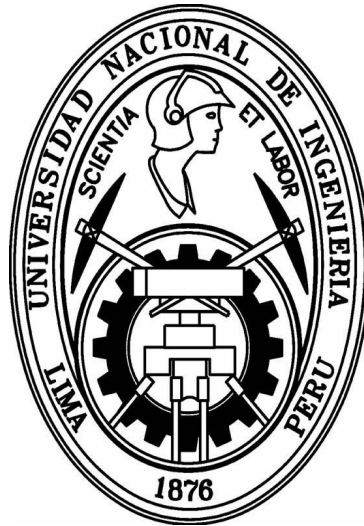


**UNIVERSIDAD NACIONAL DE INGENIERÍA  
FACULTAD DE CIENCIAS**

**Sección de Posgrado y Segunda Especialización Profesional**



Tesis para Optar el Grado Académico de  
Maestro en Ciencias con mención en Energía Nuclear

**“Software para el Conteo y Clasificación de Huellas  
Nucleares Alfa de Incidencia Normal Sobre el Polímero  
CR-39”**

Presentado por:

Luis Alberto Sánchez Rodas

Asesor:

Dr. Agustín Zúñiga Gamarra

**LIMA - PERÚ**

**2010**

Dedicado con mucho cariño a mis padres:  
Toelinda Rodas Ramirez  
Regulo Sánchez Figueroa

## **AGRADECIMIENTOS**

Deseo expresar mi más sincero agradecimiento a todas aquellas personas que de un modo u otro han hecho posible la realización de esta tesis: al Dr. Agustín Zúñiga gamarra que eligió el tema de investigación y de quien recibí todo el apoyo, dedicación y colaboración científica que las tareas de investigación precisaron a lo largo de estos años.

A los doctores Juan Rodríguez, José Solís, Justo Rojas y Walter Estrada por haberme brindado la oportunidad de compartir con ellos la maestría.

Finalmente, a todos los miembros de mi familia, quienes me han brindado todo su apoyo tanto moral como económico. Mi más enorme gratitud a mis padres, Toelinda y Regulo, y a mis hermanos, Nel, Miguel y Flor.

## RESUMEN

El paso de partículas cargadas a través de un sólido dieléctrico crea zonas de daño intenso a escala atómica llamadas huellas latentes. Por medio de un ataque químico con un agente químico adecuado es posible aumentar el tamaño de las huellas hasta hacerlas visibles bajo un microscopio óptico después de lo cual pueden ser capturadas usando un sistema de adquisición de imágenes. Cuando se analizan las imágenes de los daños dejados por partículas alfa, después del ataque químico, estas presentan diversidad de formas, tamaños y tonalidades. A partir de la determinación de la densidad de huellas (número de huellas /  $\text{cm}^2$ ) y de la determinación de características geométricas y ópticas de las mismas es factible obtener algunas aplicaciones tecnológicas, dos de las aplicaciones más importantes de las huellas nucleares alfa son la dosimetría y la espectroscopia alfa.

En este trabajo de tesis se desarrolló un software para el conteo, caracterización y clasificación de imágenes de huellas de partículas alfa de incidencia normal sobre detectores de estado sólido CR-39. Usando para ello técnicas de procesamiento digital de imágenes y redes neuronales artificiales haciendo uso intensivo de las cajas de herramientas *Image Processing Toolbox* y *Neural Network Toolbox* de Matlab.

El software desarrollado consta de 2 módulos: El primer módulo está orientado al conteo y determinación automática de los parámetros de las huellas, parámetros tales como: área, eje mayor, eje menor, excentricidad, opacidad, radio equivalente, etc. En total 22 parámetros pueden ser determinados con este módulo. El segundo módulo ofrece la capacidad de clasificar las huellas de acuerdo a la energía de las partículas alfa incidentes que las causan.

Para solucionar el problema de clasificación se diseñó e implementó una red neuronal multicapa tipo “backpropagation”. Para obtener los patrones entrada-salida necesarios para la etapa de entrenamiento de la red neuronal, se hizo un análisis de los parámetros de huellas extraídas de una base de datos de imágenes características de huellas de partículas alfa con energías entre 0.15 y 8 MeV.

