigaciones Biomédicas, específicamente con la División de Farmacéuticos del CIGB (5). Esta investigación comprende como objeto de estudio el procesamiento y análisis de imágenes de *microarray*, cuyo campo de acción está enfocado al procesamiento y cuantificación de la expresión genética en imágenes de *microarray*.

Desde la conceptualización de esta tarea con el CIGB se trazaron métricas que permitieran cumplir la misma eficientemente, identificándose como objetivo general, analizar y diseñar un software para el procesamiento y cuantificación de imágenes de *microarray*. En correspondencia, se han desglosado los objetivos específicos siguientes:

- Definir el alcance del Software para el Procesamiento de Imágenes de *Microarray*.
- Analizar un software que procese y cuantifique las imágenes de *microarray*.
- Diseñar un software que procese y cuantifique las imágenes de *microarray*.

Adhiriéndose a los objetivos mencionados, se trazaron una serie de tareas que dan cumplimiento a los mismos:

- Revisión bibliográfica sobre el procesamiento de imágenes de *microarray*.
- Entrevistas con investigadores del Centro de Ingeniería Genética y Biotecnología, específicamente de la División de Farmacéuticos.

- Estudio sobre las aplicaciones de software existentes para el procesamiento y cuantificación de las imágenes de *microarray*.
- Modelación del dominio.
- Especificación de procesos que serán automatizados, que permitan procesar y cuantificar imágenes de *microarray*.
- Construcción de los artefactos propios del análisis, que comprendan el procesamiento y cuantificación de imágenes.
- Elaboración de los artefactos esenciales de diseño, ajustados a la aplicación de software para el procesamiento y cuantificación de imágenes.

El presente trabajo cuenta de tres capítulos estructurados de la siguiente forma:

Capítulo1: Este capítulo hace referencia a los conceptos y temáticas más importantes a considerar en el procesamiento y análisis de imágenes de *microarray*, define las metodologías y tecnologías que guiarán todo el proceso, reseña las aplicaciones de software afines con la cuantificación y procesamiento de imágenes que han servido como basamento para la solución del problema.

Capítulo2: El presente capítulo resume las características más importantes del sistema, a partir de la descripción de objetos a automatizar en el procesamiento de imágenes y de un análisis crítico de la ejecución de dichos procesos. Incluye el modelo de dominio usado, la especificación de los casos de uso de sistema y los requisitos asociados a estos, se representan además los diagramas correspondientes a cada caso de uso descrito.

Capítulo3: En este capítulo se presentan y describen los principales artefactos del flujo de trabajo de análisis y diseño, según lo planteado por RUP y relacionado al procesamiento de imágenes. Se define la línea arquitectónica y los patrones que regirán el diseño propuesto.

CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

Es indiscutible que en las últimas décadas se ha producido una revolución tecnológica simultánea en el campo de la biomedicina y en el área de la informática. Estas dos ciencias han progresado a ritmo vertiginoso, destinadas a encontrarse y conformar una disciplina: la Bioinformática. Tal vez uno de los hitos más importantes que puede ilustrar la trascendencia del encuentro entre estas dos ciencias, es el *Proyecto del Genoma Humano* (6).

Una vez más la ciencia da un resultado antes de lo esperado, referente a la capacidad de secuenciar y analizar computacionalmente toda una avalancha de información que genera las investigaciones genéticas, contribuyendo de manera decisiva a conocer la identidad de nuestros genes en un tiempo.

1.2 Características y evolución del *microarray*.

La evolución histórica de la Biología Molecular ha estado sustentada en dos grandes pilares. El primero, establecido en la segunda parte de los setenta, consistió en el desarrollo de los métodos de secuenciación de ácidos nucleicos. El segundo pilar, fue la posibilidad de la amplificación «in vitro» de secuencias mediante la reacción en cadena de la polimerasa (PCR), que revolucionó a finales de los ochenta, toda la tecnología del ADN recombinante. Al cabo de otra década, cuando parecía que las bases técnicas de la Biología Molecular estaban firmemente establecidas, surge una nueva tecnología que potenció un nuevo desarrollo del análisis molecular. La técnica, conocida como *microarray*.

Desde la publicación de los primeros trabajos a finales de los 90 se ha incrementado y diversificado el número de preguntas abordadas por medio de esta técnica.

1.2.1 ¿Qué es un *microarray*?

Un concepto básico en *microarray* es el posicionamiento preciso en un soporte sólido de elementos que funcionen como detectores moleculares en altas densidades (Figura 1). En la práctica, los *microarrays* abarcan una amplia gama que puede tener diferentes soportes (membrana o vidrio) y diferentes moléculas que interaccionan en este medio.

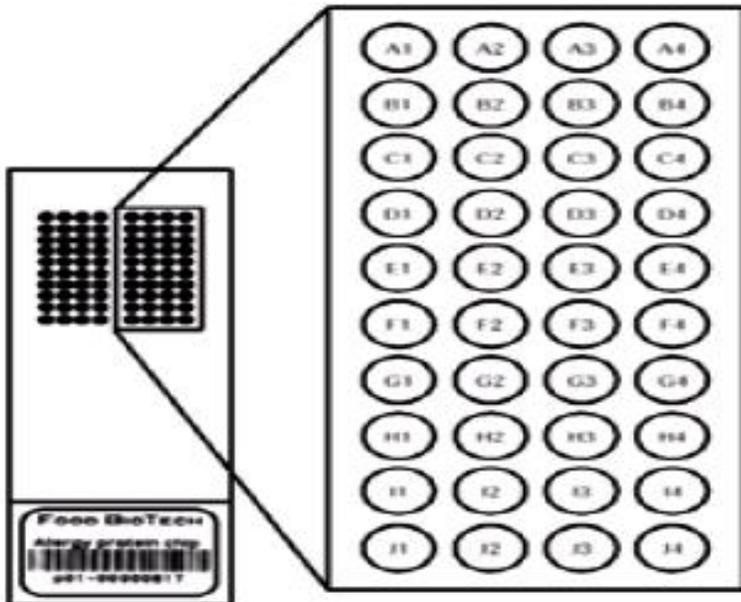


Figura 1 Muestra de imagen de *microarray* en un soporte de vidrio.

1.2.2 Construcción y funcionamiento de un *microarray*

La construcción de un *microarray* consiste en colocar en el mismo, cada uno de los genes cuyos niveles de expresión pueden ser cuantificados. Para ello se sintetiza el material genético y se insertan de forma automática en una capa de cristal, silicio o plástico, colocándose en unas casillas que actúan a modo de tubo de ensayo. Después se hibrida y se eliminan todas las cadenas que no se han unido mediante lavados, sólo las moléculas que hibridan permanecerán en el *microarray* y se procede al revelado mediante un escáner óptico o con microscopía láser confocal. Luego la imagen obtenida es procesada computacionalmente y se alcanza un patrón de intensidades en cada casilla. (Figura 2).

En los estudios de *microarrays* sólo habrá fluorescencia en los puntos del portaobjeto donde haya ocurrido hibridación y la intensidad de la fluorescencia detectada será proporcional al nivel de expresión del gen que se analiza (7).

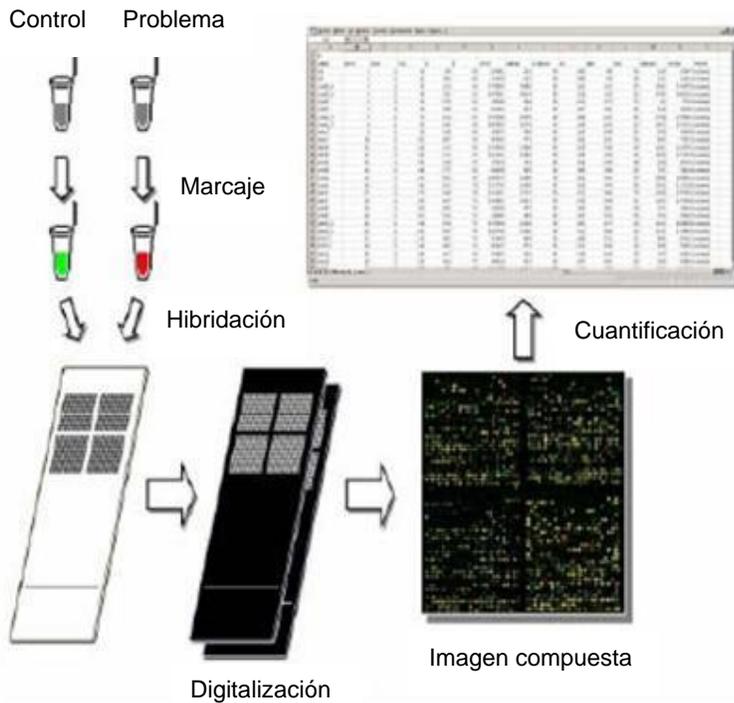


Figura 2 Representación de experimento básico de *microarray*.

1.2.3 ¿Qué tipo de *microarrays* existen?

Para poder clasificar los *microarrays* debemos dejar claro qué se entiende por blanco y qué se define como sonda. Existen controversias al respecto pero se tomará por blanco a la molécula libre y como sonda a la inmovilizada.

Existen diversos tipos de *microarrays* (8) según las sondas utilizadas que abarcan metodologías muy variadas, desde lo más rudimentario hasta lo más sofisticado. De los tipos más comunes, en el CIGB el *microarray* de ADN es la técnica actual:

Microarrays de proteínas

Las sondas son anticuerpos fijados a un portaobjetos de vidrio y los blancos son muestras de suero o tejido. Esta técnica se ve por el momento restringido por varios puntos, entre ellos podemos mencionar la dificultad de fabricar e inmovilizar estructuras 3-D como son las proteínas y detectar interacciones de proteínas plegadas, sin olvidar mencionar que no se dispone aún de colorantes fluorescente que permitan cuantificar eficientemente a estas moléculas.

Microarrays de tejidos (TMA)

Esta técnica trata de resolver uno de los problemas principales y limitantes en el análisis molecular de tejidos: el tamaño limitado de la muestra. Se utiliza una aguja hueca para tomar muestras milimétricas de las regiones de interés de tejidos embebidos en parafina, en especial biopsias. Luego se depositan de manera ordenada en un nuevo bloque de parafina y se cortan con un micrótopo entre 100-500 veces y se reordenan sobre portaobjetos de vidrio donde se realizarán pruebas múltiples a nivel ADN, ARN y proteínas. Es una técnica costosa y poco práctica si se necesitan resultados inmediatos.

Microarrays de ADN

Esta técnica, muy usada en biología molecular y conocida en este ámbito con el calificativo de «*microarray*», consiste en la colección de fragmentos de ADN en una superficie sólida monitorizándose los niveles de expresión de ellos de forma simultánea. Las superficies empleadas para fijar el ADN son muy variables y pueden ser de vidrio, plástico e incluso de silicio. Este tipo de *microarray* es utilizado para identificar genes que producen ciertas enfermedades mediante la comparación de los niveles de expresión entre células sanas y células que están desarrollando otras enfermedades. Desde un punto de vista de longitud de la sonda usada, hay 2 tipos de *microarrays* de ADN: *microarrays* de cADN que presenta sondas de cientos de bases y de oligonucleótidos con sondas de 20-25 bases o 50-80.

Microarray de oligonucleótidos

En los *microarray* de oligonucleótidos, las pruebas son designadas a partir de una secuencia conocida o un ARN predicho. Estos *microarrays* dan estimaciones del nivel de expresión, pero por distintas condiciones no pueden ser observados en una misma matriz, por lo que por cada condición se ha de utilizar un chip.

Microarray de cADN

En este tipo de *microarray* se hibrida el cADN (ADN copia) de dos condiciones que son marcados, cada uno de esas condiciones con dos fluoróforos diferentes. Las condiciones son mezcladas e hibridadas en el mismo *microarray*. Una vez realizado este primer paso se procede a escanear el resultado y a visualizar el mismo. De esta forma se pueden observar genes que se activan o se reprimen en distintas condiciones. En estos experimentos no se pueden observar niveles absolutos en la expresión.

1.2.4 Aplicaciones de los *microarrays*

Las aplicaciones de esta tecnología fundamentalmente se localizan en el área biomédica, farmacéutica y clínica (9). Esta capacidad de análisis masivo y simultáneo es la que ha permitido un salto en las posibilidades que la técnica de *microarray* ofrece, no sólo cuantitativo (se analizan muchos más genes), sino cualitativo: el poder de la técnica radica en la posibilidad de aportar una visión molecular global de un determinado proceso patológico, por ejemplo de un tipo tumoral concreto. Este conocimiento global permite entrever el funcionamiento orquestado de los genes, como ellos y las proteínas para las que codifican, interaccionan entre sí en una circunstancia determinada, que es diferente entre un estado normal y uno patológico.

Los *microarrays* se han aplicado al estudio de casi cualquier tipo de problema biológico, esto se evidencia en estudios de genes que se expresan diferencialmente y la búsqueda de terapias más efectivas contra el cáncer. Los estudios se realizan según las condiciones siguientes:

- Sanos/enfermos, tratados/no tratados.
- Clasificación molecular en enfermedades complejas.
- Identificación de genes característicos de una patología.
- Predicción de respuesta a un tratamiento.
- Detección de mutaciones y polimorfismos de un único gen (SNP).

Actualmente, la aplicación más frecuente de los *microarrays* es en el análisis genético, pero en un futuro próximo sus aplicaciones serán mucho más amplias.

1.3 Imágenes digitales de un *microarray*

La formación de una imagen digital es el primer paso para cualquier procesamiento de imágenes, y consiste básicamente en un sistema óptico y el digitalizador, mediante el cual la imagen óptica se transforma en una señal eléctrica que permitirá el procesamiento.

Una imagen digital es una función discreta bidimensional $f(x,y)$ la cual ha sido muestreada (discretizada en ambos ejes (x,y)) y cuantizada en amplitud ó intensidad.

Puede ser considerada como una matriz cuyas filas y columnas x e y identifican un punto en la imagen y el valor de cada elemento de la matriz identifica el nivel gris en ese punto. Los elementos de tal arreglo digital son llamados elementos de imagen, elementos de cuadros o *píxeles*.

Las imágenes digitales se pueden dividir en cuatro tipos (10): *bitmapped*, monocromática, *color-mapped* y *true-color*.

El término blanco y negro se utiliza para describir imágenes binarias. En las imágenes binarias -tipo de imagen obtenida en el CIGB- los niveles de intensidad son 0 para el negro (intensidad nula) y 255 para el blanco (intensidad máxima). Así mismo una imagen digital puede ser representada por una matriz I de tamaño $M \times N$:

$$I = \begin{pmatrix} I(1,1) & \cdots & I(1,M) \\ \vdots & \ddots & \vdots \\ I(N,1) & \cdots & I(N,M) \end{pmatrix}$$

Las columnas y las filas de la matriz de la imagen, por tanto, tiene un rango de:

$$0 \leq i \leq N - 1 \quad 0 \leq j \leq M - 1$$

En un arreglo bidimensional de este tipo los índices i, j de un sistema coordenado en una matriz corresponden a los ejes x, y respectivamente.

El archivo digital de un *microarray* se denomina *bitmap*, provee una representación numérica biunívoca de los valores de fluorescencia de cada *píxel*. Una imagen digital *bitmapped* es una imagen que ha pasado por un proceso de conversión, para que pueda ser almacenada en forma de mapa de bits. El formato estándar de estos archivos es TIFF (*tagged image file format*). A continuación se hace alusión al tipo de formato que se utiliza en el caso particular del CIGB y las características de este:

TIFF: se considera como formato de archivo de imágenes con etiquetas, esto se debe a que estos ficheros contienen, además de los datos de la imagen propiamente dicha, etiquetas en las que se archiva información sobre las características de la imagen, que sirven para su tratamiento posterior (10). Estas etiquetas describen el formato de las imágenes almacenadas, que pueden ser de distinta naturaleza:

- Binarias: pueden ser especialmente usadas en procesos de detección de contornos y extracción de intensidades.
- Niveles de gris: adecuadas para imágenes de tonos continuos como fotos en blanco y negro.

- Paleta de colores: adecuadas para almacenar diseños gráficos con un número limitado de colores.
- Color real: adecuadas para almacenar imágenes de tono continuo, como fotos en color.

1.4 Tratamiento de imágenes. Procesamiento de imágenes de *microarray*

El tratamiento digital de imágenes comprende el procesamiento y el análisis de imágenes. El procesamiento está referido a la realización de transformaciones, la restauración y mejoramiento de las imágenes. El análisis consiste en la extracción de propiedades y características de las imágenes, así como la clasificación, identificación y reconocimiento de patrones (11). La importancia del procesamiento y análisis digital de imágenes se encuentra en:

- Desarrollo de aplicaciones en biología y medicina de gran importancia social. El biólogo molecular puede presentar cuadros de moléculas y ganar conocimientos penetrando en su estructura.
- Incremento del potencial científico-técnico y de la capacitación de especialistas.
- Aumento del volumen de información para el procesamiento digital y análisis de imágenes con amplia repercusión social (tv, vídeo, medicina, biología, geología, etc.).

1.4.1 Preprocesamiento

El preprocesamiento comprende algoritmos cuya finalidad es conseguir una mejora en la apariencia de la imagen original. Se resaltan determinadas características o se elimina aquello que las oculta. El resultado de esta etapa es que la imagen pueda ser analizada de una manera simple de modo que podamos reducir el ruido: mediante la mejora del contraste, brillo, borrado de puntos negros aislados, etc.

El ruido se define como el proceso de alteración de la información original por factores no controlables. Si consideramos a la imagen como un conjunto de información, tenemos que considerar que al momento de captarla o procesarla, se pierde un porcentaje de la información original.

1.4.2 Métodos de procesamiento

El procesamiento de las imágenes escaneadas de *microarrays* (12) puede ser dividido en tres tareas:

- Grillado: es el proceso de asignación de coordenadas a cada uno de los *spots*.

- Segmentación: clasificación de los *píxeles* en *foreground*¹ y *background*².
- Extracción de la intensidad: calcular para cada *spot* del arreglo las intensidades de fluorescencia y medidas de calidad del *spot*.

1.4.2.1 Grillado

La estructura básica de la imagen está determinada por la construcción del *microarray*. Refiriéndose a la cantidad de filas y columnas que tiene la *grid*, así como en cada *subgrid*. Para extraer las características de la *grid* el software requiere la información siguiente:

- Cuántas *subgrids* tiene el arreglo en cada dirección (*x* e *y*).
- Cuántos *spots* hay por *subgrids* en cada dirección (*x* e *y*).
- Espaciado entre las filas y las columnas de la *grid*.

1.4.2.2 Segmentación

La segmentación de una imagen puede definirse como el proceso de particionar la imagen en diferentes regiones de diferentes propiedades. En un experimento de *microarray* la segmentación permite clasificar los *píxeles* en *foreground* o *background*, de manera que se puedan calcular intensidades como medidas de la abundancia de la secuencia transcrita para cada secuencia de ADN que se encuentre en la muestra. Todos los procedimientos de segmentación producen una máscara del *spot* (*spot mask*) que consiste en el conjunto de los *píxeles* donde se supone se encuentra un *spot*. Aunque se mencionan otros métodos se hará uso en el trabajo de la segmentación por umbral.

Los métodos de segmentación pueden clasificarse en:

- Círculo fijo.
- Círculo variable.
- Forma variable.
- Histograma (basada en el umbral).

¹ Intensidad correspondiente al *spot*.

² Intensidad lumínica del fondo.

Segmentación con círculos fijos: Superpone un círculo del mismo tamaño a todos los *spots* de la imagen. Este método es fácil de implementar y funciona bien cuando todos los *spots* son circulares y del mismo tamaño. Pero cuando los *spots* tienen forma variable, como ocurre en la mayoría de los *microarrays*, tiende a dar resultados imprecisos.

Segmentación con círculos de tamaño variable: En este tipo de segmentación, el diámetro del círculo es estimado en forma separada para cada *spot*. Algunos software proveen la opción de ajustar a mano el diámetro de los círculos, *spot* por *spot* pero esto lleva mucho tiempo. Además en la práctica los *spots* rara vez son circulares y una máscara circular tendrá un mal ajuste. Las fuentes de no circularidad incluyen los procesos de la impresión o el post procesamiento de los portaobjetos luego de la impresión.

Forma variable: El crecimiento de regiones como usualmente se le conoce, es uno de los métodos conceptualmente más simples para la segmentación; usado para extraer regiones de la imagen que están conectadas según cierto criterio predefinido. Este criterio puede estar basado en información de intensidades y/o bordes de la imagen. En su forma más simple, este método requiere un punto semilla (*seed point*) que es seleccionado manualmente por el usuario, y extrae todos los *píxeles* conectados a la semilla, que tengan el mismo valor de intensidad, es decir los *píxeles* adyacentes de amplitud similar se agrupan juntos para formar una región segmentada. Este método es poco implementado debido a su complejidad computacional, además la intervención del usuario en el proceso de selección puede incurrir en errores.

Segmentación por histograma (basada en umbral): Es el proceso de segmentación más simple. La mayor parte de las regiones de interés en una imagen están caracterizadas por una reflectividad o absorción de luz más o menos constante de su superficie (los *píxeles* de un *spot* tienen, aproximadamente, el mismo nivel de gris). La idea es obtener un umbral de brillo³ que separe el *spot* del fondo.

Dado el umbral T , la imagen umbralizada se obtiene mediante la expresión:

$$g(x, y) = \begin{cases} 1 & f(x, y) \geq T \\ 0 & f(x, y) < T \end{cases}$$

³ Valor del nivel de gris.

Algoritmo de umbralización: recorrer todos los *píxeles* de la imagen; si su nivel de gris es mayor que el umbral es parte del *spot*, si es menor es parte del fondo. La umbralización es rápida computacionalmente. La parte más difícil es encontrar el valor correcto del umbral.

La selección del umbral correcto es crucial en el proceso de segmentación por umbralización. La selección del umbral puede ser interactiva o puede ser el resultado del algún método de detección de umbral, el que se explica a continuación es el seleccionado para el desarrollo de la aplicación, no obstante se hace mención de otras elecciones de umbral (13):

- Segmentación umbralización p -cubrimiento.
- Segmentación por umbralización de histograma.
- Segmentación umbralización óptima.
- Segmentación umbralización multiespectral.
- Segmentación umbralización jerárquica.

Segmentación por umbralización de histograma: Dado que el histograma representa la distribución de probabilidad de que un *píxel* tenga un nivel de gris (u otra propiedad) dado y que es razonable suponer que un objeto tiene un rango de variación en la propiedad pequeño, es lógico suponer que un objeto de la imagen vendrá representado por un máximo local en el histograma. El proceso segmentación se convierte en la búsqueda de picos en el histograma. Este tipo de segmentación se implementa para buscar mínimos locales entre máximos locales significativos y utilizar el nivel de gris de esos mínimos como umbral de segmentación.

Esta estrategia de segmentación es sencilla. En primer lugar se localiza el mayor máximo local del histograma⁴. A continuación se localiza el segundo mayor máximo local del histograma. Por último, se localiza el mínimo local entre ambos máximos y se utiliza su valor como valor para el umbral.

1.4.2.3 Extracción de la intensidad

A partir de los *píxeles* que representan los *spots* y su fondo, los software de procesamiento de imágenes deben calcular medidas numéricas para cada parte. A continuación describimos las medidas incluidas en la mayoría de ellos, aunque los 6 primeros fueron los utilizados en la cuantificación de la imagen.

⁴ Corresponderá a un máximo de probabilidad asociado a un spot o al fondo de la imagen.

1. Media de la *spot*: media de las intensidades los *píxeles* que constituyen el *spot*.
2. Media del fondo: media de las intensidades de los *píxeles* determinados como *background*.
3. Mediana de la *spot*: mediana de las intensidades de los *píxeles* que constituyen el *spot*.
4. Mediana del fondo: mediana de las intensidades de los *píxeles* determinados como *background*.
5. Desvío estándar de la *spot*: de las intensidades de los *píxeles* que constituyen el *spot*.
6. Desvío estándar del *background*: de las intensidades de los *píxeles* que constituyen el *background*.
7. Medidas de la forma del *spot*: cantidad de *píxeles* que lo conforman. Medida de circularidad $=4 \pi$ área / (perímetro²).
8. Marca (*flag*), indica la calidad del *spot*.

La medida más importante es la intensidad del *spot* que puede elegirse entre la media y la mediana, pero esta última es la más recomendable.