Finalmente se dotó al software de una interfaz gráfica de usuario amigable escrita en Java usando las librerías de componentes gráficos *Swing* y *SwingX* y que integra los algoritmos desarrollados en Matlab, gracias a la tecnología *Matlab Builder JA*, en una sola aplicación, proporcionando una manera cómoda e intuitiva de manejar el software. Además de permitir la elección del módulo con la cual desea trabajar así como el ingreso de los distintos parámetros, necesarios para las etapas de preprocesamiento y segmentación de las imágenes, el software permite el procesamiento de una o un grupo de imágenes, realizar un procesamiento supervisado o no supervisado de estas. Adicionalmente el software permite gravar, exportar los resultados a una hoja de cálculo Excel para su posterior tratamiento.

# ÍNDICE GENERAL

<b>DEDICATORIA.....</b>	<b>I</b>
<b>AGRADECIMIENTOS.....</b>	<b>II</b>
<b>RESUMEN .....</b>	<b>III</b>
<b>1. INTRODUCCIÓN.....</b>	<b>1</b>
<b>2. MARCO TEÓRICO.....</b>	<b>4</b>
<b>2.1. INTERACCIÓN DE PARTÍCULAS CARGADAS CON LA MATERIA.....</b>	<b>4</b>
2.1.1. Poder de frenado .....	5
2.1.2. La fórmula de Bethe-Bloch.....	7
2.1.3. La curva de Bragg .....	8
2.1.4. Ley de Bragg .....	9
2.1.5. Rango .....	9
2.1.6. Dispersión estadística de energía (straggling) .....	10
2.1.7. Dispersión múltiple Coulombiana.....	10
<b>2.2. DETECTORES DE HUELLAS NUCLEARES DE ESTADO SÓLIDO .....</b>	<b>11</b>
2.2.1. Mecanismo de formación de una huella.....	12
2.2.2. Ataque químico .....	13
2.2.3. Geometría de la formación de una huella.....	14
<b>2.3. PROCESAMIENTO DIGITAL DE IMÁGENES .....</b>	<b>17</b>
2.3.1. Representación Digital de una Imagen.....	17
2.3.2. Técnicas de procesamiento digital de Imágenes (PDI) .....	18
2.3.3. Procesamiento morfológico de imágenes.....	22
<b>2.4. REDES NEURONALES ARTIFICIALES .....</b>	<b>25</b>
2.4.1. Neurona Biológica .....	25
2.4.2. Neurona Artificial.....	27
2.4.3. Red Neuronal Artificial.....	27
2.4.4. Características de una Red Neuronal .....	31
2.4.5. Aspectos para el diseño de la topología.....	36
2.4.6. Red Backpropagation .....	37
<b>3. INSTRUMENTACIÓN .....</b>	<b>43</b>
<b>3.1. BANCO DE DATOS DE IMÁGENES .....</b>	<b>43</b>
3.1.1. Fuentes emisoras alfa.....	43
3.1.2. Montaje de irradiación.....	44

3.1.3. Ataque químico de los detectores.....	45
3.1.4. Adquisición de las imágenes de las huellas.....	45
<b>3.2. DEFINICIÓN DE LAS ENERGÍAS .....</b>	<b>46</b>
3.2.1. Determinación de la energía de incidencia .....	46
<b>4. CONTEO Y CLASIFICACIÓN DE HUELLAS NUCLEARES ALFA.....</b>	<b>52</b>
<b>4.1. PREPROCESAMIENTO .....</b>	<b>52</b>
4.1.1. Eliminación del ruido .....	53
4.1.2. Mejorando el contraste de la imagen.....	55
<b>4.2. SEGMENTACIÓN.....</b>	<b>57</b>
4.2.1. Binarización.....	58
4.2.2. Detección de contornos .....	59
4.2.3. Etiquetado.....	60
4.2.4. Extracción automática de huellas .....	61
<b>4.3. EXTRACCIÓN DE CARACTERÍSTICAS .....</b>	<b>63</b>
4.3.1. Características de forma .....	64
4.3.2. Contraste de la huella.....	66
4.3.3. Contraste del fondo.....	68
4.3.4. Contraste relativo .....	69
<b>4.4. CONTEO DE HUELLAS NUCLEARES ALFA .....</b>	<b>70</b>
<b>4.5. CLASIFICACIÓN DE LAS HUELLAS NUCLEARES ALFA A TRAVÉS DE UN CLASIFICADOR BASADO EN RNAS .....</b>	<b>70</b>
4.5.1. Definir la RNA a utilizar.....	70
4.5.2. Definir patrones de entrenamiento.....	70
4.5.3. Preprocesar la información.....	71
4.5.4. Probar el clasificador.....	72
<b>5. DISEÑO DEL SOFTWARE .....</b>	<b>73</b>
<b>5.1. DESCRIPCIÓN GENERAL DEL SOFTWARE .....</b>	<b>73</b>
5.1.1. Barra de herramientas.....	74
5.1.2. Modos de operación .....	75
5.1.3. Tipos de proceso .....	75
5.1.4. Resultados.....	76
<b>5.2. HERRAMIENTAS DE DESARROLLO .....</b>	<b>80</b>
5.2.1. Matlab® .....	81
5.2.2. Java.....	83
<b>6. RESULTADOS Y DISCUSIONES.....</b>	<b>86</b>
<b>6.1. ANÁLISIS DE IMÁGENES DE HUELLAS NUCLEARES ALFA.....</b>	<b>86</b>
6.1.1. Fuentes de Uranio .....	86

6.1.2. Fuentes de Torio .....	90
<b>6.2. CLASIFICACIÓN DE LAS HUELLAS NUCLEARES ALFA A TRAVÉS DE UN CLASIFICADOR BASADO EN RNAS .....</b>	<b>101</b>
6.2.1. Definir RNA a utilizar.....	101
6.2.2. Definir patrones de entrenamiento.....	101
6.2.3. Preprocesamiento de la información .....	102
6.2.4. Pruebas del clasificador.....	106
<b>7. CONCLUSIONES .....</b>	<b>118</b>
<b>8. BIBLIOGRAFÍA.....</b>	<b>119</b>
<b>9. ANEXOS.....</b>	<b>124</b>
9.1. ANEXO 1.....	124
9.2. ANEXO 2.....	125
9.3. ANEXO 3.....	139
9.4. ANEXO 4.....	140
9.5. ANEXO 5.....	140

# 1. INTRODUCCIÓN

Cuando una partícula nuclear cargada, penetra un material dieléctrico, ya sea un sólido cristalino, vidrio o plástico, produce a lo largo de su trayectoria (del orden de los 20  $\mu\text{m}$  para partículas alfa de 5 MeV en un dieléctrico típico) una zona de daño intenso, de ancho del orden de los ángstroms, llamada huella latente.

En 1958, Young [1] descubrió una técnica particularmente sencilla para agrandar el daño o huella latente producido por partículas cargadas en dieléctricos al observar cavidades grabadas en la superficie de un cristal de fluoruro de litio que había sido irradiado con fragmentos de fisión y sometido posteriormente a un ataque químico con una mezcla de ácido fluorhídrico y acético.

Un Detector de Huellas Nucleares de Estado Sólido (SSNTD, por su siglas en inglés) es básicamente un material sólido dieléctrico en el cual un ataque químico, después de que este material es sometido a una irradiación, permite incrementar el tamaño de las huellas latentes, esto debido a que las regiones dañadas constituyeron zonas químicamente más activas que las regiones indemnes circundantes. La diferencia en la velocidad de ataque entre estas dos regiones crea canales de daño que aparecen como perforaciones del material, de diámetros del orden de las decenas de micrómetros, y que pueden ser observados con un microscopio óptico común de poco aumento [2,3].