Intensidad del *spot*: cada valor de *píxel* en una imagen escaneada representa el nivel de hibridación en una posición del vidrio. La cantidad total de hibridación para una secuencia particular de ADN es proporcional a la fluorescencia total del *spot*. Por lo tanto una medida natural de la intensidad del *spot* es la suma de las intensidades de los *píxeles* dentro del *spot mask*. Como los cálculos posteriores se basan en el cociente de intensidades de fluorescencia, se calcula el promedio de los valores de los *píxeles* sobre el *spot mask*.

Otros valores estadísticos que se pueden calcular dentro de cada *spot mask* son, la mediana de los valores de los *píxeles* y algunas medidas de variabilidad.

Valores del fondo (*background values*) (14): la intensidad del fondo asociado a cada *spot* se puede calcular a través de métodos como:

- *Local*
- *Morphological opening*
- *Constant* (global)
- *None*

De los mencionados anteriormente se utiliza *local background* para definir el fondo circundante al *spot*. La representación gráfica de este método por distintos programas se muestra en (Figura 3). La intensidad del fondo se calcula como la suma de los *píxeles* que lo conforman.

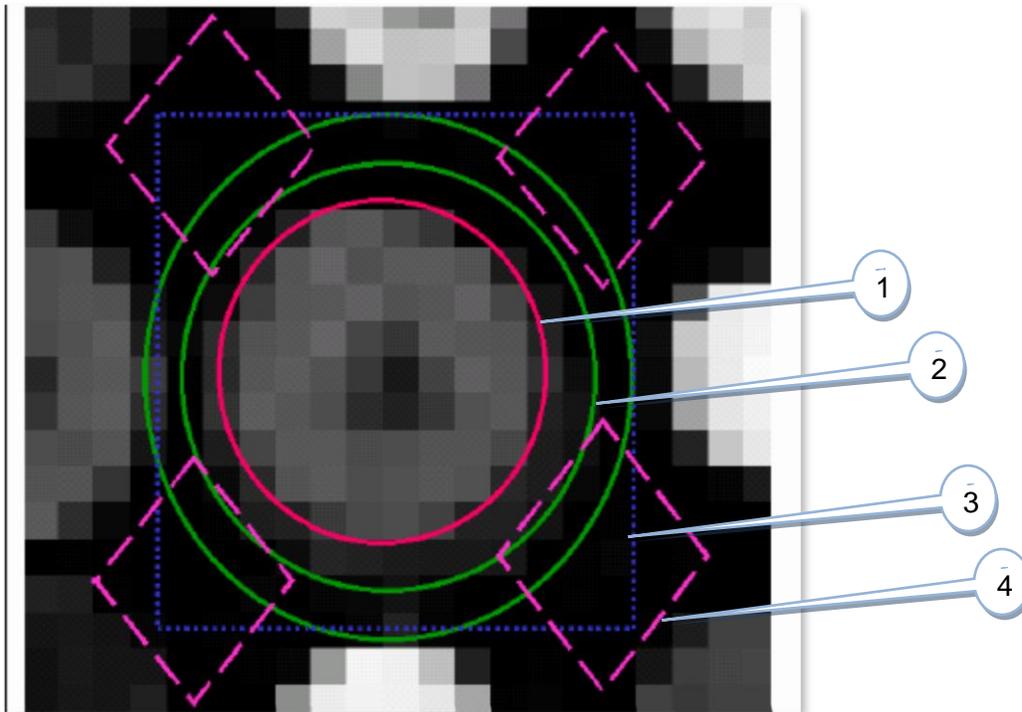


Figura 3 La región que está simbolizada con el número 1 representada la máscara del *spot* (*spot mask*) y la región simbolizada con el número 2, 3 y 4 se utiliza para calcular el fondo local (*background local*), por diferente software. 2: usado en *QuantArray*; 3: usado en *ScanAnalyze* y el 4 se usa en *ArrayVision*.

El procesamiento de imágenes tiene relevante importancia práctica en disímiles áreas de las ciencias de la vida evidenciándose en: la evaluación de fotos o imágenes con microscopios ópticos, electrónicos o *scanner*; en la observación y análisis de procesos de crecimiento bacteriano u otros; en el análisis de micro estructuras biológicas en varios rangos espectrales, en el análisis celular y cromosómico, además de mediciones por ultrasonido, tomografía axial computarizada (TAC), evaluación de imágenes radiológicas, imágenes por resonancia magnética nuclear (RMN) y elaboración de catálogos electrónicos con integración de bases de datos y bancos de imágenes.

1.5 Programas vinculados al procesamiento y cuantificación de imágenes

Respecto a este tipo de herramientas, además de cotizarse en el mercado a altos precios, no es posible agrupar el tipo, una parte de él es para el análisis y tratamiento de las imágenes y otra parte especializada en la minería de datos. Se describe brevemente aquellos que sirvieron de referencia al trabajo y se citan otros de importancia (Tabla 1).

Tabla 1 Diferentes software para el procesamiento y cuantificación de imágenes en el mundo.

Empresa-URL	Producto	Características
http://www4.amershambiosciences.com	<i>ImageQuant</i>	Análisis de imágenes de <i>microarray</i> .
http://www.genomicsolutions.com/genomics	<i>QuantImage</i>	Análisis de imágenes de <i>microarray</i> .
http://www.bipl.ahc.umn.edu/quantarray.html	<i>QuantArray</i>	Análisis de imágenes de <i>microarray</i> .
http://www.imagingresearch.com/ArV	<i>ArrayVision</i>	Cuantificación de imágenes de arrays.
http://www.molecularware.com/	<i>ArrayAnalyzer</i>	Análisis de imágenes de <i>microarrays</i> .
http://graphics.stanford.edu/software/	<i>ScanAlyze</i>	Análisis de imágenes de <i>microarrays</i> .
http://www.tigr.org/software/	TIGR <i>Spofinder</i>	Análisis de imágenes de <i>microarrays</i> .

ArrayVision: paquete de software flexible que se ha desarrollado para la cuantificación de la expresión genética y de los arreglos proteínicos de los genes. Proporciona un análisis rápido y automatizado para las imágenes de arreglo. *ArrayVision* utiliza protocolos configurables para satisfacer cualquier formato de arreglos.

Ofrece la opción de cálculo de métrica de la calidad basada sobre el *spot*. Requiere como sistema operativo *Windows NT® 4.0* o *Windows 2000®*. El software estará disponible en una sola computadora o versión del servidor. Este es fabricado por una compañía canadiense que impone un alto precio de adquisición. Otra de sus desventajas es que *ArrayVision* incurre en errores de distorsión geométrica que implica errores en la cuantificación de la imagen (1).

TIGR *Spotfinder*: fue diseñado para el análisis rápido, reproductivo y automatizado de imágenes *microarray* y la cuantificación de la expresión del gen. Lee los archivos 16-bit o 8-bit, cuenta con dos métodos disponibles de la segmentación (uno de ellos es mediante histograma) definen los límites entre cada punto y el fondo local circundante. Las intensidades del punto se calculan como integral de *píxeles* no saturados, aunque dispone de otras opciones que incluyen valores medianos y medios del punto. TIGR *Spotfinder* fue escrito en C++.

La construcción semiautomática de la *grid* define las áreas de la diapositiva donde están los puntos. Los ajustes automáticos y manuales de la *grid* ayudan a asegurarse de que cada célula rectangular de la *grid* está centrada en un punto, sin embargo *Spotfinder* no incluyen la información de la anotación para cada gen, no ha sido probada aún para *Windows*, además para adquirir este software es necesario tomar el paquete completo de TM4 que consta de cuatro usos importantes, el encargado de los datos de *microarray* (MADAM), TIGR_ *Spotfinder*, el sistema del análisis de datos de *microarray* (MIDAS), y el espectador de *Multiexperiment (meV)*. Perteneciente al Instituto de Investigación de Genómica, *Maryland, USA*.

1.6 Importancia del procesamiento y cuantificación de imágenes de *microarray*

Los programas *ArrayVision* y *Spotfinder*, sirvieron como precedentes para el desarrollo de una aplicación propia, que permita de manera eficiente el procesamiento de toda información procedente de la investigación biológica plasmada en una imagen de *microarray*, así como la cuantificación rápida y automática de las imágenes mediante diferentes funciones matemáticas, métodos de análisis y procesamiento. Con la finalidad de que sea útil en todos los centros científicos cubanos dedicados al estudio de la extracción de información de los genes, aunque en principio esté orientado a facilitar el trabajo de los investigadores del CIGB.

De manera general el procesamiento y cuantificación de imágenes ofrece una visión más global del estudio de los genes y además de la implicación en el aumento de la eficiencia del descubrimiento y desarrollo de nuevos fármacos, y la resistencia de enfermedades a drogas determinadas.

1.7 Metodologías de desarrollo

Todo desarrollo de software es riesgoso y difícil de controlar, es necesario una metodología de desarrollo (15) de software de por medio, que permita obtener un resultado con calidad, de ahí que tanto desarrollador como cliente queden satisfechos.

Extreme Programming, (XP) es una metodología que consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Utilizada para proyectos de corto plazo y que estén conformados por un pequeño equipo. Lo fundamental en este tipo de metodología es:

- La comunicación, entre los usuarios y los desarrolladores.
- La simplicidad, al desarrollar y codificar los módulos del sistema.
- La retroalimentación, concreta y frecuente del equipo de desarrollo, el cliente y los usuarios finales.

La metodología XP es exitosa porque enfatiza la satisfacción del cliente y promueve el trabajo en equipo. XP propone un proceso de desarrollo liviano, eficiente, de bajo riesgo, flexible, predecible y científico, sin embargo, a pesar de brindar grandes ventajas y gran facilidad de uso, esta metodología requiere de un contacto directo y continuo con el usuario final, por lo cual no es factible para nuestro problema.

Además la carencia de la documentación del diseño con que cuenta XP, limita las oportunidades del trabajo para aprovechar la reutilización, necesaria para suscitar posibles prestaciones en el procesamiento de imágenes de *microarray*. En XP los métodos de ingeniería se describen brevemente, esto conlleva a que se adopten determinados elementos sin hacer caso del resto, en la práctica esto falla y particularmente la descripción a detalle de los métodos para procesar una imagen y extraer determinados valores de intensidades, es vital para la comprensión y realización de una solución afín.

Sin embargo, *Rational Unified Process, (RUP)* es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto, obviamente se emplea un grado variable de artefactos en cada flujo de trabajo que lo caracteriza (16).

Entre las principales características que distinguen el ciclo de vida de RUP están: dirigido por casos de uso, iterativo, incremental y centrado en la arquitectura.

- Dirigido por casos de uso: Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos

que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).

- **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. RUP se desarrolla mediante iteraciones, comienza por los casos de uso relevantes desde el punto de vista de la arquitectura. El modelo de arquitectura se representa a través de vistas en las que se incluyen los diagramas de UML.
- **Iterativo e incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto.

Además divide en fases el desarrollo del software, y cada una de esas etapas se desarrolla mediante un ciclo de iteraciones.

- **Conceptualización (Concepción o Inicio):** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los requerimientos (funcionales y no funcionales) identificados de acuerdo al alcance definido.
- **Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene uno o varios *release* del producto que han pasado las pruebas. Se ponen estos *release* a consideración de un subconjunto de usuarios.
- **Transición:** El *release* ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

Cada una de estas fases comprende 9 flujos de trabajo. Los 6 primeros son flujos de ingeniería y los tres últimos son conocidos como flujos de apoyo (17).

Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo

- Ingeniería de Negocios: se entienden las necesidades del negocio.
- Requerimientos: se trasladan las necesidades del negocio a un sistema automatizado.
- Análisis y Diseño: se trasladan los requerimientos dentro de la arquitectura de software.
- Implementación: se crea software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
- Pruebas: se asegura que el comportamiento requerido es el correcto y que todo lo solicitado esta presente.

Disciplina de Soporte

- Configuración y administración del cambio: se guardan todas las versiones del proyecto.
- Administrando el proyecto: se administran horarios y recursos.
- Ambiente: se administra el ambiente de desarrollo.
- Distribución: se hace todo lo necesario para la salida del proyecto

Se concibe como obligatorio el uso de artefactos en cada ciclo de iteración. Se centra la atención en la construcción de cada artefacto. Se cumplen los objetivos trazados desde el Modelamiento del Negocio hasta el Análisis y Diseño, sin obviar que los métodos de procesamiento están debidamente justificados en cuanto a la terminología empleada y la descripción misma de las actividades.

Los argumentos expuestos conducen a la selección de RUP como metodología de desarrollo para producir una mejor gestión global del proyecto, considerando las fases e iteraciones de RUP y sobretodo la documentación con los artefactos.

Este tipo de documentación es vital pues el trabajo tendrá continuidad en flujos de trabajo posteriores al análisis y diseño, además de la importancia para la comprensión de los métodos de procesamiento de imágenes de *microarray*. Garantizando que en la disciplina de Análisis y Diseño se concentren las

actividades en un flujo de trabajo determinado para producir información basado en los requerimientos del sistema.

El Proceso Unificado utiliza el Lenguaje Unificado de Modelado (UML) para preparar todos los esquemas de un sistema software. De hecho, UML es una parte esencial del Proceso Unificado. RUP emplea UML como base para el desarrollo de software en cada una de sus fases y disciplinas.

Abordar el modelado de un sistema exige saber de antemano qué artefactos de la notación del Proceso Unificado de Modelado (UML) van a ser más relevantes para representar su estructura estática y cuáles serán más apropiados para documentar su comportamiento, o bien, la dinámica de sus estados.

UML es un lenguaje gráfico para visualizar, construir, especificar y documentar los elementos de los sistemas orientados a objetos. Notación unificada con la que se permite lograr un entendimiento que propicie el intercambio entre los usuarios y los desarrolladores.

Se ha convertido en un estándar de la industria del software, debido a que ha sido impulsado por los autores de los tres métodos más usados de orientación a objetos *Grady Booch, Ivar Jacobson y Jim Rumbaugh*.

El UML estándar está compuesto por tres partes: bloques de construcción (tales como clases, objetos, mensajes), relaciones entre los bloques (tales como asociación, generalización) y diagramas (por ejemplo, diagrama de actividad). Cubre la especificación de las decisiones de análisis, diseño e implementación que se realicen al desarrollar y desplegar un sistema de software.

1.7.1 Herramientas CASE

Las herramientas CASE de modelado con UML nos permiten aplicar la metodología de análisis y diseño orientados a objetos y abstraernos del código fuente, en un nivel donde la arquitectura y el diseño se tornan más obvios y más fáciles de entender y modificar.

Rational Rose Enterprise Edition 2003, es una herramienta que permite especificar, analizar, diseñar el sistema antes de codificarlo. Desarrollada por los creadores de UML (*Booch, Rumbaugh y Jacobson*), cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases.

Se encuentra dentro del grupo de herramientas más técnicas debido a que se encarga de llevar a cabo tanto la automatización de los sistemas para la posterior generación de código (esto es, realización de los

distintos diagramas y generación del código posterior), como para labores de ingeniería inversa (es decir, realización de los diagramas una vez conocido el código).

El navegador UML de *Rational Rose* nos permite establecer una trazabilidad real entre el modelo (análisis y diseño) y el código ejecutable. Facilita el desarrollo de un proceso cooperativo en el que todos los agentes tienen sus propias vistas de información (vista de Casos de Uso, vista Lógica, vista de Componentes y vista de Despliegue, Vista de Procesos), pero utilizan un lenguaje común para comprender y comunicar la estructura y la funcionalidad del sistema en construcción.

Por otro lado se encuentra *Visual Paradigm* (VP) considerada como una potente herramienta para visualizar y diseñar elementos de software. Entre sus peculiaridades, el diseño centrado en casos de uso y enfocado al negocio, el entorno de creación de diagramas para UML 2.0, la sincronización en todo el ciclo de desarrollo del modelo y código, así como la disponibilidad de integrarse a los principales entornos de desarrollo (IDEs) y en múltiples plataformas. Son características que le confieren un valor determinado en el mundo actual.

Facilita la interoperabilidad con otras herramientas CASE y se integra con las siguientes herramientas *Java: Eclipse/IBM, WebSphere, NetBeans* IDE, además de que está disponible en varias ediciones: *Enterprise, Professional, Community, Standard, Modeler y Personal*. Sin embargo, herramientas como la generación automática de código no son aplicables y cualquier diagrama generado automáticamente a partir de otro diagrama podría perder su utilidad.

La selección de *Rational Rose Enterprise Edition* como herramienta CASE, se concibe como la necesidad de vincular a través de UML una traza entre el modelo de análisis y diseño propuesto, y el código ejecutable, además de garantizar la realización de los principales artefactos de los flujos de trabajo que establece la metodología de RUP.

1.7.2 Artefactos generados

Al conjunto de actividades necesarias para transformar los requisitos del CIGB en un prototipo de diseño de software deseado, las conforman los artefactos generados en los flujos de trabajo de Modelamiento del Negocio, Requerimientos y Análisis y Diseño, de su totalidad, se exponen aquellos que se obtuvieron como resultados de este trabajo, adecuándose siempre a las necesidades del CIGB:

- Modelación del dominio, como vía alternativa para describir el negocio.

- Especificación de los requisitos de software correspondientes al procesamiento y cuantificación de imágenes.
- Definición de prototipos no funcionales del software para el procesamiento de imágenes de *microarray*.
- Descripción de los procesos que se deben automatizar para procesar y cuantificar imágenes de *microarray*.
- Realización del diagrama de clases de análisis vinculado al procesamiento y cuantificación de imágenes.
- Realización del diagrama de interacción de análisis vinculado al procesamiento y cuantificación de imágenes.
- Fundamentación de los patrones utilizado en la propuesta de diseño para la aplicación de software asociada al procesamiento de imágenes de *microarray*.
- Construcción de paquetes, subsistemas e interfaces acordes al procesamiento y cuantificación.
- Realización del diagrama de clases de diseño vinculado al procesamiento de imágenes.
- Realización del diagrama de interacción de diseño vinculado al procesamiento de imágenes.

1.8 Metodología de investigación

Para realizar esta investigación, se partió del supuesto que no puede haber conocimiento al margen de la práctica. La veracidad de este juicio condujo a una búsqueda de conocimientos ordenados y de reflexión analítica, a fin de explicar el procesamiento de imágenes de *microarray*.

1.8.1 Método utilizado para recopilar información

Para la recopilación de la información utilizada en este trabajo, se referenciaron dos métodos: revisión bibliográfica y entrevistas.

Revisión bibliográfica

Se basó fundamentalmente en consultar un conjunto de fuentes de información referidas al tema, como fueron artículos (7), (11), (12), libros (16), (18), sitios web (3), (6), y toda una gama de materiales tanto impresos como digitales, también se localizaron lecturas especializadas sobre el tema para documentar la

base teórica de este trabajo. Esta fuente de información, registra o corrobora el conocimiento inmediato de la investigación, fue debidamente referenciada para consultas posteriores. Resulta de interés para aquellas personas que deseen estudiar más de los restantes capítulos de este trabajo.

Técnica de entrevista

Mediante esta forma específica de interacción social, se logró obtener información y detalles del sistema que deseaban obtener finalmente los clientes. A partir de las repuestas y aclaraciones que brindaron los especialistas del Centro de Ingeniería Genética y Biotecnología, como el Dr. Julio Hernández y la Dra. Isabel Guillén del Departamento de Investigaciones Biomédicas, se obtuvieron las informaciones que se necesitaban. Gracias a que esta técnica permite reunir datos primarios, generalmente con un carácter más práctico y menos teórico, se logró definir todos los detalles requeridos para el diseño del sistema.

1.8.2 Método utilizado para seleccionar y organizar los datos

La elección de esta técnica esta determinada por las características del estudio que se desarrolló, es decir del procesamiento y cuantificación de imágenes de *microarray*. La selección de información de la bibliografía consultada se realizó considerando a aquellas que tuviesen relación directa con el problema en cuestión. Esta se organizó utilizando el método Inductivo–Deductivo, a partir de la idea de comprender los problemas más generales y según los resultados interpretar conceptos particulares. Fue necesario además tener acceso a sistema que contuvieran información y recursos de apoyo para recopilar datos, ejemplo el software *ArrayVision*.

1.9 Tendencias y tecnologías actuales

En este epígrafe se hace un análisis de las tendencias y tecnologías más utilizadas actualmente a nivel mundial en el desarrollo de aplicaciones de escritorio, según los lenguajes de programación y entornos de desarrollo.

El tipo de tecnología a usar fue concebida por el cliente y se ha mantenido como tal, no obstante la migración a software libre es un requisito que deberá ser considerado posteriormente para lograr un mayor beneficio de esta aplicación. Se argumenta a continuación dos tipos de tecnologías, enfatizando la seleccionada por el CIGB.

Microsoft .NET es un proyecto que ha creado una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma y que permita un rápido desarrollo de

aplicaciones. Basado en esta plataforma, *Microsoft* intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado (19). Podría considerarse .NET una respuesta de *Microsoft* al creciente mercado de los negocios en entornos Web, como competencia a la plataforma *Java* de *Sun Microsystems*.

La plataforma .NET brinda un conjunto único y expansible con facilidad, de bloques interconectados, diseñados de forma uniforme y bien documentados, que permitan a los desarrolladores tener a mano todo lo que necesitan para producir aplicaciones sólidas. La tecnología .NET puede integrar los objetos COM. de forma transparente y bidireccional, significa que se puede utilizar COM desde .NET. Un objeto COM es una instancia de una clase objeto componente, o sencillamente componente (el componente asociado a una DLL), los mismos están fuertemente encapsulados, por lo que únicamente pueden ser accesibles a través de las interfaces que exportan.

Se tiene por otra parte a *Java 2, Enterprise Edition* conocido como J2EE (20) que es la edición empresarial del paquete *Java* creada y distribuida por *Sun Microsystems*. Comprende un conjunto de especificaciones y funcionalidades orientadas al desarrollo de aplicaciones empresariales. J2EE está basado en la arquitectura del lado del servidor, añade el soporte completo para componentes *Enterprise Java Beans*, el API *Java Servlets* y la tecnología *JavaServer Pages*.

1.9.1 Herramientas de desarrollo y *framework*

En este epígrafe se especifica, en base a la plataforma .NET y a un modelo matemático usado, las herramientas de desarrollo necesarias.

Microsoft Visual Studio 2005 (21) es un entorno integrado de programación para sistemas *Windows*. Proporciona una plataforma de programación eficaz en equipo para generar aplicaciones. Dentro de sus características principales están: incorpora a .NET *Framework 2.0* como el entorno robusto y plenamente funcional de la plataforma *Microsoft*, incluye nuevas funciones y mejoras en el editor de texto como la refactorización, los desarrolladores disfrutan de un entorno de edición altamente productivo, permite construir soluciones para *SQL Server 2005* utilizando herramientas visuales integradas de diseño de bases de datos.

Microsoft ofrece dos opciones para los desarrolladores profesionales trabajando en solitario o en pequeños equipos, *Visual Studio 2005 Professional Edition*, y *Visual Studio 2005 Tools para Microsoft*

Office System. Cada edición añade funcionalidad a las características básicas que incluye la edición *Standard*.

.NET *Framework* 2.0, el «*framework*» o marco de trabajo (19), incluida en la plataforma .NET, denota la infraestructura sobre la cual se reúnen un conjunto de lenguajes, herramientas y servicios que simplifican el desarrollo de aplicaciones en entorno de ejecución distribuido.

Entorno independiente del hardware para ejecutar programas en dispositivos informáticos con recursos insuficientes. Hereda la arquitectura completa de .NET *Framework* de CLR⁵, es compatible con un subconjunto de la biblioteca de clases de .NET *Framework* y contiene clases diseñadas exclusivamente para .NET *Framework* 2.0. Mejora del control de excepciones, es compatible con clases parciales, tal y como se utilizan en *Microsoft Visual Studio 2005*.

MATLAB es una herramienta para hacer cálculos matemáticos y una plataforma de desarrollo de aplicaciones totalmente integrado (análisis numérico, cálculo matricial, proceso de señal y visualización gráfica). A continuación se exponen algunas características de MATLAB:

- Cálculos intensivos desde el punto de vista numérico.
- Gráficos y visualización avanzada.
- Lenguaje de alto nivel basado en vectores, *arrays* y matrices.
- Colección de funciones de aplicación.

Dispone en la actualidad de un amplio abanico de programas de apoyo especializados, denominados *Toolboxes*, que extienden significativamente el número de funciones incorporadas en el programa principal. Insertado en esta gama, se encuentra el *toolbox* para el procesamiento de imágenes (*Image Processing Toolbox*), que por la importancia que se le confiere es imprescindible detenerse a describirlo:

- Ofrece un amplio conjunto de herramientas que le permite fácilmente manipular, procesar y analizar datos de imágenes, interactivamente mostrar pantallas de imágenes 2-D o 3-D, visualizar datos temporarios cuando es necesario, y comentar sus resultados para publicaciones técnicas. Como resultado, es fácil e intuitivo efectuar procesamiento de imágenes y operaciones de análisis

⁵ *Common Language Runtime*: entorno Común de Ejecución para Lenguajes.

tales como morfología binaria, manipulación geométrica, conversión de espacios de colores, compresión, análisis de componentes conectados y más (22).