La técnica descrita por Young no fue aprovechada si no hasta 1962, en el que R. L. Fleischer et.al [4,5] empezaron sus estudios referentes al daño producido por partículas cargadas en sólidos. A partir de ese momento, el método de revelado de huellas en sólidos por ataque químico ha tenido gran desarrollo, tanto en lo referente al comportamiento de los materiales detectores (SSNTD), el método de ataque químico y las aplicaciones que se le han dado. Entre estas últimas aparecen mencionadas en la literatura aplicaciones en múltiples campos como: física nuclear, física de reactores, geología, metalurgia, dosimetría, medicina, biología, rayos cósmicos, etc. [6-13]

La información contenida en las huellas reveladas son generalmente utilizadas de 2 maneras: se puede utilizar el número de huellas por unidad de área (Densidad superficial de huellas) para aplicaciones de dosimetría [14-16] o se puede utilizar las características de las imágenes de las huellas reveladas (diámetro mayor, diámetro menor, redondez, nivel de gris promedio, etc.). Los trabajos dentro de este segundo enfoque se iniciaron en la segunda mitad de la década del 60 [17]. Estos trabajos muestran que existe una relación entre las características de las huellas reveladas y las propiedades de la partícula incidente (carga, masa, energía etc.). Como consecuencia es posible emplear esta técnica en la identificación de partículas [18, 19] y para la espectroscopia de partículas alfa [20-23].

La espectroscopia de partículas alfa está adquiriendo una importancia creciente en el campo nuclear para la cuantificación e identificación de emisores alfa, clave en la gestión de residuos y caracterización ambiental. Los detectores activos proporcionan mediciones más exactas y precisas de la energía de las partículas alfa comparado a los detectores de huellas nucleares de estado sólido, SSNTDs. Sin embargo los SSNTDs tienen muchas ventajas sobre los detectores electrónicos tales como: bajo costo, fácil manejo, gran tiempo de exposición y permanente grabado de huellas. Esto hace de ellos adecuados para muestras de baja tasa así como también para muestras de alta tasa.



Una de las tareas más difíciles en la aplicación de detectores de huellas nucleares de estado sólido (SSNTDs) es la clasificación automática de huellas grabadas debido a las partículas alfa con diferentes energías de incidencia. En esta última década se reportaron algunos avances importantes en lo que respecta a este tema, basándose principalmente en la determinación de algunos parámetros geométricos de la huella como: Área, eje mayor, eje Menor [24,28] así como también de parámetros ópticos de la huella [29,30], encontrándose una buena correlación de los parámetros de las huellas, tanto geométricos como ópticos, y las energías de las partículas alfa que las forman para el caso de incidencia normal. Sin embargo los valores de energías estudiados están restringidos a ciertos rangos reducidos de energía por ejemplo en la referencia [25] se estudia valores energías menores de 1.47 MeV, en las referencias [26,27] se estudian el rango de energías entre 3.2 y 5.8 MeV, en [28] de 5.3 a 8.7 MeV y en [29,30] de 1.5 a 4.5 MeV.

En este trabajo de tesis se aborda el problema de clasificación automática de imágenes de huellas nucleares alfa grabadas debido a partículas alfa con diferentes energías de incidencia e incidencia normal que cubren un amplio rango de energías entre 0.15 y 8 MeV cubriendo gran parte de las energías de las partículas alfa naturales, cuyo mayor valor proviene de la cadena del Torio (isótopo Po-212, de 8.78 MeV). Elaboraremos una serie de algoritmos para determinar automáticamente ambos tipos de parámetros tanto geométricos como ópticos, adicionalmente introduciremos 3 nuevos parámetros que relacionan el nivel de gris de la huella y del entorno de esta; en total se determinarán 22 parámetros por cada huella. Las imágenes que se trabajarán en esta tesis forman parte de una amplia base de datos de imágenes obtenidas durante la realización de la tesis doctoral de A. Zuñiga, donde se estudió las principales características dejadas por las partículas alfa al incidir sobre el detector de huellas nucleares de estado sólido CR-39 “Polyallyl diglycol Carbonate” (PADC) [31].