MATLAB cuenta con un constructor para .NET que ofrece la posibilidad de interactuar con otras aplicaciones a través de la utilización de la tecnología COM sobre *Windows*, se presenta como servidor de la automatización de COM, así que podemos utilizar la interoperabilidad de COM de .NET para utilizar sus funciones: se puede trabajar directamente con un tipo biblioteca o librería específica.

1.9.2 Lenguajes de programación

Ante la presencia de tantos lenguajes, ¿en qué se basa un ingeniero de software para adoptar en un proyecto el uso de uno en concreto? Generalmente, el lenguaje se elige por preferencia personal de los clientes de una empresa, o bien, porque los desarrolladores poseen un conocimiento del mismo, lo cual es razonable. En este caso en particular el CIGB ha seleccionado como lenguaje de programación a C#, aunque se menciona brevemente a *Java* para establecer alguna comparación.

C# es un lenguaje de programación orientado a objetos desarrollado y estandarizado por *Microsoft* como parte de su plataforma .NET. Su sintaxis básica deriva de C/C++ y 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#. Entre otras características de este lenguaje se encuentra (23):

- 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).

Por otro lado, *Java* (20) es un lenguaje de programación orientado a objetos desarrollado por *Sun Microsystems*. Las aplicaciones *Java* están típicamente compiladas en un *bytecode*⁶, aunque la compilación en código máquina también es posible.

⁶ Código intermedio más abstracto que el código máquina.

El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel como punteros. Sin embargo, las aplicaciones *Java* necesitan gran cantidad de memoria física. *Sun Microsystems* proporciona una implementación GNU *General Public License* de un compilador *Java* y una máquina virtual *Java*.

1.10 Conclusiones

Tras la realización de una investigación en detalle acerca del procesamiento y en específico la cuantificación de imágenes de *microarray*, enmarcando los requisitos con los que debe contar el propio software en el ámbito internacional, evaluando y seleccionando técnicas, métodos y algoritmos para trabajar con la calidad esperada, estimando el alto costo de algunas funcionalidades y por ende el freno que significan la implementación de estas, en la actual situación económica del país. Se ha concebido el diseño de un software propio (nacional) para el procesamiento de imágenes de *microarray*, del CIGB, analizando los valores de expresión génica; usando la metodología de RUP con lenguaje UML, y como herramienta de modelado *Rational Rose Enterprise Edition 2003*, y contando con MATLAB como plataforma de apoyo para el procesamiento de imágenes. Utilizar la plataforma *Microsoft .NET*, específicamente *Microsoft Visual Studio .NET 2005* y como lenguaje de programación C# los cuales brindan una gran productividad elemento necesario dado el corto período de entrega del software.

CAPÍTULO 2 CARACTERÍSTICAS DEL SISTEMA

2.1 Introducción

En el capítulo anterior quedó determinado el tipo de metodología a usar, la metodología RUP, como guía para la construcción del análisis y diseño de este sistema, que se propone desplegar en el Centro de Ingeniería Genética y Biotecnología. Como objetivo principal en este capítulo, se hará una descripción general del objeto de estudio y de los artefactos más importantes para la actividad de negocio y las referentes al sistema (Figura 4); se tiene como antecedente la fundamentación anterior y las necesidades de la institución científica; sin obviar los requerimientos mínimos a los que está sujeta la herramienta en dependencia de donde se instale.

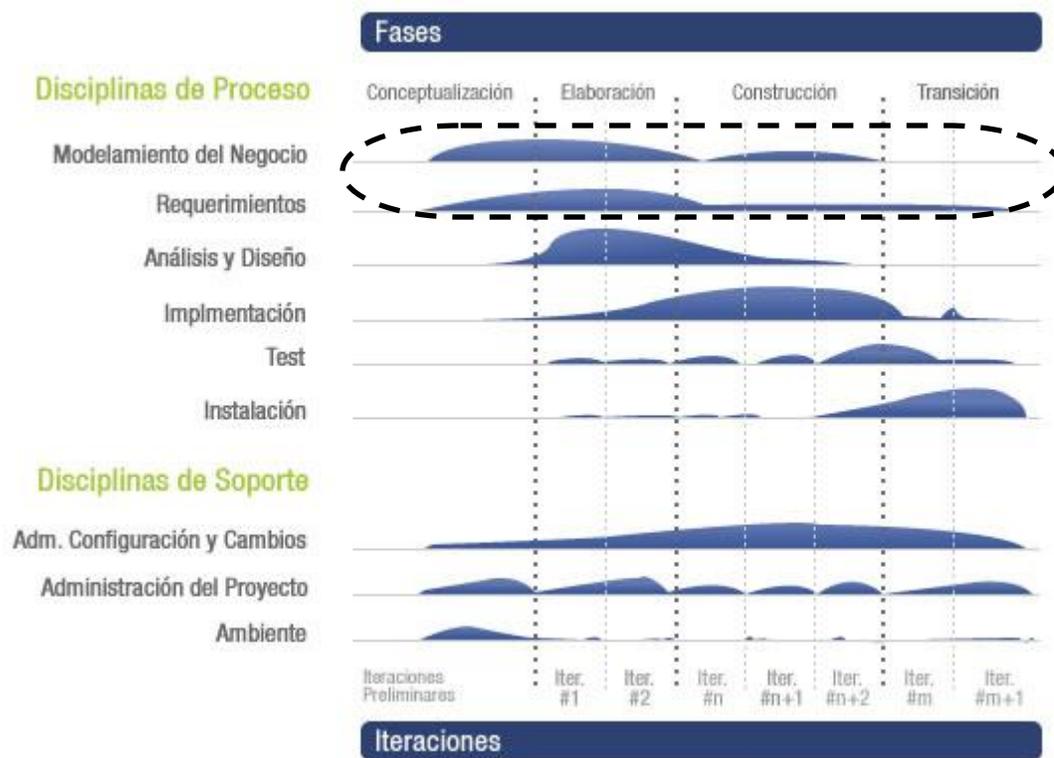


Figura 4 Representación del flujo de trabajo de Modelamiento de Negocio y Requerimientos durante el ciclo de RUP.

2.2 Objeto de Estudio

Este trabajo está condicionado por la necesidad del polo científico, CIGB, de mejorar el procesamiento imágenes de *microarray* para mejorar cualitativamente los estudios de enfermedades de implicación genética en la población cubana, asociado a este problema el desarrollo del trabajo se ha centrado en el estudio del procesamiento y análisis de imágenes de *microarray*.

2.2.1 Flujo actual de procesos

En el CIGB, se localizan los recursos básicos para procesar cada una de las imágenes de *microarray* obtenidas. Todo el proceso de análisis de la expresión de genes se desarrolla a través de diferentes etapas entre las que se encuentran la fabricación y diseño del *microarray*, la preparación de la sonda a utilizar en la hibridación del soporte, la colección de las muestras, su procesamiento y la lectura e interpretación de las intensidades de las señales.

El proceso en su conjunto, como tendencia mundial, se realiza con un equipamiento cada vez más sofisticado y costoso que no muchos laboratorios poseen, tal es el caso del CIGB, es importante contar con equipos que controlen con exactitud parámetros como la temperatura y el tiempo de hibridación, además es necesario un escáner lector de fluorescencia, quimioluminiscencia y/o radiactividad de gran sensibilidad y altamente programable. Los resultados, imágenes de *microarray* reveladas, son procesados por herramientas bioinformáticas (Figura 5).



Figura 5 La figura representa los cinco componentes fundamentales para el diseño, hibridación, cuantificación y análisis de los *microarray*.

El Centro de ingeniería Genética y Biotecnología precisa de un software que le facilite el trabajo con las imágenes que manualmente obtienen, pues las mismas carecen de la calidad que requieren otras aplicaciones para exponer sus resultados; además la propuesta de la nueva aplicación elimina en gran medida las restricciones impuestas por *ArrayVision* en cuanto a la cuantificación de este tipo de imagen.

Mucha de la información relevante acerca de la función de los genes está contenida en la literatura médica; esto no resta relevancia al diseño de una herramienta propia, para la cuantificación de este tipo de imágenes que daría un impulso en las investigaciones biomédicas del país.

2.2.2 Análisis crítico de la ejecución de procesos

La primera necesidad que origina el desarrollo de este trabajo; es que no se dispone de una herramienta con la calidad requerida para procesar las imágenes de *microarray* en el tiempo estimado, dado por el monto del costo y las constricciones en la comercialización que impone el bloqueo para adquirir aplicaciones como estas, disponible en países desarrollados. Todo lo cual provoca retrasos en el análisis

de las muestras adquiridas así como insuficientes posibilidades de automatización del proceso, dando lugar al trabajo manual por parte del investigador susceptible de incurrir en errores.

Otra de las causas que origina la situación problemática; se debe a que el software llamado *ArrayVision* no respondió satisfactoriamente a los requerimientos del CIGB, en cuanto a: los valores de hibridación pueden ser identificados en posiciones incorrectas dentro de la imagen y, los elementos de la plantilla configurable por el usuario pueden no corresponder con la posición geométrica del *spot* en la imagen.

La alineación manual de la plantilla con los *spots* es poco rigurosa y puede introducir errores significativos, puesto que un observador humano es altamente propenso a fallos de precisión.

Por último, en relación con *ArrayVision* los resultados deben ser organizados manualmente, conllevando a otros errores, sea para comprobar su validez o para posterior análisis y estudios de ellos.

De manera general no se puede garantizar la calidad del resultado final del proceso, por lo que se hace necesario diseñar un software para mejorar los tiempos de respuestas y la fiabilidad del procesamiento de imágenes de *microarray*.

Con lo expuesto anteriormente se evidencia la necesidad de analizar y diseñar un software que se encuentre exento de este tipo de dificultad, garantice la calidad en el procesamiento de imágenes y facilite el manejo de la herramienta.

2.3 Objeto de automatización

Desde el año 1999 el CIGB inició negociaciones con Canadá para obtener el software *ArrayVision* y no fue hasta el 2002 que lo adquirió por un tercer país. *ArrayVision* se ha desarrollado para la cuantificación de la expresión genética y de los arreglos proteínicos de los genes, proporciona un análisis rápido y automatizado para las imágenes de arreglos en unos cuantos pasos. Acepta datos de casi cualquier sistema de detección e incluye procedimientos para la alineación automatizada. Ofrece la opción de cálculo de métrica de la calidad basada sobre el *spot* (cuando un valor de intensidad es «bueno»). Las imágenes pueden ser indicadas con letras, combinado, o filtrado de muchas maneras, además ofrece variedad amplia de las ayudas de los formatos de la imagen, entre otras.

Sin embargo, algunas de las prestaciones de este software no contribuyeron con los requisitos que exigía el centro, esta herramienta resulta inasequible al investigador e incurre en algunos errores relacionados con la alineación de los *spots* en la imagen.

La necesidad de optimizar tiempo y calidad en el flujo de procesos que actualmente realiza el CIGB en cuanto al procesamiento de imágenes; ha devenido en el diseño de una herramienta que garantice una mejora cualitativa en la cuantificación de imágenes y que resulte de fácil manejo para el investigador, cumplimentándose en los siguientes objetos:

- Procesamiento de la imagen obtenida, a través de métodos estandarizados internacionalmente como: grillado, segmentación y extracción de intensidades.
- Visualización de los resultados en términos de imágenes, de los métodos de grillado y extracción de intensidades, en el caso de este último, la superposición de los valores de intensidades obtenidos a la imagen inicial.
- Obtención de valores de intensidades del *spot* y el fondo correspondiente, y de otras medidas de interés investigativo asociadas a la imagen, que pueden ser corroboradas visualmente por el investigador.

Esta secuencia describe cómo debe funcionar el sistema, basándose en el análisis y diseño propuesto. En el próximo epígrafe se especifica la solución desde el entorno del sistema.

2.4 Propuesta de sistema

Según los métodos de procesamiento justificados con anterioridad, se describe la propuesta de sistema para el procesamiento y cuantificación de imágenes de *microarray*:

1. A partir de las imágenes obtenidas en el CIGB (Figura 6), el investigador selecciona la que desea analizar, y una vez mostrada, especifica la región de interés a procesar, puede ser la imagen completa o parte de ella (Figura 7).

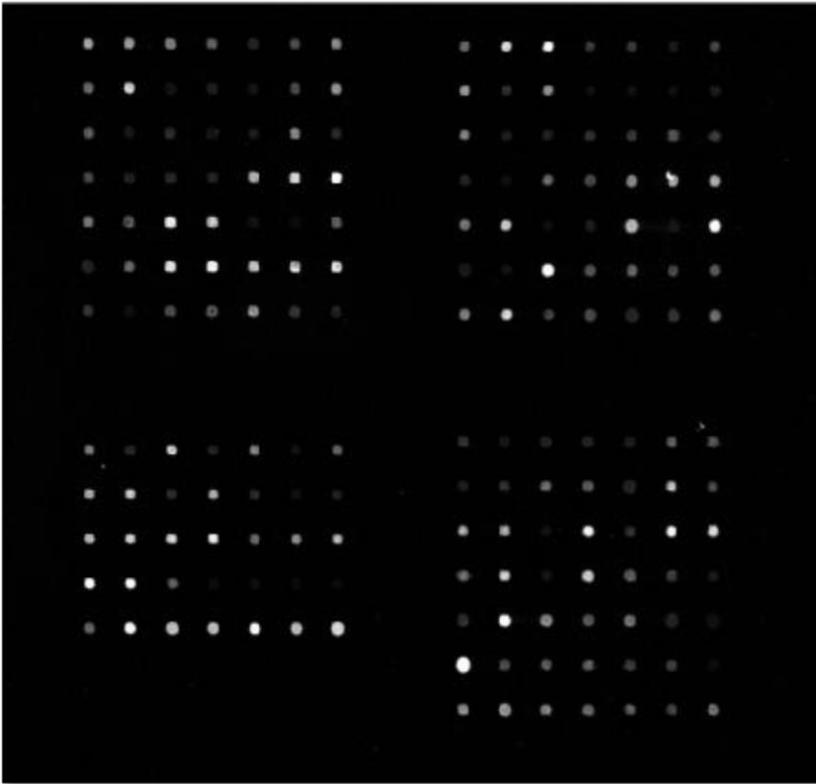


Figura 6 Imagen de *microarray* obtenida en el CIGB.

2. Para localizar los *spots* automáticamente, primero se deben encerrar en un espacio rectangular donde exista dentro un *spot* y solo un *spot*. Para ello lo primero es generar un histograma a partir del promedio de los valores de intensidad de cada columna de *píxeles*, quedando un máximo en la señal en las posiciones de las columnas de *spots* y un mínimo donde se encuentran los espacios vacíos entre estas columnas. Este histograma luego se mejora, acentuando las laderas y los picos, estos últimos se segmenta para hallar los centros de las columnas de *spots* y se toman además los valores de posición de las depresiones para localizar las divisiones entre columnas (Figura 8).

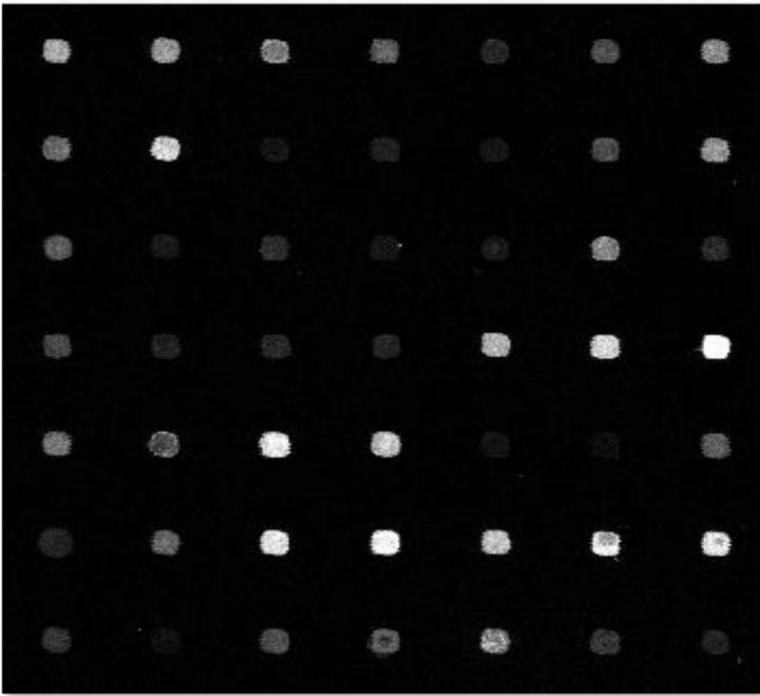
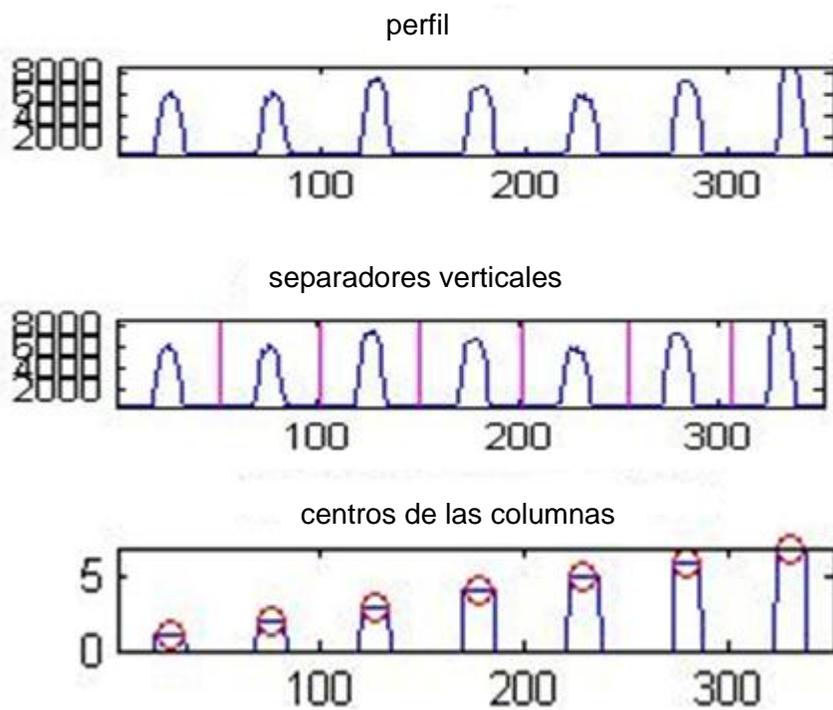


Figura 7 Región seleccionada de la imagen original.



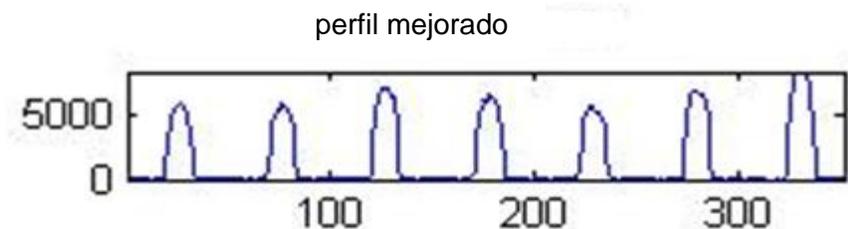


Figura 8 Representación de la secuencia a seguir para analizar el histograma correspondiente Figura 6.

Nótese la relación existente entre el histograma y la imagen en cuanto a la posición de los máximos y mínimos y las columnas de *spots*.

Una vez hecho esto se traspone la imagen y se repite el proceso para lograr determinar los espacios verticales encerrando cada *spot* de la siguiente manera (Figura 9):

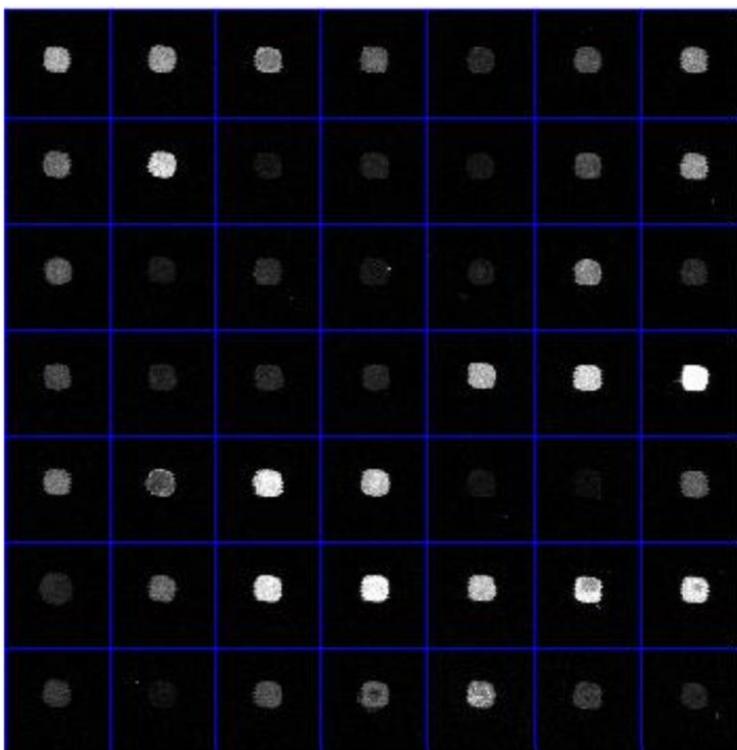


Figura 9 Imagen Grillada, resultado del primer método de procesamiento.

3. Para eliminar el ruido, se crea desde MATLAB una imagen binaria, pasándole el nivel medio de la imagen, la cual será 1 en las posiciones donde hay *spots* y 0 en las posiciones donde hay fondo. Luego los huecos de dicha imagen se rellenan usando el método de Segmentación por

umbralización de histograma, discutido en el capítulo anterior. Esto permite obviar, a la hora de calcular las intensidades, aquellos *píxeles* identificados como de fondo pues en la imagen binaria esa posición será 0, en el caso de los *píxeles* que constituyen el *spot*, se comprueban que tengan valor 1 en la binaria y se suma la intensidad del *píxel* de la imagen original en esa misma posición, dividiéndose luego entre el número de *píxeles* que constituyeron ese *spot* en específico (Figura 10).

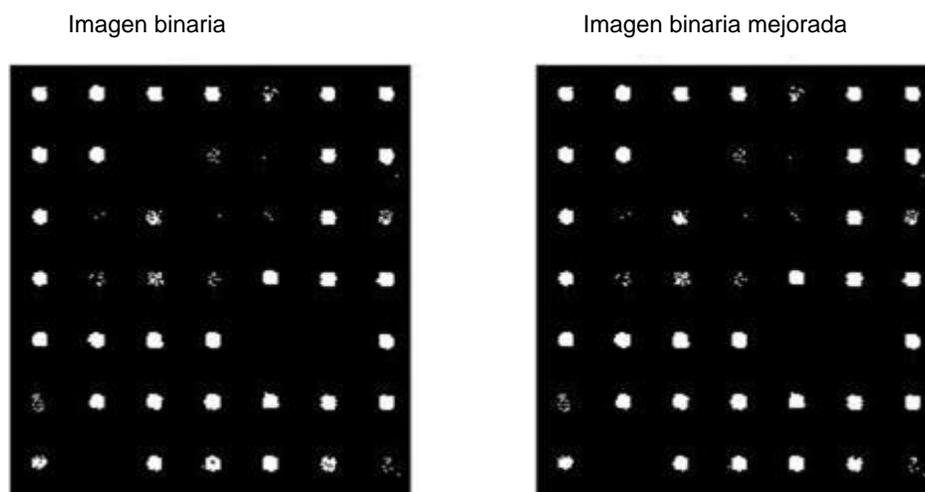


Figura 10 Nótese que se elimina ruido en alguno de los *spots* de la imagen de la derecha. Aquellos que quedan sin rellenar son demasiado tenues y serán descartados posteriormente por el software identificándolos como moléculas ausentes.

4. A partir de los separadores y la imagen binaria se pueden ejecutar ciclos para cada *spot* por separado que nos permiten saber que cantidad de *píxeles* pertenecen a cada *spot* (sin el fondo) y así saber, comparando dicha cantidad con la media del resto, si el *spot* tiene ruido o está ausente. Usando funcionalidades del MATLAB se calculan también la media de la imagen y la media del fondo, así como la mediana y desviación estándar. Estos valores son los que posteriormente serán interpretados por los investigadores según el experimento analizado.

2.5 Modelo de Dominio

En los inicios de la conceptualización del proyecto y posteriormente como resultados de los encuentros con los clientes del polo científico, se concluyó que el tipo de modelo para representar el flujo de acciones

concebidas por las cuales transcurriría todo el proceso de cuantificación de imágenes, sería a través de un modelo de dominio, fundamentado en:

- No se encontraban definidos los procesos o actividades a desarrollar, poca exactitud en la identificación del flujo de procesos para la cuantificación de imágenes que rigen el proyecto.
- No se contaba formalmente con el personal encargado de interactuar con la herramienta e interpretar los resultados obtenidos finalmente, se solapaban los roles.
- No se estaba desarrollando trabajo o estudio alguno con imágenes de *microarray*, aunque cuenta con un software *ArrayVision* las funcionalidades de este no cumplían en gran medida con la flexibilidad solicitada por los investigadores.

2.5.1 Descripción de clases

El modelo de dominio se describe a través de diagramas de clases UML. Se descompone el espacio del problema en unidades comprensibles (conceptos), que describen cuales son los términos importantes y como se relacionan entre sí (Figura 11).

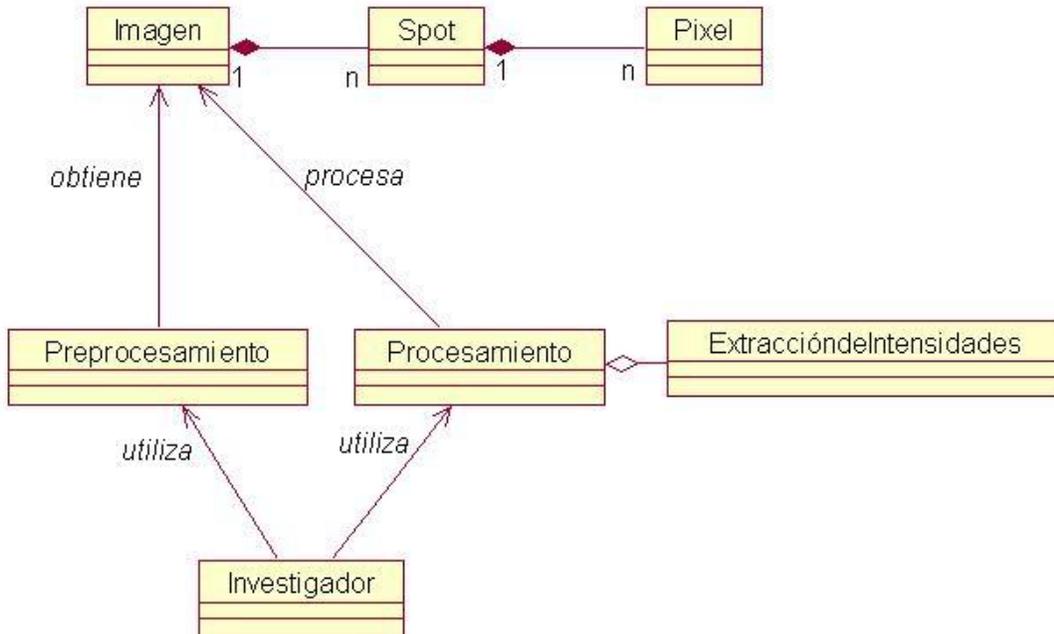


Figura 11 Diagrama que representa los conceptos o clases fundamentales identificadas en el dominio del problema.

Se describen brevemente las clases representadas, contribuyendo a esclarecer la terminología para el procesamiento y cuantificación de imágenes.

1. Investigador: es el especialista del CIGB, encargado de realizar el procesamiento y cuantificación de las imágenes de *microarray* obtenidas en el centro así como de la interpretación de sus resultados.
2. Imagen: matriz de spots que será seleccionada por el investigador, sobre la que se implementan los métodos de procesamiento.
3. *Spot*: conjunto de *píxeles* que en su totalidad equivalen a la intensidad del spot, además asociado a cada spot se tendrá un fondo.
4. Píxel: asociado a un valor de intensidad y su conjunto se hará válido tanto para la imagen, como para el spot y el fondo que conforman la imagen.
5. Preprocesamiento: conjunto de pasos iniciales para la adquisición, muestra y mejoramiento visual de la imagen, así como la selección de la región de interés que el investigador desee procesar.
6. Procesamiento: conjunto de métodos para el procesamiento de la imagen, desde la identificación de genes (Grillado y Segmentación) hasta su cuantificación (Extraer Intensidades).
7. Extracción de Intensidades: método que halla la intensidad correspondiente al spot y al fondo circundante, así como otras medidas de ayuda al investigador, particularmente la información que se genera representa el status del gen en esa imagen o muestra.

2.6 Sistema

Con el objetivo de dar solución a los problemas identificados en las entrevistas con los clientes del CIGB, se ha registrado un conjunto de requisitos que complementan las definiciones realizadas, inicialmente a nivel de dominio, y después en el diseño del software. Especificando los requerimientos funcionales y restricciones que el sistema deba cumplir.

Posterior a la lectura y muestra de la imagen, y a la selección de determinada región de interés (puede ser la imagen completa o parte) para analizar por el investigador; deviene un proceso de análisis de la misma que se divide en 3 etapas. En primer lugar se localizan los puntos a partir de los datos según la imagen seleccionada, es decir cuántos *spots* hay, cómo están agrupados, separación teórica entre los centros. A

continuación se realiza la segmentación, que consiste en identificar qué *píxeles* corresponden a un *spot* y qué *píxeles* son del fondo asociado a ese *spot*. Por último y no menos importante, se procede a la cuantificación de la imagen, es decir se extraen las intensidades relativas a los *spots* y al fondo; en este apartado es importante que el software proporcione varias medidas auxiliares como, la media, mediana y desviación de los *spots* y el fondo. Al final se obtendrá una matriz de intensidades asociada a la imagen inicial y se podrá visualizar una tabla con el conjunto de valores calculados anteriormente.

2.6.1 Requisitos Funcionales

Los requerimientos funcionales son condiciones o facilidades que el sistema debe proporcionar, ligado fundamentalmente a las actividades automatizar. Por lo que, de conjunto con el cliente se han definido como requisitos funcionales:

- R1. Cargar la imagen.
 - R1.1 Verificar el formato de la imagen a procesar.
 - R1.2 Mostrar la imagen seleccionada.
 - R1.3 Modificar visualmente la imagen, si lo desea.
- R2. Enmarcar la región de interés seleccionada.
- R3. Procesar la imagen a través de los tres métodos de procesamiento: grillar, segmentar y extraer intensidades.
 - R3.1 Estimar valores evaluados en *píxeles*.
- R4. Analizar el histograma correspondiente a la imagen
 - R4.1 Verificar el tipo de histograma.
- R5. Realizar el grillado de la imagen, es decir, obtener *grid* de la imagen.
 - R5.1 Mostrar la imagen grillada.
- R6. Realizar la segmentación de la imagen, identificar *píxeles* del *spot* y el fondo asociado.
 - R6.1 Crear una máscara de *spot*.
- R7. Realizar la segmentación por umbral, método específico de segmentación.
 - R7.1 Buscar picos en el histograma.

R7.2 Calcular valor del umbral.

R7.3 Comparar *píxeles* de la imagen con valor del umbral.

R8. Extraer valores de intensidad asociados a la imagen.

R8.1 Calcular valor de intensidad del *spot*.

R8.2 Calcula valor de intensidad del fondo, a través del método local *background*.

R8.3 Hallar medidas auxiliares correspondientes al *spot* y al fondo.

R9. Mostrar imagen cuantificada.

R9.1 Superponer valores de intensidad del *spot* a la imagen.

R10. Registrar valores calculados, anteriormente, en una tabla.

R10.1 Mostrar tabla de valores al investigador.

2.6.2 Requisitos no Funcionales

Los requerimientos no funcionales describen las restricciones del sistema o del proceso de desarrollo, este tipo de requisito al igual que el anterior se infiere por el intercambio con el cliente. De acuerdo a la aplicación de escritorio que se desarrolla, para cumplir las necesidades del usuario, y a las diversas clasificaciones de requisitos no funcionales, se han definido:

- Apariencia o interfaz externa

Se necesita una interfaz amigable, legible, interactiva, fácil de usar, profesional, clara y sencilla (acorde con la apariencia presentada, es decir con los prototipos). La aplicación de escritorio que se genera, hará uso de ventanas principales que tendrán la información suficiente y necesaria para el investigador, se visualizan las principales imágenes en correspondencia con los métodos de procesamiento que se implementan.

- Usabilidad

El sistema podrá ser usado por aquellos usuarios que posean conocimientos básicos en el procesamiento y cuantificación de imágenes, aunque en su primera versión está concebido para que sea un investigador del centro CIGB el que trabaje con la herramienta.

- Rendimiento

La herramienta será capaz de dar respuestas de forma rápida al proceso de cuantificación de la imagen. Los cálculos que se necesitan para implementar la cuantificación de los genes deben ser rápidos y eficientes, dando lugar a medidas numéricas de relevancia para el investigador.

- Soporte

Extensibilidad: El sistema debe permitir la inserción de nuevos módulos con facilidad, sin afectar el funcionamiento y prestaciones del mismo.

Mantenimiento: El sistema debe estar bien documentado de forma tal que el tiempo de mantenimiento sea mínimo.

Instalación: La instalación del sistema debe caracterizarse por su facilidad, claridad y sencillez.

Una vez terminada la aplicación se instalará en el CIGB, luego esta solución será extendida al Instituto de Medicina Tropical Pedro Kourí (IPK), que de conjunto desarrollan estudios en la detección de enfermedades con implicación genética.

- Portabilidad

El sistema podrá ser usado en los sistemas operativos de *Windows*.

- Políticos-culturales

Algún cambio que se quiera ejecutar, será a través de la dirección del proyecto del centro (CIGB) vinculados a este y será entonces tramitado con la Dirección de Producción de Software para la Salud, de la Universidad de la Ciencias Informáticas.

- Confiabilidad

Disponibilidad: El sistema deberá estar disponible para ser usado las 24 horas del día.

Tiempo medio de reparación: La reparación del sistema en caso de surgir fallas en el mismo debe realizarse en el menor tiempo posible.

- Ayuda y documentación

El sistema contará con una ayuda y manual de usuario, donde esté presente la documentación básica que posibilite comprender el funcionamiento de la herramienta y las características generales a tener en cuenta para garantizar el uso eficiente del mismo.

- Software

Windows 2000 o superior.

Framework 2.0 de Visual Studio y *MCRInstaller*⁷ de MATLAB deben de instalarse en la máquina donde se vaya a utilizar la aplicación.

- Hardware

PC compatible con un microprocesador de más de 1.6 GHz.

Pantalla con una resolución de al menos 800 x 600.

Como mínimo 256MB + RAM.

- Restricciones en el diseño y la implementación

Lenguaje de programación a ser usado para la implementación: C#.

Herramientas de desarrollo: MATLAB, Visual Studio 2005.

Herramienta CASE: Rational Enterprise Edition 2003.

2.6.3 Actores identificados. Diagrama de Casos de Uso

El actor es una persona o sistema externo que desempeña un rol determinado al interactuar o relacionarse con el sistema. En el caso particular, se identifica como actor, al investigador del centro científico mencionado, enfocado al trabajo con imágenes de *microarray* (Tabla 2).

Tabla 2 Breve descripción del actor del sistema identificado.

Actor	Breve Descripción
Investigador	Especialista del CIGB, encargado de seleccionar la imagen a procesar e interpretar los resultados alcanzados, posteriores al procesamiento y cuantificación de la imagen de <i>microarray</i> elegida.

Los casos de uso identificados y desarrollados a continuación, completan el diagrama de casos de uso del sistema (Figura 12).

Casos de uso identificados.

1. CU Cargar Imagen.

⁷ Archivo autoextraíble que contiene las librerías de MATLAB.

2. CU Seleccionar Región de Interés.
3. CU Procesar Imagen.
4. CU Realizar Grillado.
5. CU Analizar Histograma.
6. CU Realizar Segmentación.
7. CU Segmentar por Umbralización.
8. CU Extraer Intensidades.
9. CU Mostrar Imagen Cuantificada.
10. CU Tabular Valores.

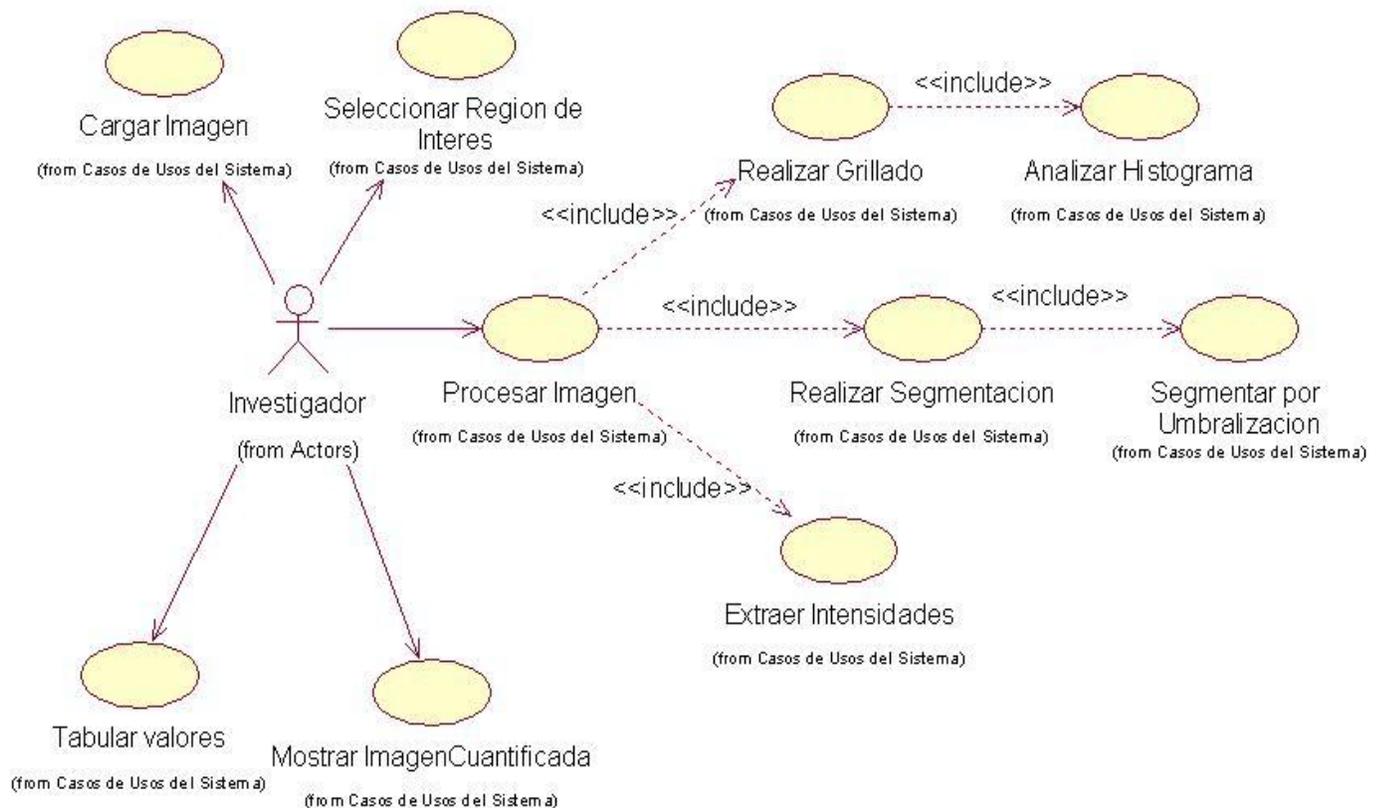


Figura 12 Diagrama de casos de uso del sistema.

2.6.4 Descripción textual de Casos de Uso

La descripción de casos de uso a continuación expuesta tiene como objetivo, esbozar el flujo de información y especificidades de cada caso de uso de sistema, la realización de los mismos es de gran

importancia para el desarrollo en el flujo de trabajo siguiente, específicamente en la construcción de los diagramas de interacción.

Tabla 3 CU-Cargar Imagen.

Caso de uso	
CU- 1	Cargar imagen
Propósito	Permitir al investigador la entrada al sistema de la imagen para cuantificar.
Actores: Investigador (Inicia)	
Resumen: El caso de uso se inicia cuando el investigador selecciona en la pestaña Imagen, la opción Abrir Imagen y busca la que desea analizar, mostrándose la misma en el monitor para su posterior análisis, antes de mostrarla se verifica que el formato de la imagen, sea correcto, por último el investigador puede adecuar visualmente la imagen a su gusto.	
Referencias	R1, R1.1, R1.2, R1.3.
Precondiciones	
Poscondiciones	Se muestra imagen que se va a procesar.
Acción del actor	Respuesta del sistema
1-Selecciona imagen que desea analizar, tras haber tomado la opción Abrir Imagen, de la pestaña Imagen.	1.1-El sistema verifica que el archivo de la imagen seleccionada tiene como formato JPG o TIFF.
	1.2-El sistema muestra imagen seleccionada, en caso de cumplir con el formato.
2-El investigador mejora calidad visual de la imagen, mejorando brillo y/o contraste, según desee.	2.1- El sistema aumenta o disminuye brillo y/o aumenta o disminuye contraste de la imagen.
	2.2- El sistema muestra imagen con mejora visual para posterior procesamiento.
Flujo alternativo del 1.1	
	1.1- Verifica que el archivo de la imagen no es correcto.
	1.2-Muestra mensaje de error «Por favor entre

	correctamente la imagen que desea analizar, cerciórese del formato». Y regresa para que el investigador seleccione otra imagen, paso 1.
Prioridad	Crítico

Tabla 4 CU-Seleccionar Región de Interés.

Caso de uso	
CU-2	Seleccionar Región de Interés
Propósito	Permitir al investigador seleccionar el grupo de <i>spots</i> de la imagen que desea analizar(esta selección puede ser la imagen completa)
Actores: Investigador (Inicia)	
Resumen: El caso de uso se inicia cuando el investigador selecciona de la imagen mostrada la región de <i>spots</i> de interés para procesar.	
Referencias	R2.
Precondiciones	El usuario debe visualizar la imagen seleccionada para procesamiento.
Poscondiciones	Se tienen las coordenadas de la región(s) seleccionada, que bien pueden ser de la imagen completa o grupo de <i>spots</i> .
Acción del actor	Respuesta del sistema
1-Accede a la opción Seleccionar Región de Interés (pestaña Procesamiento), que facilita marcar región de <i>spots</i> y desmarcar la selección.	1.1- El sistema visualiza región enmarcada por el investigador en cuadro de color.
	1.2- El sistema permite desmarcar las región de <i>spots</i> no deseada de la elección anterior, visualizando el nuevo cambio dentro de la imagen y regresa al paso 1.1
	1.3-El sistema recoge coordenadas de la región(s) seleccionada.
Prioridad	Secundario

Tabla 5 CU-Procesar Imagen.

Caso de uso	
CU- 3	Procesar Imagen
Propósito	Permitir al investigador acceder a los métodos necesarios para procesar y cuantificar las regiones de <i>spots</i> señaladas en la imagen.
Actores: Investigador (Inicia)	
Resumen: El caso de uso se inicia cuando el investigador selecciona la opción Procesar Imagen, ahí se encuentra los métodos por los cuales transita el procesamiento de imágenes, el cual se puede dividir en 3 etapas: grillado, segmentación y extraer intensidades, en este caso de uso se hallarán valores que serán la base para cálculos de otros métodos.	
Referencias	R3, R3.1. Caso de Uso Realizar Grillado (inclusión). Caso de Uso Realizar Segmentación (inclusión). Caso de Uso Extraer Intensidades (inclusión).
Precondiciones	Cuenta con las coordenadas de la región de <i>spots</i> seleccionada en la imagen, elegida inicialmente.
Poscondiciones	Se tiene los valores asociados al <i>spot</i> (tamaño, espacio), se localizan los <i>spots</i> (grillado y segmentación) y se resumen intensidades (extraer intensidades).
Acción del actor	Respuesta del sistema
1-El investigador selecciona opción Procesar Imagen, de la pestaña Procesamiento.	1.1-El sistema estima valores, evaluado en <i>píxeles</i> , dimensión de parámetros como: <ul style="list-style-type: none"> • Tamaño del <i>spot</i> (<i>spot size</i>) • Espacio entre <i>spot</i> adyacentes (<i>spot spacing</i>) • Distancia entre las regiones de <i>spots</i> (pin x y pin y).
	1.2- El sistema localiza <i>spots</i> de la imagen Ver Caso de Uso Realizar Grillado.
	1.3-El sistema identifica qué <i>píxeles</i> forman parte del <i>spot</i> y cuál del fondo Ver Caso de Uso Realizar

	Segmentación.
	1.4-El sistema calcula intensidades correspondientes a los <i>spots</i> de la imagen, además de otras medidas numéricas Ver Caso de Uso Extraer Intensidades.
Prioridad	Crítico

Tabla 6 CU-Realizar Grillado.

Caso de uso	
CU-5	Realizar Grillado
Propósito	Asignar coordenadas a cada uno de los <i>spots</i> , y visualizar la <i>grid</i> creada, en correspondencia con la imagen.
Resumen: Posterior a la selección de procesamiento de la imagen, el primer método en ejecutarse será asignar coordenadas a cada uno de los <i>spots</i> , localizar los puntos y asignarle a cada uno coordenadas, dicho caso de uso obtiene del CU-Analizar Histograma los elementos para visualizar la <i>grid</i> de la imagen, como resultado.	
Referencias	R5, R5.1, R4, R4.1. CU Analizar Histograma (inclusión).
Precondiciones	Cuenta con las coordenadas de la región seleccionada.
Poscondiciones	Se muestra la imagen con la <i>grid</i> correspondiente.
Acción del actor	Respuesta del sistema
	1.1-El sistema estima las coordenadas de cada uno de los <i>spots</i> de la región seleccionada, a partir de las coordenadas de la región de interés elegida anteriormente.
	1.2-El sistema accede a Ver Caso de Uso Analizar Histograma, para poder encerrar cada <i>spot</i> en un espacio rectangular.
	1.3-El sistema muestra al investigador la imagen con la

	<i>grid</i> asociada.
Prioridad	Crítico

Tabla 7 CU-Analizar Histograma.

Caso de uso	
CU-4	Analizar Histograma
Propósito	Facilitar trabajo del investigador referente a un conjunto de pasos para la reducción de ruido y creación de la <i>grid</i> , de la imagen.
Resumen: Dado que, el histograma representa la distribución de probabilidad de que un <i>píxel</i> tenga un nivel de gris; este caso de uso debe realizar un conjunto de pasos (crear histograma, segmentar picos, localizar centros, divisiones las columnas, etc.) para obtener finalmente los elementos necesarios que necesita para el posterior trabajo.	
Referencias	R4, R4.1.
Precondiciones	Tiene coordenadas de cada uno de los <i>spots</i> de la región de interés seleccionada.
Poscondiciones	Se conoce el tipo de histograma y los elementos para mostrar la <i>grid</i> de la imagen.
Acción del actor	Respuesta del sistema
	1.1-El sistema crea el histograma de la señal correspondiente a la imagen, se tiene en cuenta coordenadas adquiridas anteriormente en la selección de <i>spots</i> .
	1.2-El sistema estima espacio entre los picos del histograma de la señal.
	1.3-El sistema decide si el histograma es multimodal o bimodal; si los <i>spots</i> presentes en la imagen tienen, aproximadamente, el mismo número de <i>píxeles</i> que el fondo y entre ellos tienen niveles de gris próximos, el

	histograma presenta un aspecto bimodal.
	1.4-El sistema estima las posiciones de las columnas del histograma.
	1.5-El sistema extrae los centros de las columnas, que corresponden con los centros horizontales de los <i>spots</i> .
	1.6-El sistema determina las divisiones entre los <i>spots</i> , básicamente los puntos medianos entre los picos adyacentes del histograma proporcionan localizaciones del <i>spot</i> en la <i>grid</i> .
	1.7-Se traspone la imagen y se hace desde el paso 1.1 (el mismo proceso pero vertical), y se tiene en cuenta los espacios verticales y horizontales.
Flujo Alterno	
	1.3-El sistema decide que el histograma presenta aspecto multimodal, si los <i>spots</i> presentes en la imagen no tienen, aproximadamente, el mismo número de <i>píxeles</i> que el fondo y tienen diferentes niveles de gris entre ellos y regresa al paso 1.4.
Prioridad	Crítico

Tabla 8 CU-Realizar Segmentación.

Caso de uso	
CU-6	Realizar Segmentación
Propósito	Clasificar los <i>píxeles</i> en <i>foreground</i> o <i>background</i> .
Resumen: Posterior a la selección de procesamiento de la imagen, y después de haber realizado el grillado de la imagen, se ejecutan procedimientos para discernir entre los <i>píxeles</i> que corresponden al <i>spot</i> y los que corresponden al fondo, se crea una máscara de <i>spot</i> que agrupa el conjunto de <i>píxeles</i> donde se encuentra el <i>spot</i> .	

Referencias	R6, R6.1, R7. CU Segmentar por umbralización (inclusión).	
Precondiciones	Muestra la grid correspondiente a la imagen.	
Poscondiciones	Se tiene una máscara de <i>spots</i> .	
Acción del actor	Respuesta del sistema	
	1.1-El sistema implementa un método para separar los <i>spots</i> del fondo Ver Caso de Uso Segmentar por Umbralización.	
	1.2-El sistema obtiene los <i>píxeles</i> asociadas a un nivel de gris en el histograma.	
	1.3-El sistema produce una máscara de <i>spot</i> , esto se repite para todos los <i>spots</i> que conforman la imagen.	
Prioridad	Crítico	

Tabla 9 CU- Segmentar por Umbralización.

Caso de uso	
CU-7	Segmentar por umbralización
Propósito	Obtener imagen umbralizada, es decir, aquella tiene discernidos que <i>píxeles</i> son del <i>spot</i> y cuales pertenecen al fondo.
Resumen: Se implementa un algoritmo de umbralización, para obtener un umbral de brillo que se compara con cada uno de los <i>spots</i> , como resultado de dicha comparación determina que <i>píxel</i> es parte del <i>spot</i> y cual del fondo, para obtener un umbral correcto se usa el método de segmentación por umbralización de histograma., es necesario tener en cuenta el tipo de histograma del CU-Analizar Histograma.	
Referencias	R7, R7.1, R7.2, R7.3, R4.
Precondiciones	Tiene el tipo de histograma (multimodal o bimodal).

Poscondiciones	Se tienen los <i>píxeles</i> asociados al <i>spot</i> y los correspondientes al fondo, es decir la imagen umbralizada.	
Acción del actor	Respuesta del sistema	
	1.1-El sistema realiza una búsqueda de picos en el histograma, en función de tipo de histograma (multimodal o bimodal).	
	1.2-El sistema, en caso de que sea bimodal, considera que el umbral este en el nivel de gris del histograma que tiene un valor mínimo entre los dos picos.	
	1.3- El sistema localiza el mayor máximo local del histograma (corresponderá a un máximo de probabilidad asociado a un <i>spot</i>).	
	1.4- Se localiza el segundo mayor máximo local del histograma (máximo de probabilidad asociado a otro <i>spot</i> o al fondo de la imagen).	
	1.5- Localiza el mínimo local entre ambos máximos y se utiliza su valor como valor para el umbral.	
	1.6-El sistema dado el umbral, halla imagen umbralizada, para esto recorre todos los <i>píxeles</i> de la imagen; y si su nivel de gris es mayor que el umbral es parte del <i>spot</i> , si es menor es parte del fondo.	
Flujo Alterno		
	1.2- En caso de que sea multimodal, considera un mínimo entre cada pareja de máximos, y regresa al paso 1.3.	
Prioridad	Crítico	

Tabla 10 CU-Extraer Intensidad.

Caso de uso	
CU-8	Extraer Intensidad
Propósito	El investigador puede conocer el valor de las intensidades asociadas a los <i>spots</i> y al fondo, y de esta forma evaluar la expresión de los genes en la imagen.
Resumen: Al finalizar todo el proceso de grillado y segmentación; permite calcular el valor de expresión del gen, valor de intensidades tanto de fondo como del <i>spot</i> .	
Referencias	R8, R8.1, R8.2, R8.3.
Precondiciones	Se tiene los <i>píxeles</i> del fondo y del <i>spot</i> , así como la <i>spot mask</i> .
Poscondiciones	Valores numéricos correspondientes al fondo y al <i>spot</i> , según los <i>píxeles</i> correspondientes.
Acción que inicia	Respuesta del sistema
	1.1- El sistema mide intensidades del primer <i>spot</i> y cuantifica valor de la expresión del gen. La técnica de la medida se puede entonces repetir para todos los <i>spots</i> .
	1.2-El sistema calcula intensidad del <i>spot</i> como la media de las intensidades de los <i>píxeles</i> dentro del <i>spot mask</i> .
	1.3- El sistema calcula los valores de la intensidad del fondo circundante al <i>spot</i> (<i>local background</i> , que es la mediana de los valores de los <i>píxeles</i> de la región alrededor del <i>spot</i>).
	1.4-El sistema quita el fondo de todos los <i>spots</i> , calcula otra vez y compara las medidas (la intensidad del fondo alrededor del <i>spot</i> no es cero).

	<p>1.5-El sistema halla valores auxiliares (a partir de los <i>píxeles</i> que representan los <i>spots</i> (señal) y su fondo):</p> <ul style="list-style-type: none"> • Media del fondo • Media del <i>spot</i> • Mediana del <i>spot</i> • Desviación estándar del <i>spot</i> • Desviación estándar del fondo
Prioridad	Crítico

Tabla 11 CU-Mostrar Imagen Cuantificada.

Caso de uso	
CU-10	Mostrar Imagen Cuantificada
Propósito	Sobreponer niveles cuantitativos de la expresión sobre la imagen (después de proceso de selección de <i>spots</i>) para proporcionar el aseguramiento visual.
Resumen: Al finalizar el cálculo de valores de intensidades del <i>spot</i> y el <i>background</i> , se muestra la imagen con los niveles de intensidad de los <i>spots</i> .	
Referencias	R9, R9.1, R8, R8.1.
Precondiciones	Tener los valores de intensidad, calculados, correspondientes al <i>spot</i> .
Poscondiciones	Muestra al investigador la imagen con los valores de intensidad asociados a cada <i>spot</i> .
Acción del actor	Respuesta del sistema
1-El investigador selecciona de la pestaña Datos la opción Mostrar Imagen con valores,	1.1-El sistema registra valores de intensidad de cada <i>spot</i> calculados anteriormente.
	1.2- El sistema superpone los valores de intensidad de los <i>spots</i> a la imagen seleccionada y procesada desde el principio.
	1.3-El sistema muestra la imagen con valores de intensidad, asociados a cada <i>spot</i> , para aseguramiento adicional del investigador.

Prioridad	Secundario
-----------	------------

Tabla 12 CU-Tabular Valores.

Caso de uso	
CU-11	Tabular Valores
Propósito	Expresar valores de intensidades y otros cálculos por medio de una tabla, facilitando trabajo del investigador.
Actores: Investigador(inicia)	
Resumen: Muestra una tabla de valores que recoge las intensidades de cada <i>spot</i> y, las otras medidas asociadas tanto al <i>spot</i> como al fondo (media, mediana, desviación).	
Referencias	R10, R10.1, R8.
Precondiciones	Acceder a medidas auxiliares calculadas para el <i>spot</i> y el fondo (media, mediana y desviación).
Poscondiciones	Muestra tabla de valores asociada a intensidades del <i>spot</i> y otras medidas anteriormente calculadas.
Acción del actor	Respuesta del sistema
1-El investigador selecciona la pestaña Datos, y en esta la opción Mostrar Tabla de Valores.	1.1-El sistema registra los valores de interés para el investigador, y los introduce en una tabla valores
	1.2-El sistema muestra al investigador tabla de valores correspondiente a la imagen.
Prioridad	Crítico

2.7 Conclusiones

En este capítulo se hizo un estudio de las principales actividades que manejan el procesamiento de imágenes de *microarray* en el CIGB, las cuales se representan en el modelo de dominio descrito. Se especifican los procesos que se automatizarán, para procesar y cuantificar la imagen. Se fundamenta la

propuesta de diseño de una aplicación que mejore la ejecución de dichas actividades, derivando como resultado las actividades a automatizar y el flujo de información en cada una de ellas, se define además el actor que interactúa con el sistema. Para la realización de lo mencionado anteriormente se tuvo en cuenta las funcionalidades que debe tener el software y la descripción de cada uno de los casos de uso identificados.

CAPÍTULO 3 ANÁLISIS Y DISEÑO DEL SISTEMA

3.1 Introducción

En la realización de este capítulo el punto de partida lo constituyó, las especificaciones de los casos de uso del sistema y los requisitos asociados a estos, tanto para el análisis como para el diseño; considerando que los artefactos desarrollados en el análisis constituyen la entrada esencial para los que se construyen en el diseño. Se representan la realización de casos de uso de ambos flujos de trabajo. Se describen los patrones a utilizar en el diseño y el estilo arquitectónico propuesto. Según el ciclo de vida de RUP, el flujo de trabajo de análisis y diseño se afianza en la fase de elaboración (Figura 13).

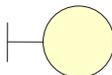
La propuesta esbozada se corresponde a las necesidades actuales del CIGB en cuanto al procesamiento de imágenes, sin dejar de ser, lo suficientemente general para adaptarse a requerimientos futuros.



Figura 13 Representación del flujo de trabajo de Análisis y Diseño durante el ciclo de RUP.

3.2 Clases del análisis

Resulta de gran ayuda para la realización de estas clases los conceptos o definiciones identificados en el dominio del problema, así como la comprensión de los requisitos, para trazar una estructura basada en clases del análisis, sin llegar a precisar cómo implementar la solución.

- Clases Interfaz 

Estas clases proveen funcionalidades básicas para interactuar con el sistema, muestran los resultados de los métodos de Grillado y Extracción de Intensidad en términos de imágenes e información numérica de valor para el investigador; el nombre de estas clases comienza con las siglas CI, en este caso se encuentran CI_AnalizarImagen, CI_MostrarImagenCuantificada y CI_Tabular Valores.

- Clases Controladora 

Clases encargadas de realizar todos los procesos de búsqueda, selección, grillado y segmentación de la imagen a procesar, así como, análisis del histograma correspondiente; cálculo del umbral y de valores de intensidad y otras medidas; el nombre de estas clases comienza con las siglas CC, en este caso se encuentran CC_GestionarImagen, CC_AnalizarHistograma, CC_ExtraerIntensidad, por mencionar algunas.

- Clases Entidad 

Principales clases que contiene la información necesaria de una imagen, el histograma correspondiente, el umbral para desarrollar el método de Segmentación, además de los datos de los *píxeles* asociados al *spot* y a su correspondiente fondo, así como la tabla de valores relativos a la imagen. Este tipo de clase se comienza con las siglas CE. Por la información contenida se menciona, algunas como: CE_histograma, CE_imagen, CE_tablaimagen.

3.3 Realización de Casos de Uso del análisis

Inicialmente una realización de caso de uso del análisis tiene una descripción textual del flujo de sucesos, diagramas de clases que muestran sus clases del análisis y diagramas de interacción que muestran la realización de un flujo particular del caso de uso en términos de clases y de sus objetos en interacción.

- Realización de CU Cargar_Imagen (Anexo 1).
- Realización de CU Seleccionar_Región_Interés (Anexo 2).
- Realización de CU Procesar_Imagen (Anexo 3).
- Realización de CU Realizar_Grillado (Anexo 4).
- Realización de CU Realizar_Segmentación (Anexo 5).
- Realización de CU Extraer_Intensidad (Anexo 6).
- Realización de CU Tabular_Valores (Anexo 7).

3.4 Diseño

Para el desarrollo de este flujo de trabajo, se tiene como premisa fundamental la modelación del sistema, estructurando la información de forma tal, que soporte los requerimientos de mantenimiento, reusabilidad, escalabilidad y robustez.

Primeramente se identifican los objetos pertinentes, clasificados dentro de las clases en la granularidad correcta, se definen interfaces y establecen relaciones entre ellos. Luego se organiza la arquitectura del sistema de manera que contemple las formas de acceder y procesar las imágenes de *microarray*.

3.4.1 Estilo arquitectónico

Según la recomendación de la IEEE-1471 (24) la arquitectura de software se entenderá como «la organización fundamental de un sistema encarnada en sus componentes, las relaciones de los componentes con cada uno de los otros y con el entorno, y los principios que orientan su diseño y evolución».

Dentro de la amplia clasificación de estilos arquitectónicos utilizaremos arquitectura en capas. Es relativamente sencillo reconocer por qué interesa una disposición en capas.

Las ventajas del estilo en capas son obvias. Primero que nada, el estilo soporta un diseño basado en niveles de abstracción crecientes, lo cual a su vez permite a los implementadores la partición de un problema complejo en una secuencia de pasos incrementales. En segundo lugar, el estilo admite muy naturalmente optimizaciones y refinamientos. En tercer lugar, proporciona amplia reutilización. Al igual que los tipos de datos abstractos, se pueden utilizar diferentes implementaciones o versiones de una misma capa en la medida que soporten las mismas interfaces de cara a las capas adyacentes. Esto conduce a la

posibilidad de definir interfaces de capa estándar, a partir de las cuales se pueden construir extensiones o prestaciones específicas.

Con este antecedente se propone una arquitectura en tres capas (Figura 14), donde cada una se ocupa de un nivel del problema y debe tener poco acoplamiento con las demás de manera que el cambio en una capa, no altera en gran medida los cambios en la otra capa. Si se desea en un futuro incorporar una capa de presentación distinta, no debe alterar a la Capa Operacional ya desarrollada. En definitiva, cambiar la implementación de una capa, debe introducir los mínimos efectos en el resto de la aplicación.

- Capa Presentación: desarrolla todos los elementos que conforman la interfaz de la aplicación, facilitando el desarrollo del sistema y visualizando el resultado del procesamiento de imágenes de distintos modos, utilizando los componentes que provee el .NET *Framework* 2.0; estos tienen correspondencia con funcionalidades de adquisición y/o visualización de imagen y datos.
- Capa Operacional: encargada de manejar todos los datos del sistema y organizar la lógica del negocio, comprende todas las clases encargadas de realizar los algoritmos y métodos necesarios para la cuantificación de imágenes, denominadas clases controladoras; aparecen además las clases entidades, que definen algún tipo de información perdurable en el marco del procesamiento de dichas imágenes, esto garantiza el flujo de información de una capa a otra.
- Capa de Soporte: de manera general esta es una capa utilitaria, o sea incluye todo lo referente a librerías, componentes, *framework*, que den soporte a la Capa Operacional.

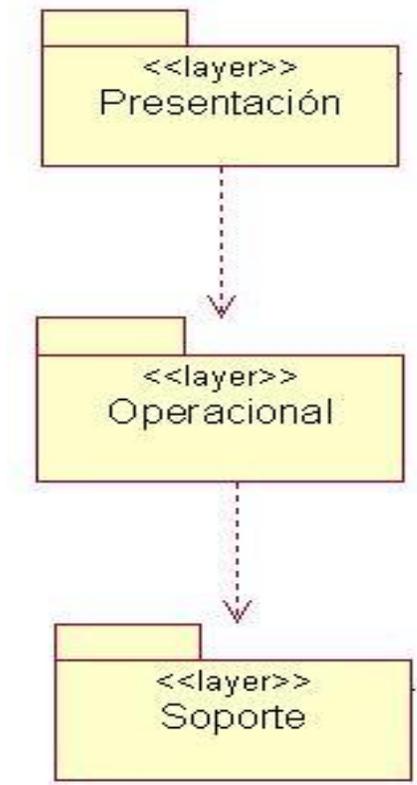


Figura 14 Arquitectura en 3 capas para el procesamiento de imágenes de *microarray*.

3.4.2 Subsistemas e interfaces

Los subsistemas deben identificarse por los servicios que proporcionan, definiendo una forma coherente de examinar un aspecto del problema, tal como calcular intensidad o hallar imagen umbralizada. De los subsistemas descritos que poseen una interfaz, esta se encuentra bien definida con el resto del sistema.

Ajustándose a esta concepción el problema general se divide, según las particularidades del procesamiento y cuantificación de imágenes, en varios subsistemas lo que posibilita el desarrollo ágil de cada uno de ellos independientemente. Basado en este criterio quedan definidos cinco subsistemas relacionados con interfaces y paquetes de complejidad mínima. Esta representación se visualiza en (Figura 15).

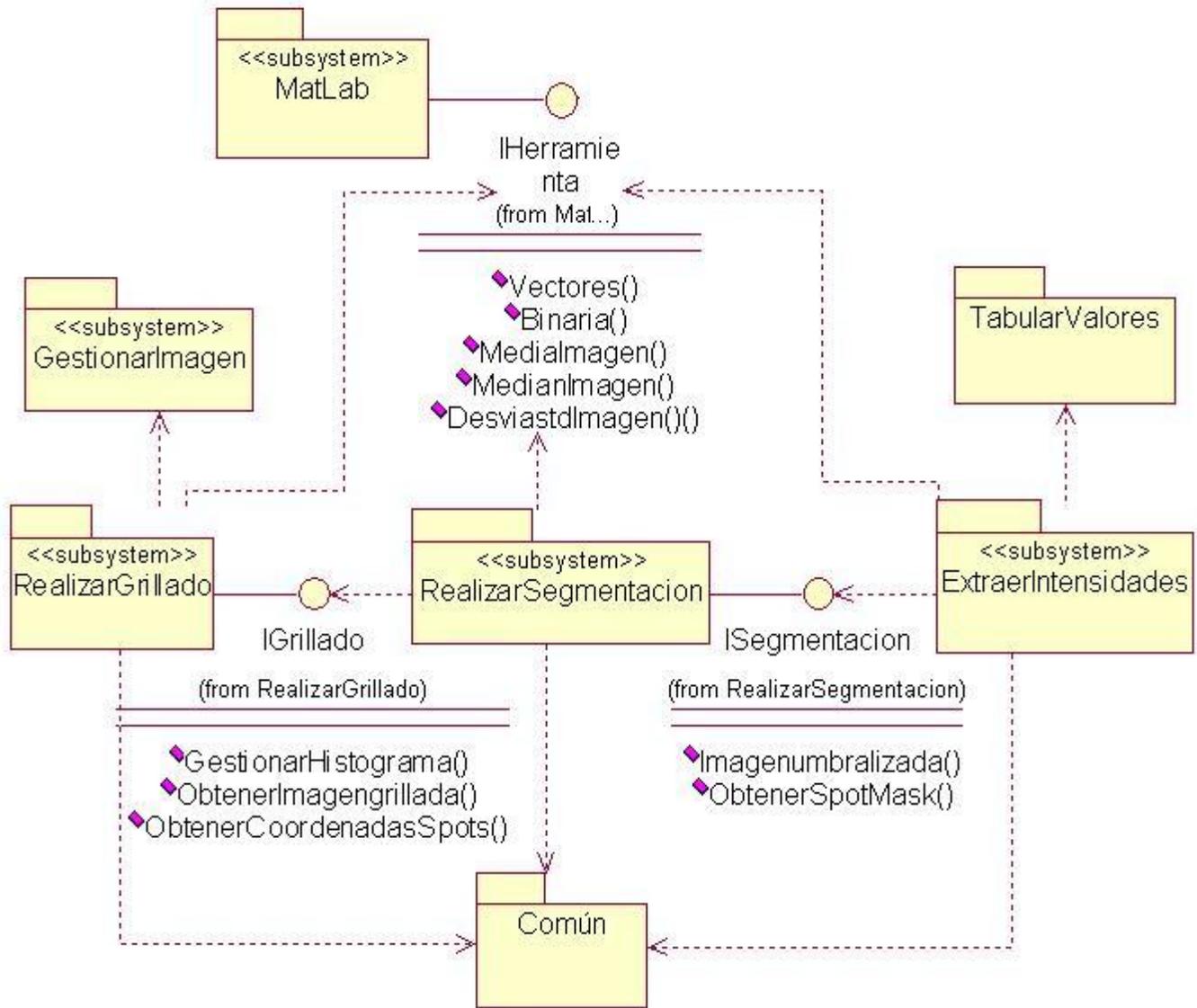


Figura 15 Representación de subsistemas e interfaces.

Los subsistemas RealizarGrillado, RealizarSegmentación y ExtraerIntensidades, acceden a las funcionalidades que genera la interfaz IHerramienta en términos de operaciones que se implementan dentro del subsistema MATLAB. En dependencia del subsistema será la correspondencia con las operaciones que implementa la interfaz IHerramienta (que utiliza funciones incorporadas en MATLAB), ejemplo:

- Subsistema RealizarGrillado accede a la operación *Vectores()*, que contiene los elementos para grillar la imagen, y analizar el histograma. Este subsistema incluye las clases necesarias para que la interfaz IGrillado genere las operaciones *ObtenerCoordenadasSpot()*, *ObtenerImagenGrillada()*, y *GestionarHistograma()*. Necesita funcionalidades y resultados de los casos de uso Cargar imagen y Seleccionar región de interés. De manera general este subsistema inicializa el primer método para procesar la imagen, estimando valores de los *spots* de la imagen y después realizando el grillado de la misma.
- Subsistema RealizarSegmentación tiene clases asociadas a la obtención de los *píxeles* del fondo y del *spot* relativo a la imagen, accede a las interfaces de programación IGrillado e IHerramienta (específicamente a la operación *Binaria()*), pues tienen la información necesaria que el subsistema mencionado necesita para su realización. Asociada al subsistema está la interfaz ISegmentación que implementa las operaciones *ImagenUmbralizada()* y *ObtenerSpotMask()*, estas son utilizadas por el subsistema anterior. El subsistema RealizarSegmentación es el segundo método de procesamiento en ejecutarse. De manera general, en este se distinguen los *píxeles* del *spot* de los del fondo de la imagen.
- Subsistema ExtraerIntensidades accede a las operaciones de la interfaz ISegmentación, relacionado además con el paquete TabularValores. Contiene las clases para calcular la intensidad del *spot* y del fondo correspondiente, así como otras medidas de valor, para esto es necesario acceder a los 3 últimos métodos de la interfaz IHerramienta. De manera general dicho subsistema conforma el método más importante, aunque no es posible realizarse sin los anteriores, por los resultados que se obtienen en cuanto a intensidades y la importancia de estos valores en la interpretación que realice el investigador.

El subsistema GestionarImagen recoge los casos de uso Cargar Imagen y Seleccionar Región de Interés, sus resultados son necesarios para realizar el procesamiento de imágenes de *microarray*, permitiéndonos seleccionar determinada imagen y una vez seleccionada, enmarcar la región a procesar en esta.

Se define el paquete TabularValores con el objetivo de separar funcionalidades por cuestiones de reutilización, considerando el mismo como un subproceso que agrupa las clases necesarias para registrar en una tabla los valores de intensidad obtenidos en el subsistema ExtraerIntensidades.

Existe un paquete exclusivo llamado Común donde se han agrupado todas las clases que son compartidas por los demás subsistemas, lo que facilita la reutilización de estas y proporcionan al diseño una alta cohesión.

3.4.3 Clases del diseño

Adhiriéndose a la línea definida por el marco arquitectónico y a los principios del buen diseño, se identificaron un conjunto de clases, procurando que la distribución de estas clases dentro de los subsistemas y paquete representados (Figura 15), brinde una solución óptima al problema.

- Interfaces o vistas.

Estas clases contienen los componentes necesarios para construir las vistas que se utilizan para la entrada de la imagen y para mostrar los resultados, del procesamiento y cuantificación, en términos de imágenes o de valores de intensidades asociadas a esta. Los atributos de estas clases son públicos de manera tal que otras clases puedan acceder a ellos en cualquier momento.

- Gestores

Estas clases contienen la lógica del negocio y manejan la mayoría de los algoritmos y operaciones que se realizan en el sistema para procesar y cuantificar la imagen. Se diseño al menos un gestor para cada uno de los procesos y subprocesos que se ejecutan en cada subsistema, a partir de la diversidad de estos.

- Entidades

Estas clases contienen la información necesaria del sistema en cuestión, la mayoría realizan operaciones simples sobre los datos que poseen. En ocasiones se complementan dando cierto nivel de complejidad en el subsistema correspondiente.

3.4.4 Realizaciones de Casos de Uso del diseño

Se refieren a continuación las realizaciones de casos de uso más importantes para la arquitectura del sistema, sobre la base de que, estas realizaciones establecen una relación en la que se describe como se realiza un caso de uso específico y como se implementa(o ejecuta) en términos de clases de diseño y sus objetos.

- Realización de CU CargarImagen y Realización de CU SeleccionarROI (Anexo 8).

- Realización de CU RealizarGrillado (Anexo 9).
- Realización de CU RealizarSegmentación (Anexo 10).
- Realización de CU ExtraerIntensidades (Anexo 11).
- Realización de CU TabularValores (Anexo 12).

3.4.5 Patrones de diseño: Singleton

Los patrones de diseño proponen una forma reutilizar la experiencia de los desarrolladores, Christopher Alexander considera que «Cada patrón describe un problema que ocurre una y otra vez en nuestro entorno, para describir después el núcleo de la solución a ese problema, de tal manera que esa solución pueda ser usada más de un millón de veces sin hacerlo ni siquiera dos veces de la misma forma» (25).

El Singleton es quizás el más sencillo de los patrones, también uno de los más conocidos y utilizados. Su propósito es asegurar que sólo exista una instancia de una clase y proporciona un punto de acceso global a esta instancia.

A continuación se ejemplifica este patrón en la clase imagen (Figura 16), pues todo el flujo que se desarrolla para procesar y cuantificar la imagen de *microarray*, será propio de una imagen, es decir no se modificará o analizará más de una imagen a la vez, aunque esta clase cuenta con otras funcionalidades que por el momento no compete su explicación.

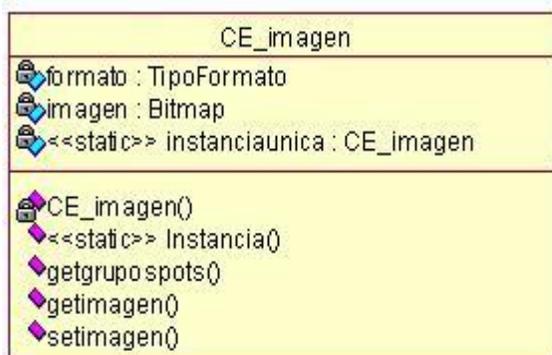


Figura 16 Representación del patrón «Singleton» en la clase imagen.

El funcionamiento de este patrón es muy sencillo y se reduce a los siguientes conceptos:

1. Ocultar el constructor de la clase imagen, para que las otras clases que necesiten después de esta, no puedan crear instancias.

2. Declarar en la clase imagen una variable miembro privada que contenga la referencia a la instancia única que se gestiona.
3. Proveer en la clase imagen una función o propiedad que brinde acceso a la única instancia gestionada por esta clase, además. Las clases que necesitan acceder a la instancia será a través de esta función o propiedad.

3.5 Conclusiones

Como resultado de este capítulo se construyó un modelo de análisis que sirvió de guía para el posterior modelado del diseño, con los artefactos necesarios para cumplir con algunas de las tareas trazadas inicialmente. Acorde a la línea de arquitectura definida, se identificaron los subsistemas y las clases correspondientes, ajustándose además el patrón de diseño Singleton a esta propuesta. El mayor peso de este capítulo recae en el modelo de diseño, caracterizándose el mismo por su adaptabilidad y lograr conducir a una implementación satisfactoria del sistema.

CONCLUSIONES

Como consecuencia del proceso de investigación-desarrollo vinculado al presente trabajo se logró una comprensión de la situación problemática existente en el CIGB lo que permitió la obtención de una propuesta para su solución. Llevándose a término las tareas implícitas en los flujos de trabajo definidos, resultantes del estudio previo, acerca de las metodologías, tendencias y tecnologías a utilizar.

El modelo de dominio descrito y los requerimientos asociados a los casos de uso identificados del sistema fueron la entrada a la propuesta de diseño, ajustada al estilo arquitectónico y patrones definidos, que se realizó para satisfacer las necesidades del cliente final en cuanto a: procesar y cuantificar las imágenes de *microarray*.

Con los antecedentes mencionados se garantiza que el análisis y diseño del software para el procesamiento de imágenes de *microarray* constituye una guía legible y comprensible para quienes generan el código, además de proporcionar, al CIGB, una imagen completa del software en cuestión.

RECOMENDACIONES

A partir de las experiencias obtenidas en el desarrollo del trabajo y con vistas de lograr un aprovechamiento óptimo del resultado alcanzado se recomienda:

- Incorporar otras técnicas para la segmentación de la imagen, ganado en funcionalidad.
- Dar continuidad al modelo propuesto desde la perspectiva de implementación.
- Tomar como referencia el análisis del diseño realizado, para aplicar métricas que permitan evaluar el nivel de calidad de los resultados obtenidos, se exhorta a comenzar por las métricas orientadas a clases (26).
- Tener en cuenta el uso de software libre para próximos procesos de desarrollo.

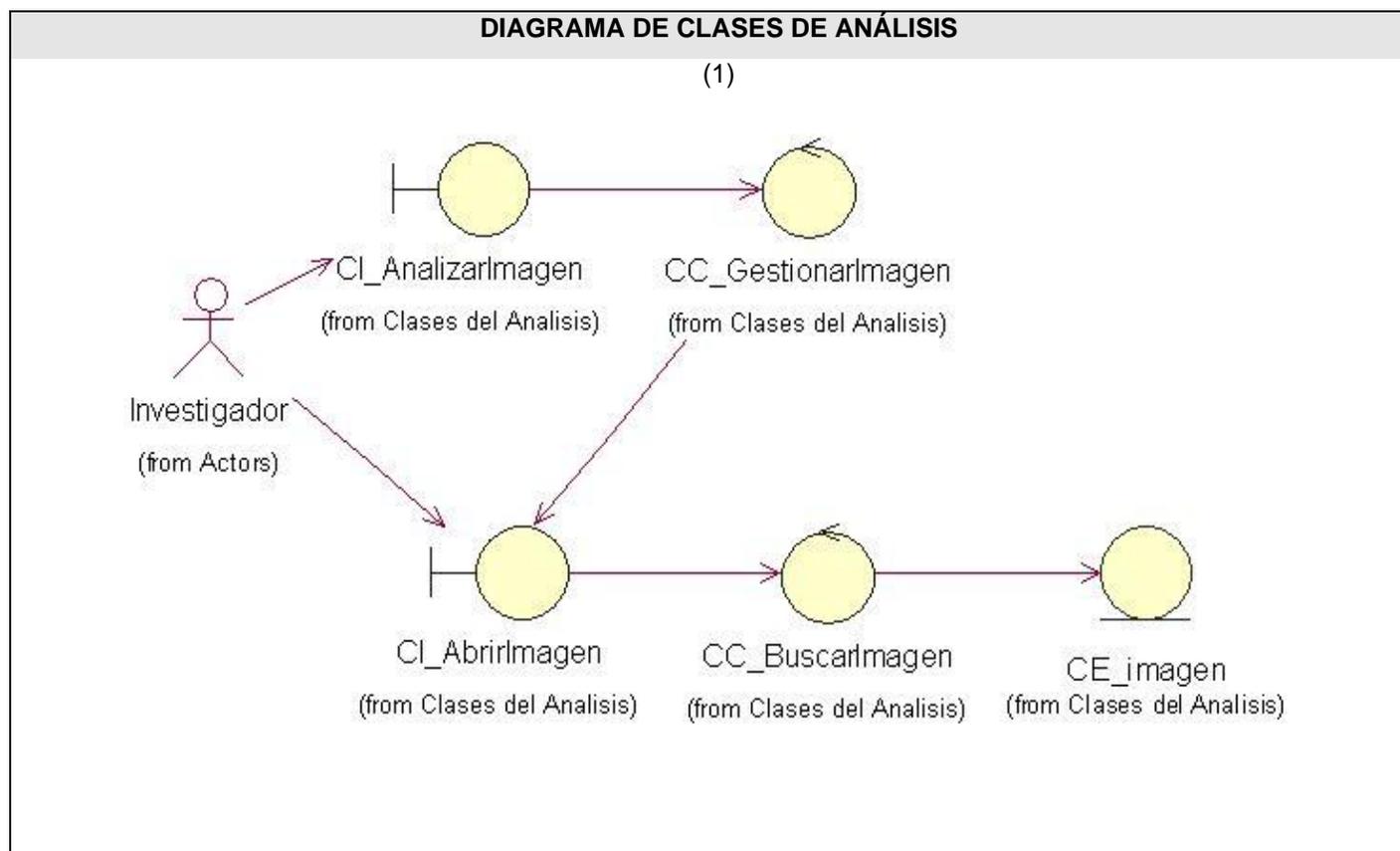
REFERENCIAS BIBLIOGRÁFICAS

1. Imaging Research Inc. www.imagingresearch.com. [En línea] http://www.imagingresearch.com/products/ARV_details.asp 2006.
2. TIGR *Spotfinder*. www.tm4.org. [En línea] [Citado el: 12 de mayo de 2006.] <http://www.tm4.org/spotfinder.html>.
3. Eisen, Michael. *ScanAlyze*. [En línea] Lawrence Berkeley National Laboratory, California, USA, 2002. <http://rana.lbl.gov>.
4. GSI Lumonics QuantArray Quantitative Microarray Analysis. [En línea] <http://www.bipl.ahc.umn.edu/quantarray.html>.
5. Grupo de Desarrollo de Negocios. *GNDP*. [En línea] Centro de Ingeniería Genética y Biotecnología. <http://gndp.cigb.edu.cu/>.
6. GeoSalud. [En línea] <http://geosalud.com/GenomaHumano/index.htm>.
7. *Quantitative monitoring of gene expression patterns with a complementary DNA microarray*. Schena M, Shalon D, Davis RW, Brown PO. 70, s.l. : Science, 2004. 270/467.
8. *Aplicaciones de la técnica de microarrays en ciencias biomédicas: presente y futuro*. Barrero, Dra. Paola Roxana. 3, 2005, *Revista QuimicaViva*. 1666/7948.
9. *From patterns to pathways: gene expression data analysis comes of age*. DK, Slonim. 32, 2002, *Nat Genet*, Vol. 2. 502/8.
10. MathWorks, Inc. [En línea] <http://www.mathworks.com/products/signal/>.
11. *Methods and approaches in the analysis of gene expression data*. Dopazo J, Zanders E, Dragoni I, Amphlett G, Falciani F. 2001, *J Immunol Methods*. 93/112.
12. Kelmansky, Dra. Diana M. *Análisis Exploratorio y Confirmatorio de Datos de Experimentos de Microarrays*. Dpto. de Matemática, Instituto de Cálculo. 1er. Cuatr. 2006.
13. González, Prof. Antonio Mosquera. *Tratamiento Computacional de Imágenes*. [En línea] 19 de abril de 2005. <http://gva1.dec.usc.es/~antonio/docencia/20042005tci/teoria/similaridad/Similaridad.html>.
14. Goldstein, Darlene. *Estadística para microarray*. [En línea] <http://statwww.epfl.ch/davison/teaching/Microarrays/ETHZ/>.
15. Sanchez, Ing. María A. Mendoza. *informatízate*. [En línea] www.informatizate.net/articulos/pdfs/metodologias_de_desarrollo_de_software_.

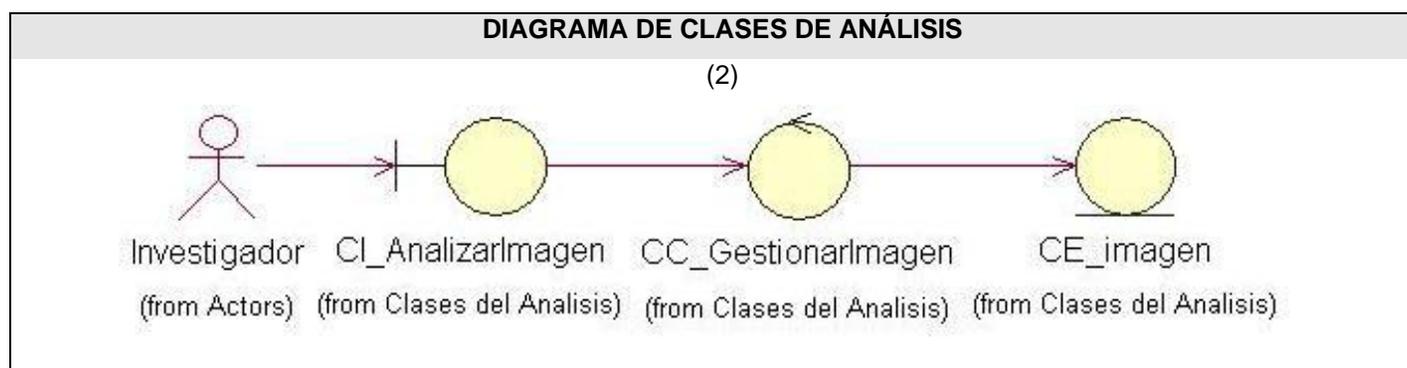
16. Ivar Jacobson, Grady Booch y James Rumbaugh. ***El Proceso Unificado de desarrollo de Software. Vol. 1.***
17. Pressman, Roger. ***INGENIERIA DEL SOFTWARE IV Edición.***
18. Pressman, R.S. ***Ingeniería de software. Un enfoque práctico. s.l. : Mc Gram Hill, 1998. Vol. 1.***
19. Microsoft Corporation. msdn. [En línea] <http://msdn2.microsoft.com/en-us/netframework/default.aspx>.
20. Sun Microsystems, Inc. Sun Developer Network(SDN). [En línea] <http://java.sun.com/javaee/index.jsp>.
21. Microsoft Corporation. msdn. [En línea] [http://msdn2.microsoft.com/es-es/library/88fx1xy0\(VS.80\).aspx](http://msdn2.microsoft.com/es-es/library/88fx1xy0(VS.80).aspx).
22. Facultad de Ingeniería en Electricidad y Computación. [En línea] <http://www.fiec.espol.edu.ec/investigacion/com-analog/documentos/matlab.pdf>.
23. Microsoft. [En línea] [Citado el: 5 de enero de 2007.] <http://www.microsoft.com/spanish/msdn/vcsharp>.
24. IEEE. ***“IEEE Recommended Practice for Architectural Description of Software Intensive Systems”.*** 2000. IEEE Std 1471/2000.
25. Gamma E., Helm, R., Johnson, R., Vlissides J. ***Design Patterns: Elements of Reusable Object Oriented Software.*** s.l. : Addison Wesley, 1995.
26. Lorenz, M. ***OBJECT-ORIENTED SOFTWARE METRICS.***

ANEXOS

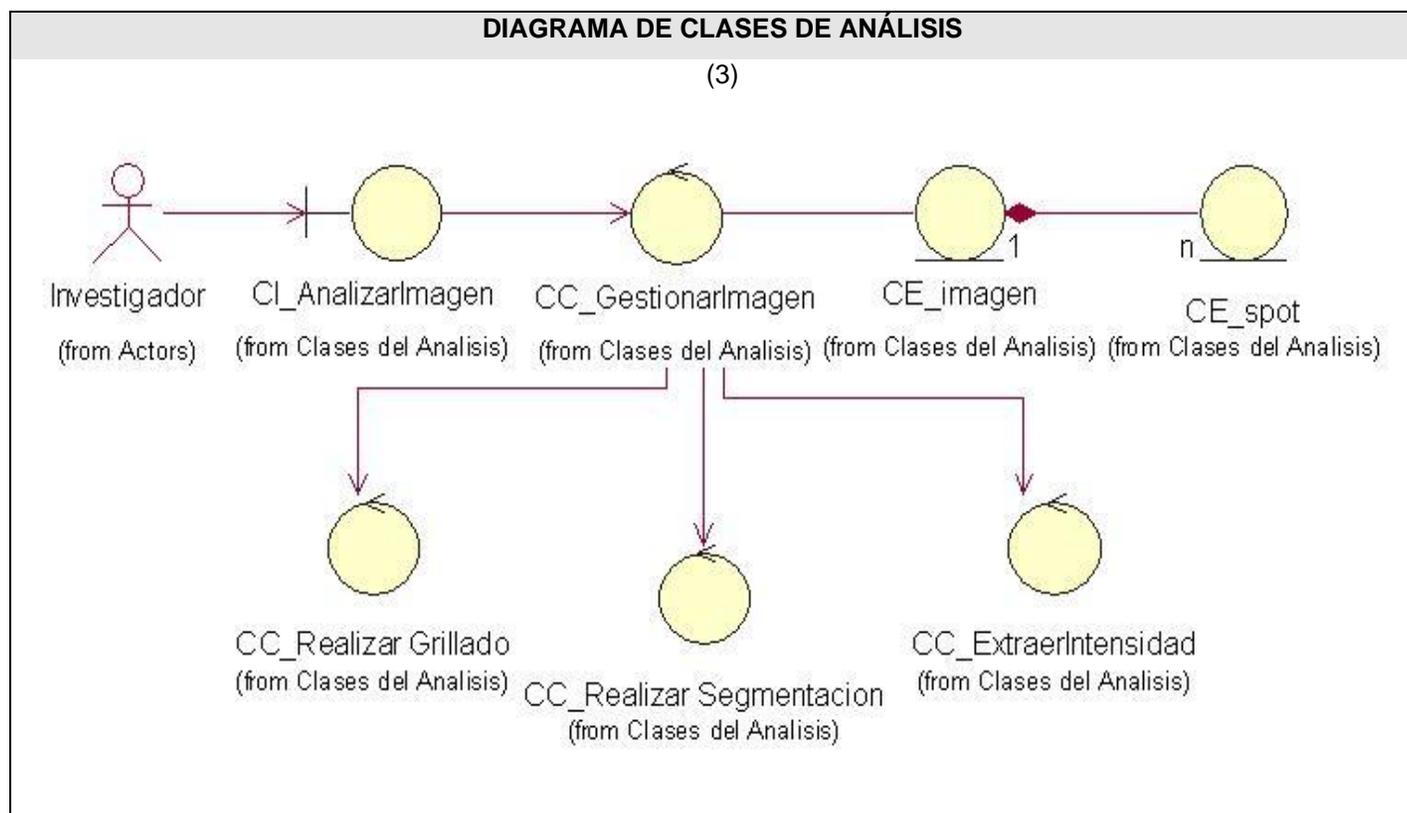
Anexo 1 CU-Cargar Imagen.



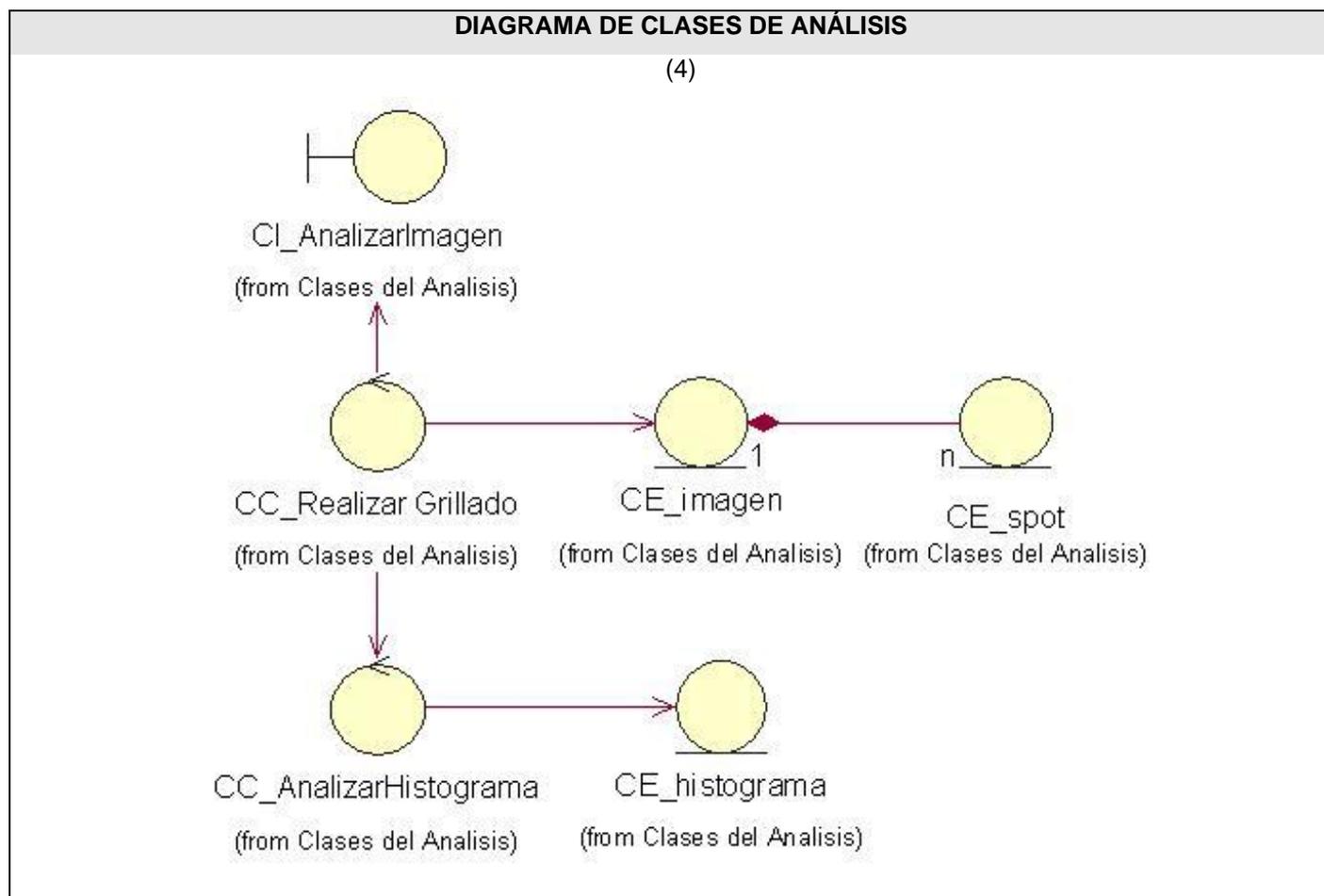
Anexo 2 CU-Seleccionar Región de Interés.



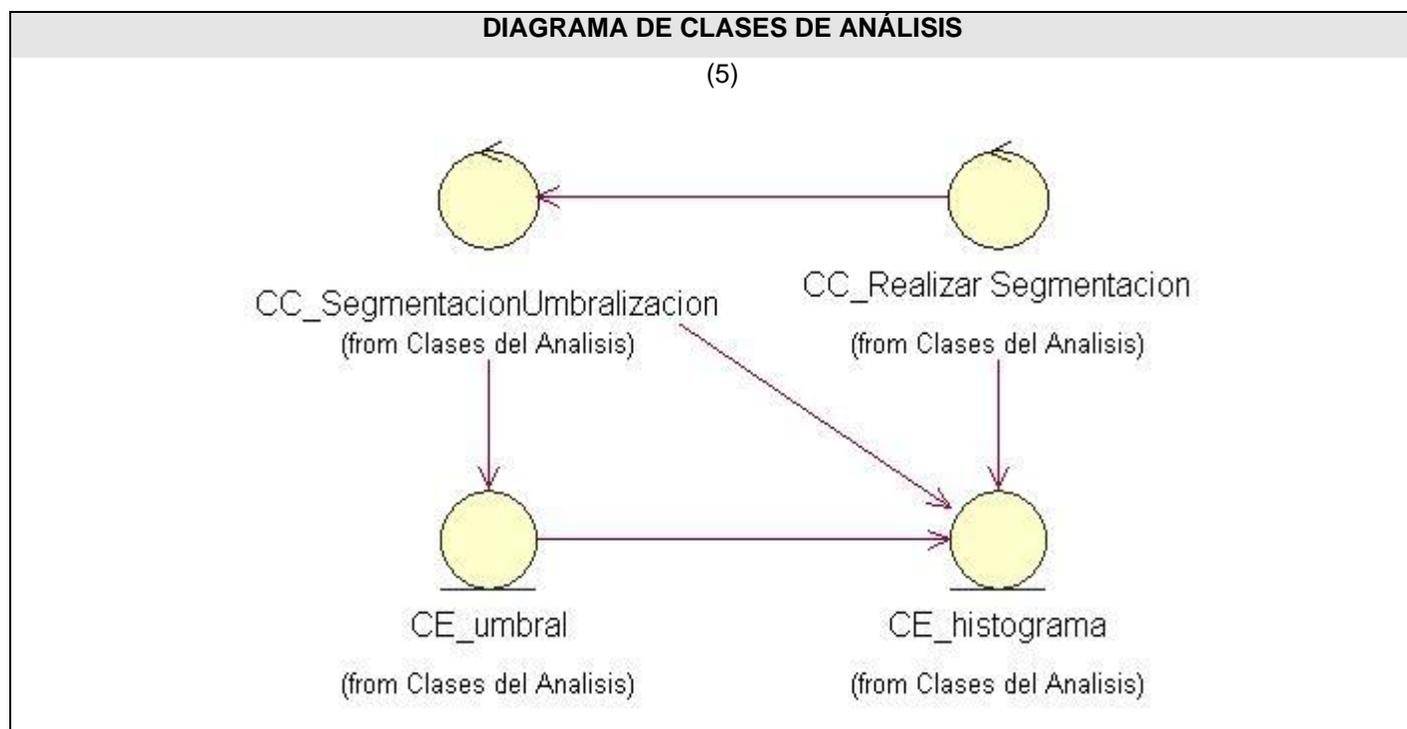
Anexo 3 CU-Procesar Imagen.



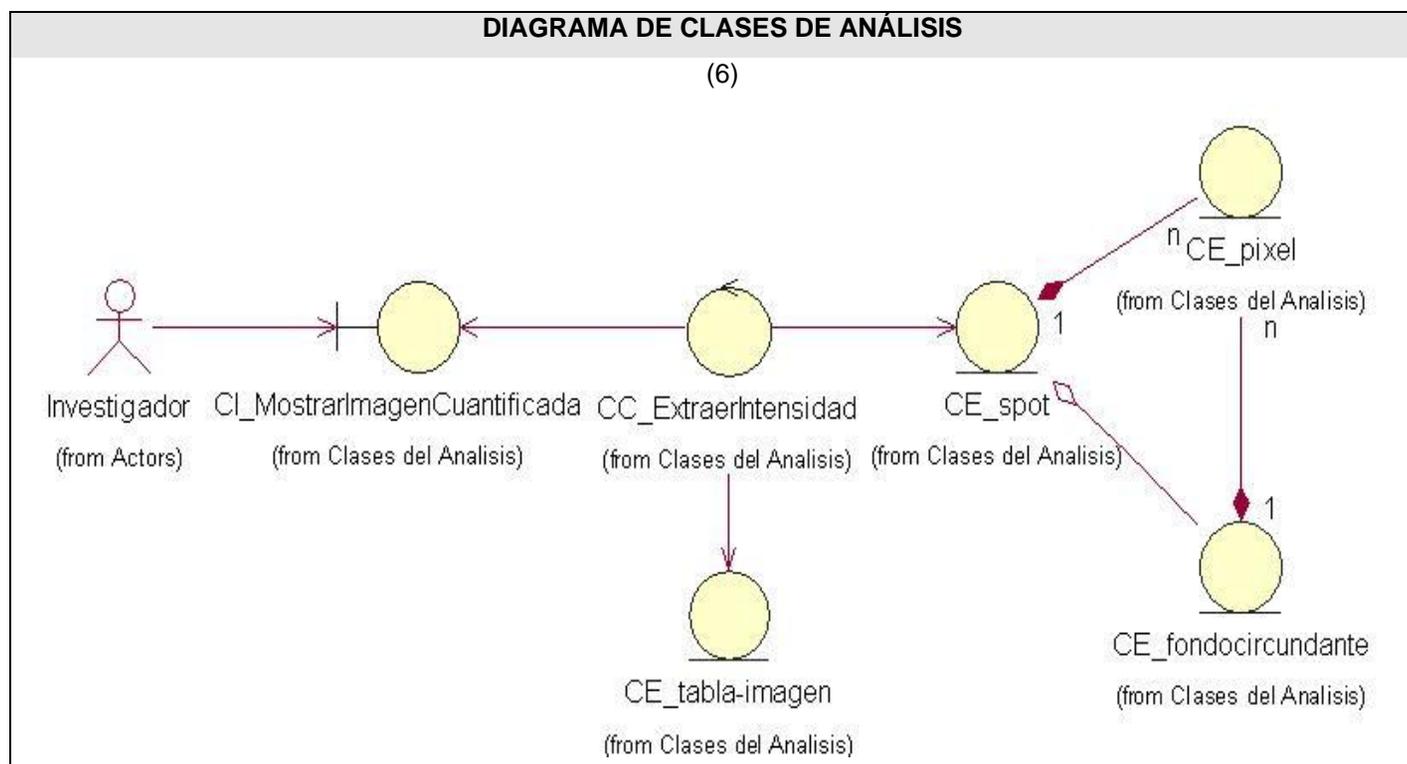
Anexo 4 CU-Realizar Grillado.



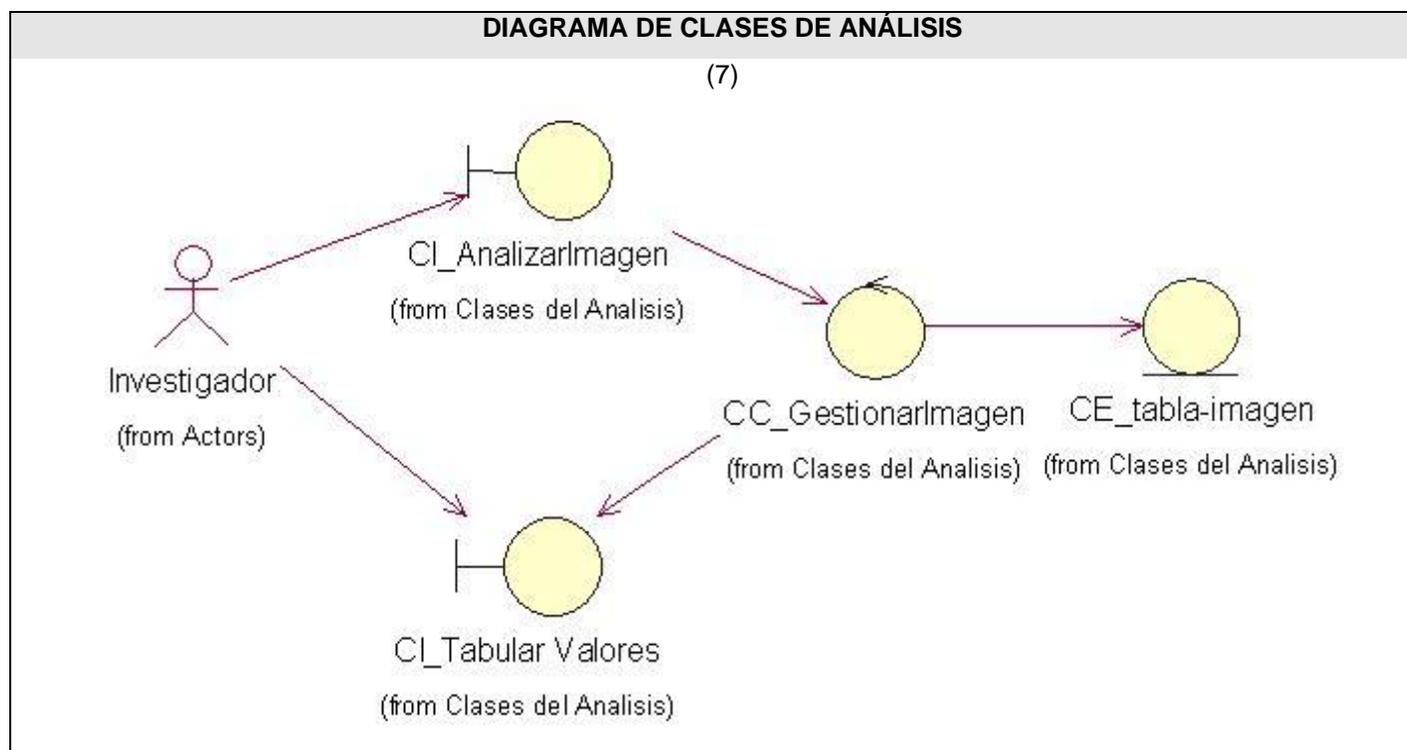
Anexo 5 CU-Realizar Segmentación.



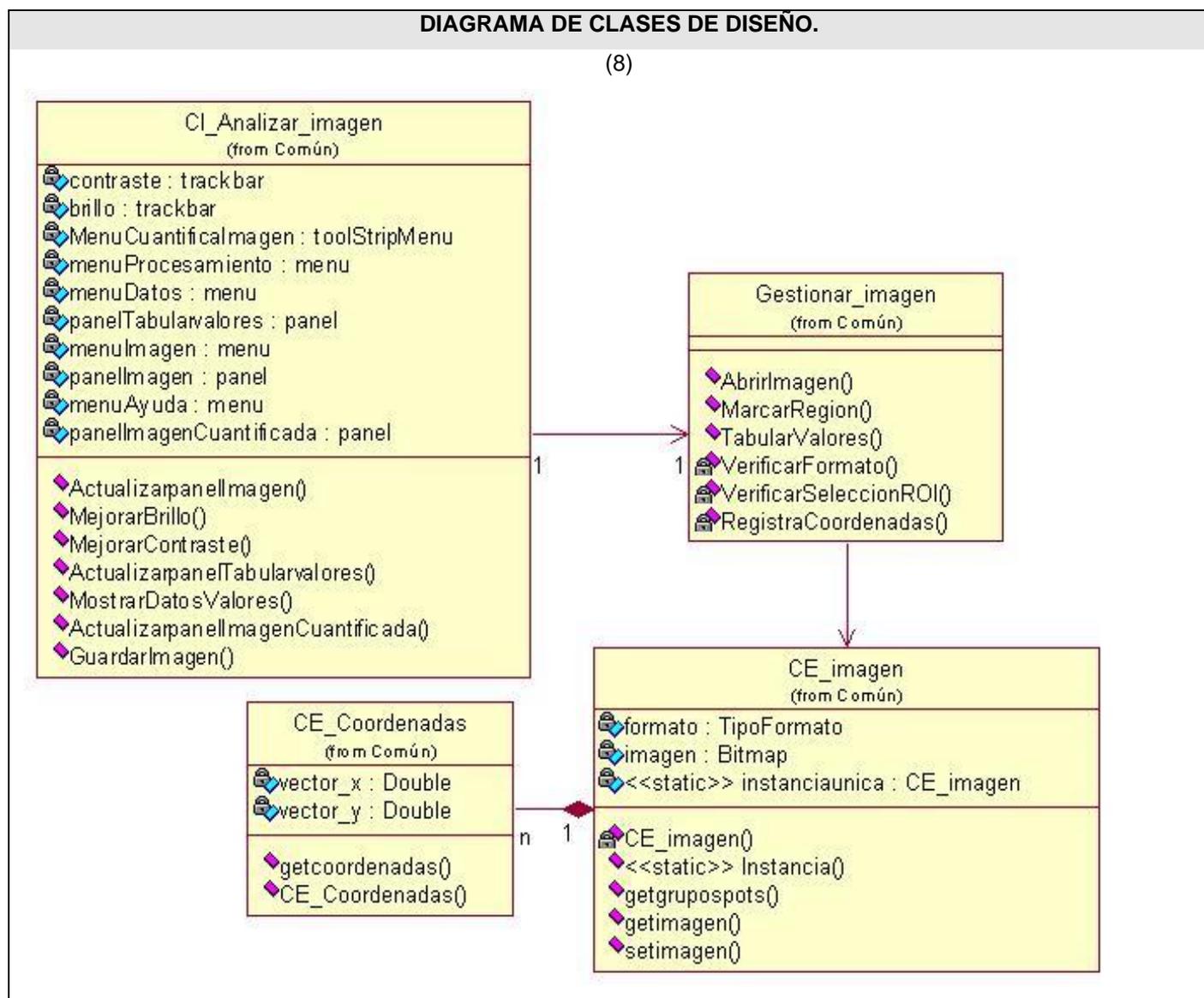
Anexo 6 CU-Extraer Intensidad.



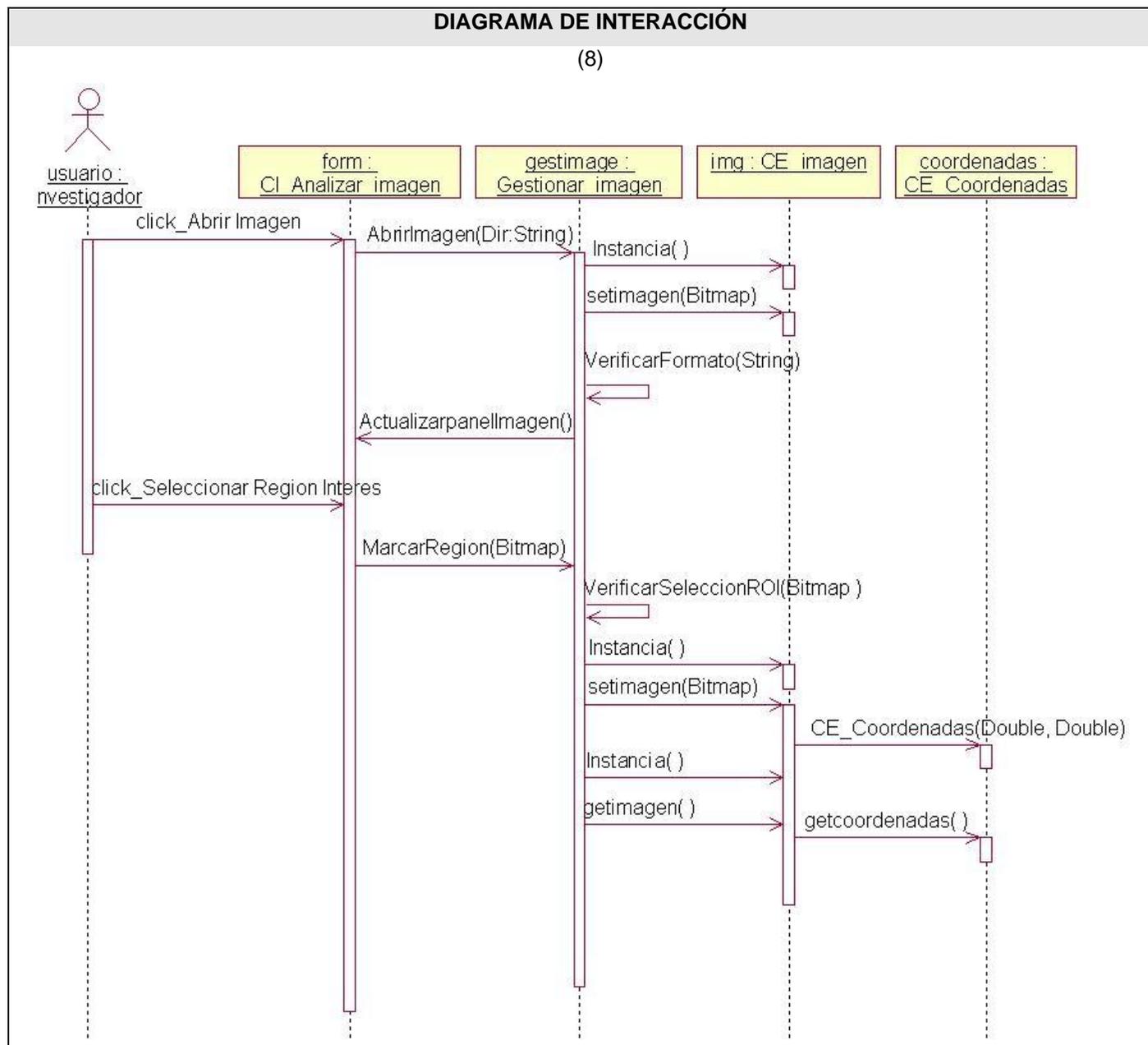
Anexo 7 CU-Tabular Valores.



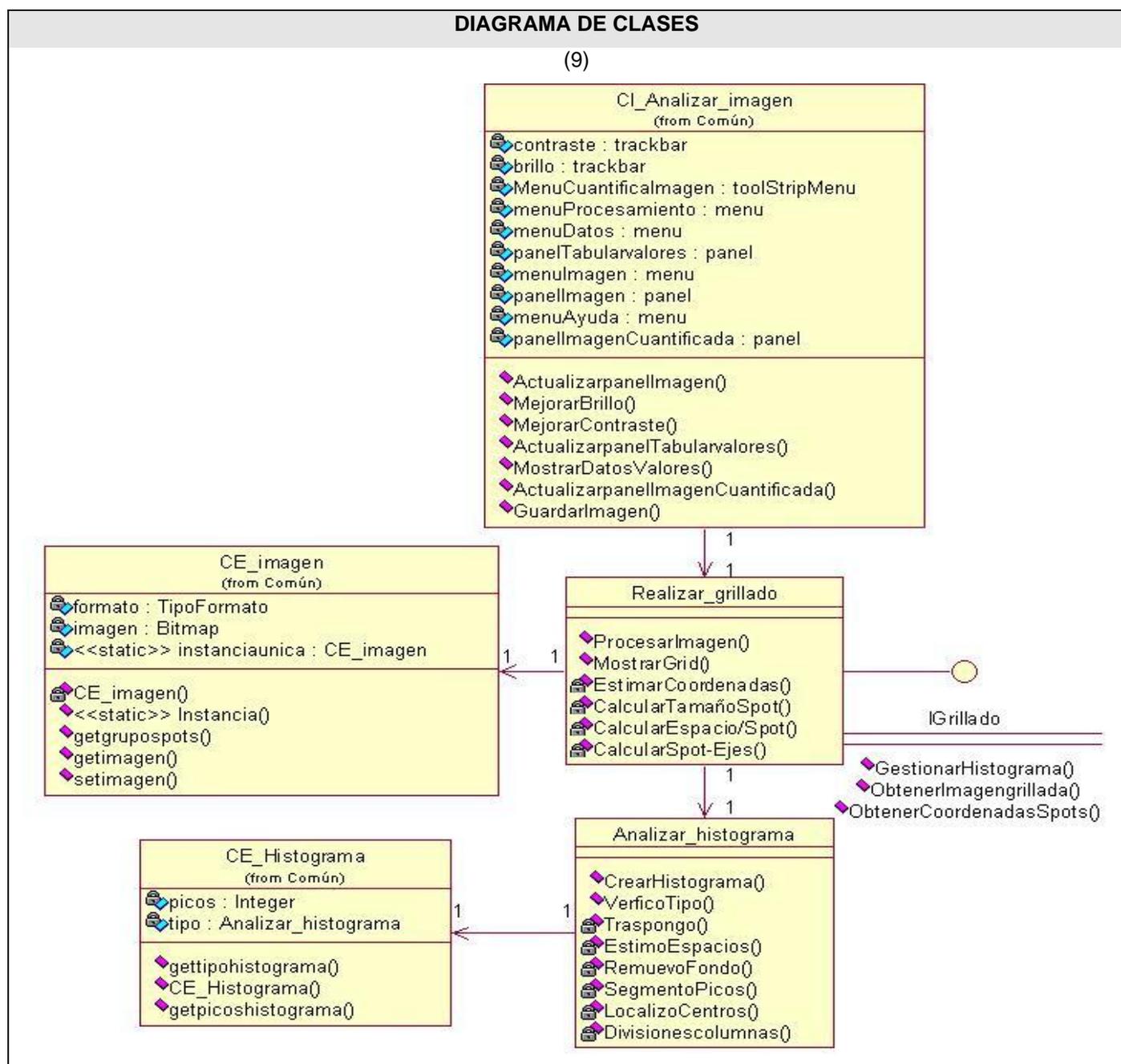
Anexo 8 Subsistema GestionarImagen (representa CU Cargar Imagen y CU Seleccionar ROI).



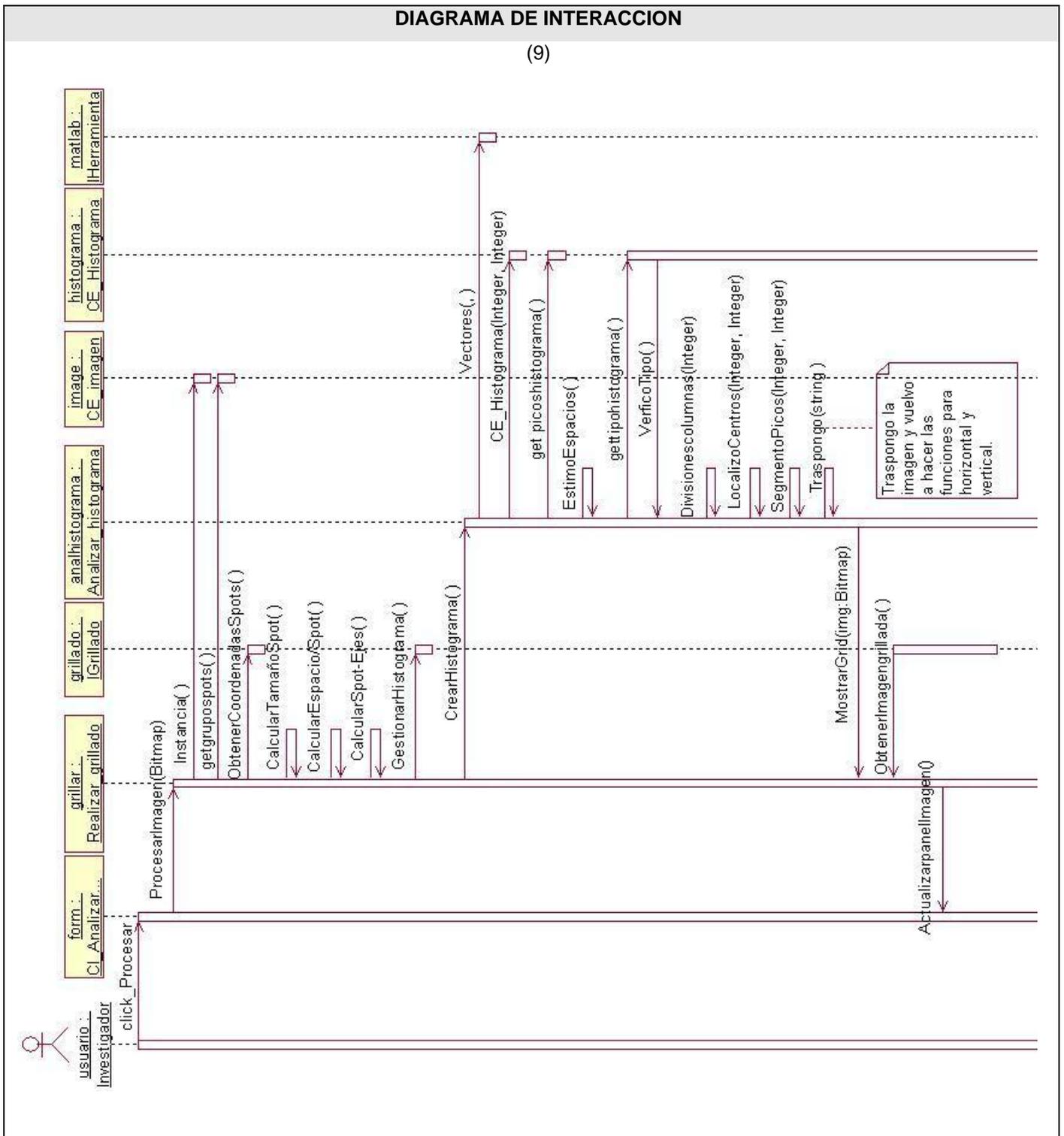
Flujo correspondiente CU Cargar Imagen y CU Seleccionar Región de Interés.



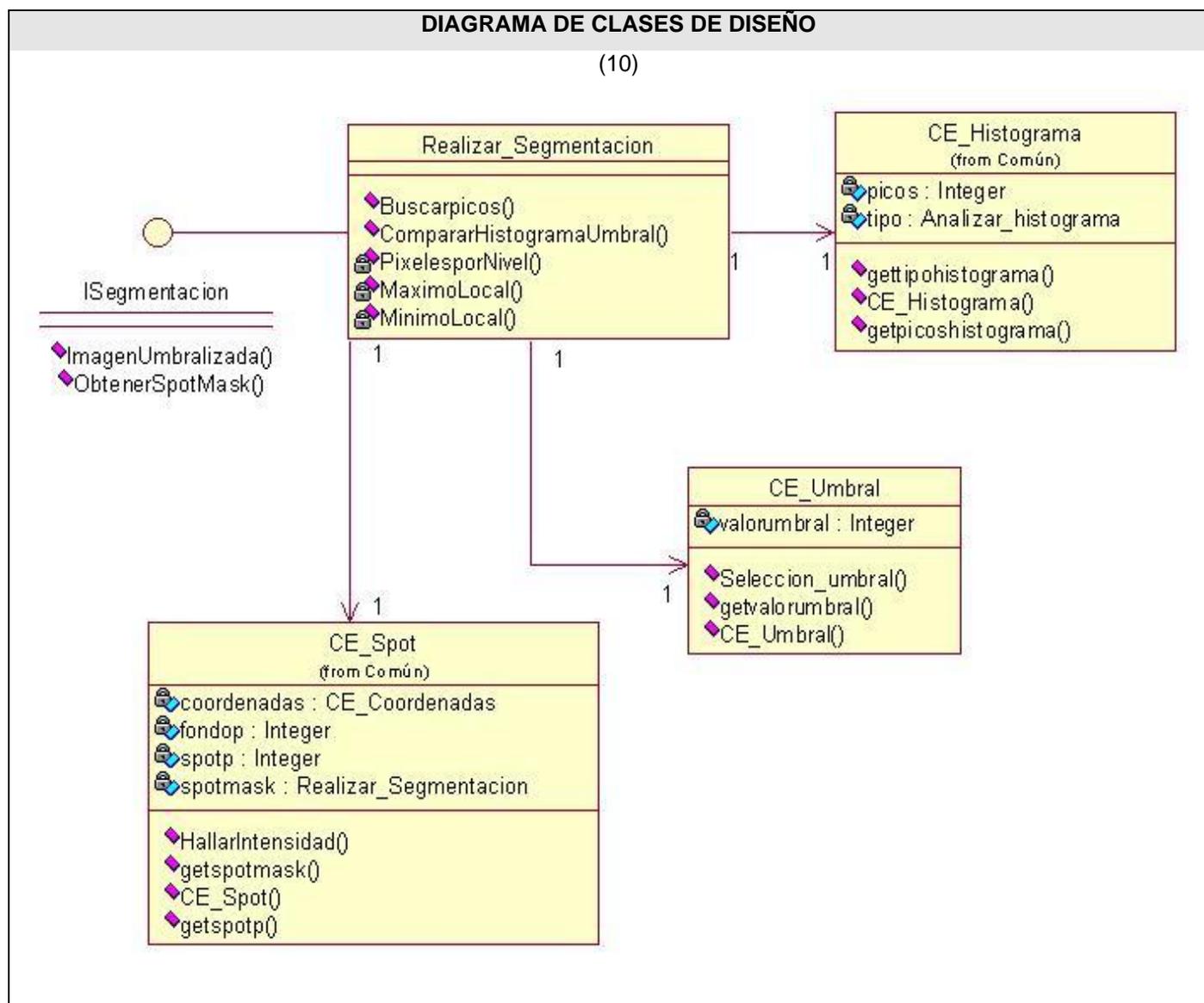
Anexo 9 Subsistema RealizarGrillado, representa CU Realizar Grillado.



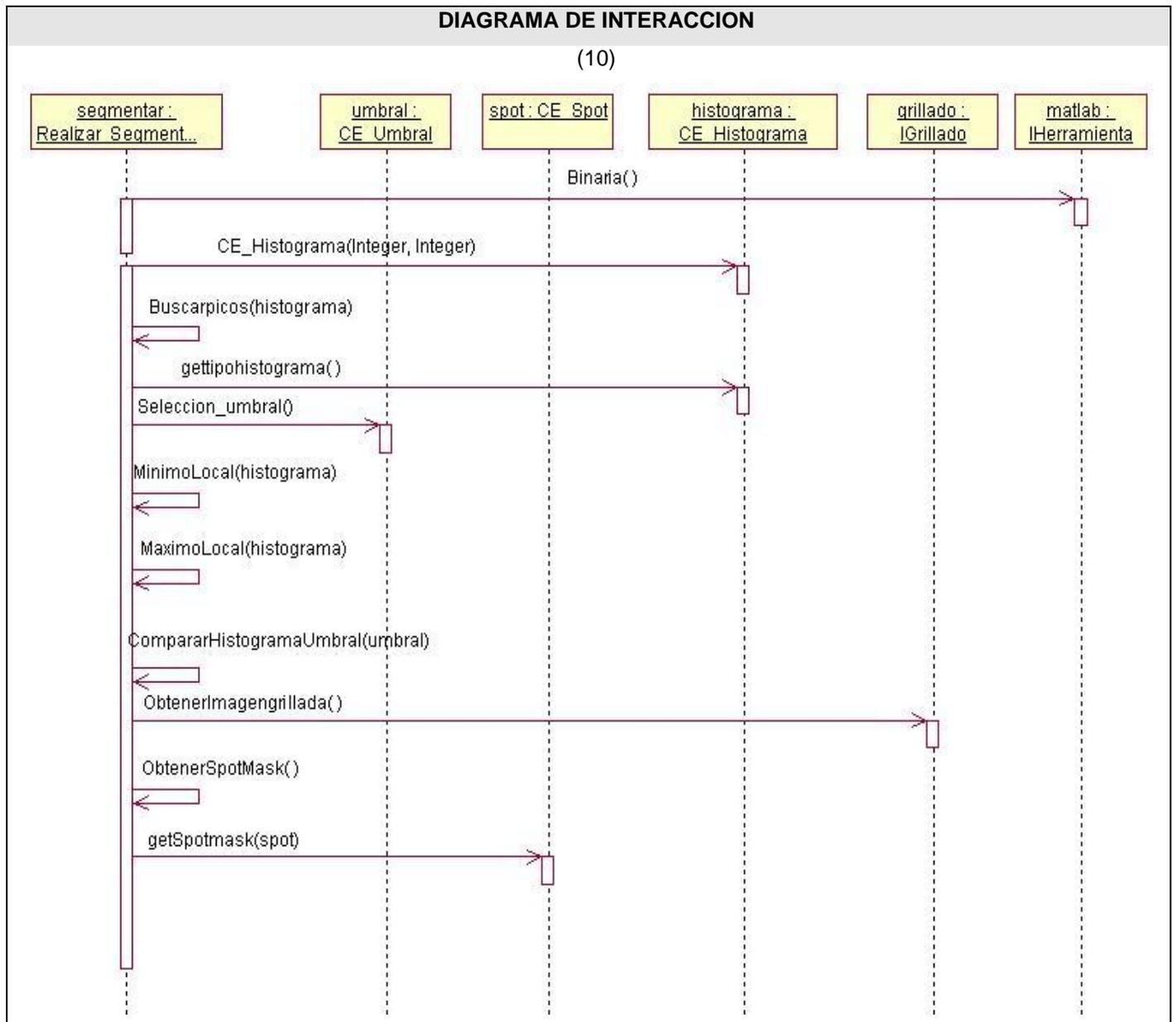
Flujo correspondiente al CU Realizar Grillado.



Anexo 10 Subsistema RealizarSegmentación, representa CU Realizar Segmentación



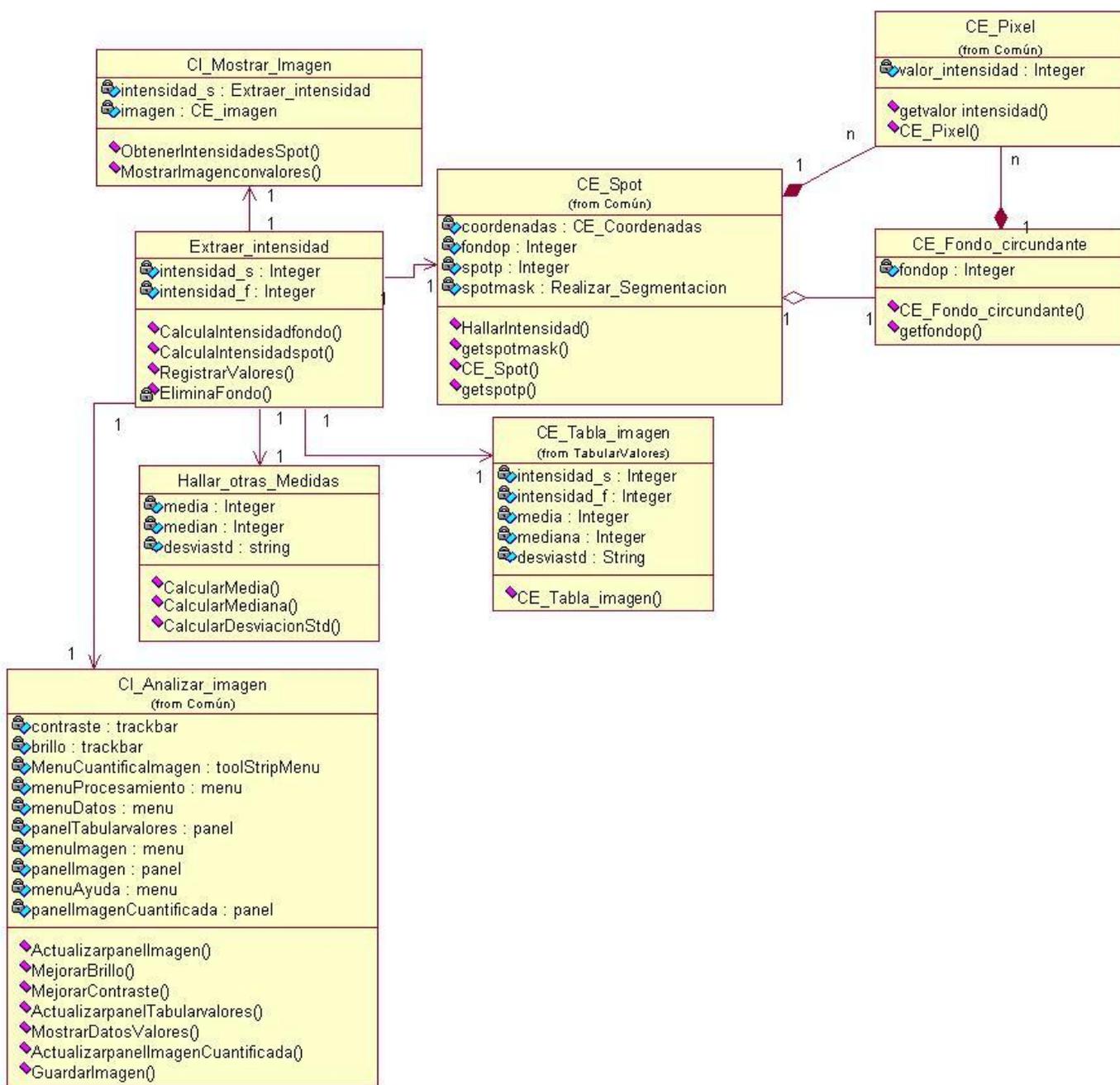
Flujo correspondiente al CU Realizar Segmentación.



Anexo 11 Subsistema Extraer Intensidades, representa CU Extraer Intensidades.

DIAGRAMA DE CLASES DE DISEÑO

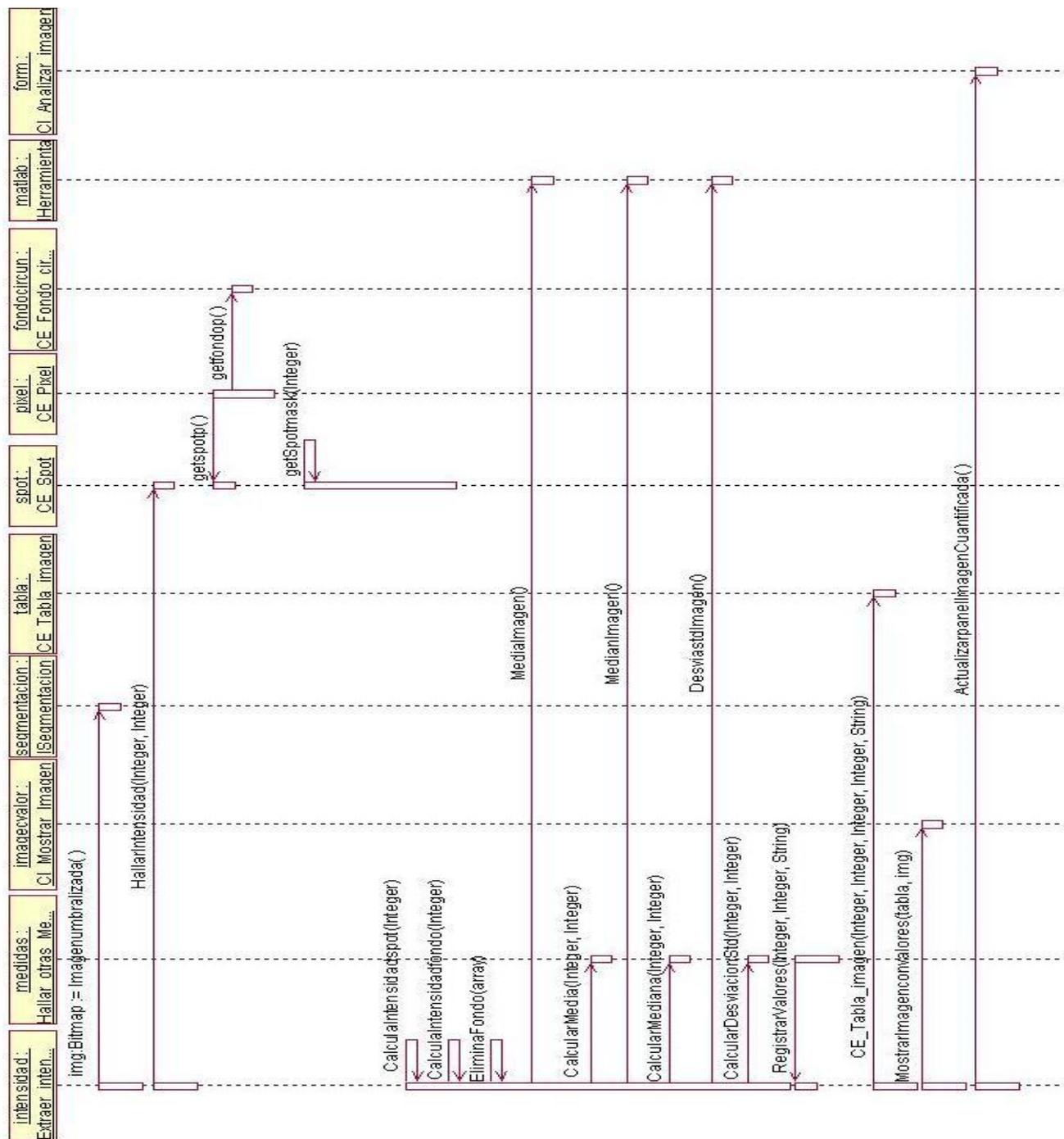
(11)



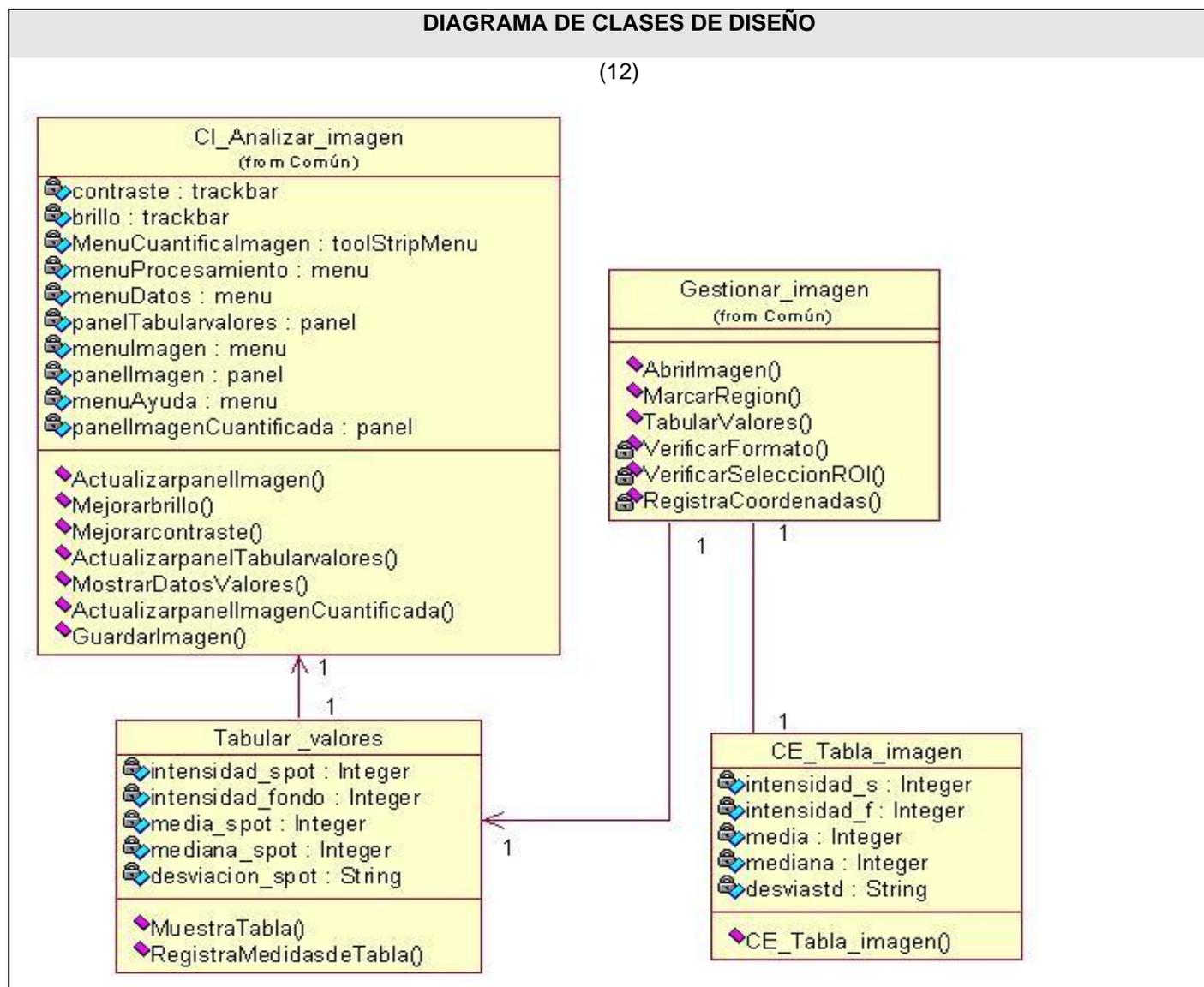
Flujo correspondiente al CU Extraer Intensidades.

DIAGRAMA DE INTERACCION

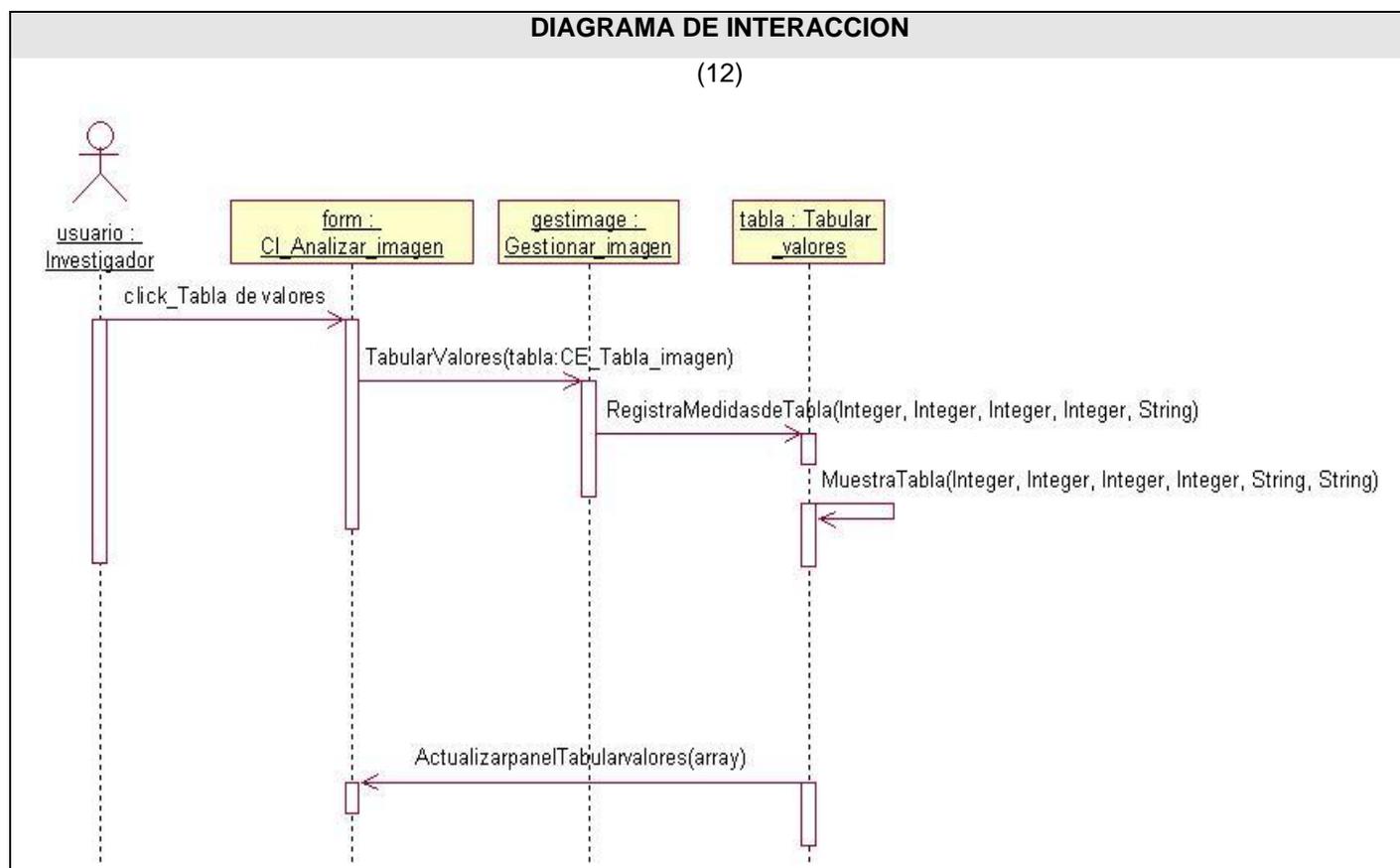
(11)



Anexo 12 Paquete TabularValores, representa CU Tabular Valores.



Flujo correspondiente al CU Tabular Valores.



GLOSARIO DE TÉRMINOS**A**

ADN: abreviatura del ácido desoxirribonucleico; es el componente químico primario de los cromosomas y el material en el que los genes están codificados.

C

cADN (ADN complementario): ADN sintetizado por transcriptasa inversa utilizando RNA como plantilla.

E

Extraer Intensidad: calcular para cada *spot* del *microarray* el valor de intensidad de fluorescencia.

F

Fluorescencia: La fluorescencia es la propiedad de una sustancia para emitir luz cuando es expuesta a radiaciones del tipo ultravioleta, rayos catódicos o rayos X.

G

Genoma: término que puede ser utilizado en más de un sentido, en dependencia del contexto, pero se refiere a todo el material genético de una célula u organismo.

Gen: segmento de ADN que sirve como unidad de información hereditaria. Coincide con los *spots* que representan la imagen de *microarray*.

Grillado: es el proceso de asignación de coordenadas a cada uno de los *spots*

H

Hibridización o hibridación: se refiere a uno de los tipos más comunes de técnicas usadas para estudiar el ADN y el ARN, empleando fragmentos de ADN o sondas de cadena sencilla marcada con una molécula.

I

Imagen de *microarray*: provee representaciones numéricas bidimensionales de las intensidades de fluorescencia. Cada número corresponde a la intensidad de un *píxel*, que conforman la imagen.

M

Microarray: método que permite estudiar cómo interactúan entre sí un gran número de genes. Tecnología para investigar los niveles de la expresión de millares de genes simultáneamente.

mRNA(RNA mensajero): plantilla para la síntesis de proteínas. La secuencia del mRNA depende de la secuencia de una cadena complementaria de ADN.

Micrótopo: es un dispositivo mecánico que permite la elaboración de cortes finos de muestras (suelen ser tejidos con patologías) para su observación al microscopio.

S

Spot: los *spots* de los *microarray* son regiones fluorescentes localizadas sobre la superficie del sustrato.

Segmentación: puede definirse como el proceso de particionar la imagen en diferentes regiones con diferentes propiedades. Permite clasificar los *píxeles* en foreground (*spot* de interés) o background (fondo).

SNP(polimorfismo de un solo nucleótido): Posición de la secuencia de ADN genómico que varía de un individuo a otro, es una variación en la secuencia de ADN que afecta a un solo nucleótido (adenina (A), timina (T), citosina (C) o guanina (G)) del genoma.

O

Oligo: el oligonucleótido es una secuencia corta de cadena única de ADN o ARN. Los oligos se usan a menudo como sondas para detectar ADN o ARN complementario.

P

Píxel: tiene una correspondencia biunívoca con una pequeña porción cuadrada de la imagen, es fácilmente visible si la imagen es ampliada en la pantalla de una computadora.

Procesamiento: el objetivo es extraer, para cada *spot* fijado en el *microarray*, una medida de su intensidad; el procesamiento de imágenes de *microarray* se puede dividir en 3 tareas Grillado, Segmentación, Extraer Intensidades.

Proteína: una molécula compuesta por una o más cadenas de aminoácidos. Las proteínas desempeñan actividades vitales en la célula.

PCR (Reacción en cadena de la polimerasa): es una técnica de biología molecular cuyo objetivo es obtener un gran número de copias de un fragmento de ADN particular, a partir de un mínimo.