El polímero detector CR-39 se obtiene mediante la polimerización del monómero  $C_{12}H_{18}O_7$ . Como resultado del proceso de polimerización, se obtiene un material de gran homogeneidad e isotropía, formado por largas cadenas, enlazadas entre sí, que dan lugar a una estructura pseudocristalina, característica de los plásticos denominados “thermoset”. El CR-39 es además extremadamente quebradizo, ópticamente muy puro y resistente a casi todos los agentes químicos, exceptuando las bases fuertes y los ácidos altamente oxidantes, en contacto con los cuales se descompone sin diluirse. Su bajo umbral de registro permite la detección de partículas ligeras (alfa, protones) incluso de energía elevada, todas estas propiedades hace del CR-39 el polímero detector más exitoso y ampliamente usado en el mundo [32,33].

En lo referente a desarrollo de software, existe en la actualidad una gran cantidad de sistemas que permiten contar automáticamente el número de huellas por unidad de área [34,40], existen programas que permiten simular la apariencia geométrica [41] y óptica [42] de las huellas a partir de datos como la energía de incidencia, ángulo de incidencia, velocidad de ataque en la huella, *función V*, ingresados por el usuario. Sin embargo no existe, software alguno (ni comercial ni libre) que permita clasificar automáticamente las imágenes de huellas nucleares alfa, de acuerdo a la energía de las partículas alfa incidente que las forman. Con esto en mente nos planteamos como objetivo principal de esta tesis: Desarrollar una herramienta computacional que realice de manera automática el conteo y la clasificación de huellas nucleares alfa de incidencia normal sobre el polímero detector CR-39 de acuerdo a la energía de las partículas alfa incidentes que las

originan, mediante la implementación de técnicas de procesamiento digital de imágenes y de Redes neuronales artificiales. Con la finalidad de lograr nuestro objetivo cuatro tareas fundamentales se desarrollaran a lo largo de esta tesis: (1) Investigar, identificar y desarrollar técnicas ó algoritmos de procesamiento digital de imágenes que permitan la detección, conteo y extracción automática de parámetros de imágenes de huellas nucleares alfa, (2) Diseñar e implementar un modelo de red neuronal que realice eficientemente la clasificación de imágenes de huellas nucleares alfa, y permita la elección de los valores más adecuados para los parámetros utilizados por la red, (3) Evaluar el modelo de red neuronal propuesto, respecto a parámetros de confiabilidad y eficiencia; y finalmente (4) Desarrollar un software que implemente los algoritmos de procesamiento digital desarrollados así como también el modelo de red neuronal propuesto.

## 2. MARCO TEÓRICO.

En este capítulo se dan los fundamentos necesarios para poder entender y dar solución al problema planteado. Cuatro grandes temas son cubiertos en este capítulo en sus distintas secciones. En la sección 2.1 se aborda el tema de la interacción de las partículas cargadas con la materia, la sección 2.2 trata de los detectores de huellas nucleares de estado sólido, la sección 2.3 del procesamiento digital de imágenes y finalmente en la sección 2.4 es para las redes neuronales artificiales.

### 2.1. Interacción de partículas cargadas con la materia

Los efectos provocados por el paso de un ión a través de un sólido dependen principalmente de la masa del ión  $M_1$ , su número atómico  $Z_1$  y su energía cinética  $E_1$  así como también de la masa  $M_2$  y número atómico  $Z_2$  del blanco. En la figura 2.1 se muestra los tres fenómenos diferentes que pueden ocurrir:

1. *Efectos en el proyectil.* Un efecto importante es el de pérdida de energía del ión incidente. El mismo presenta cambios en su estado de carga (ya sea por perder o ganar electrones con el medio) y cambios en su dirección original debido a las colisiones con los átomos del material que atraviesa.
2. *Emisión de radiaciones.* Al pasar el ión por el sólido se producen radiaciones secundarias de muy baja energía que son difíciles de detectar.
3. *Efectos internos.* Son los procesos internos que ocurren en la muestra. No son directamente medibles y están asociados a la radiación secundaria.

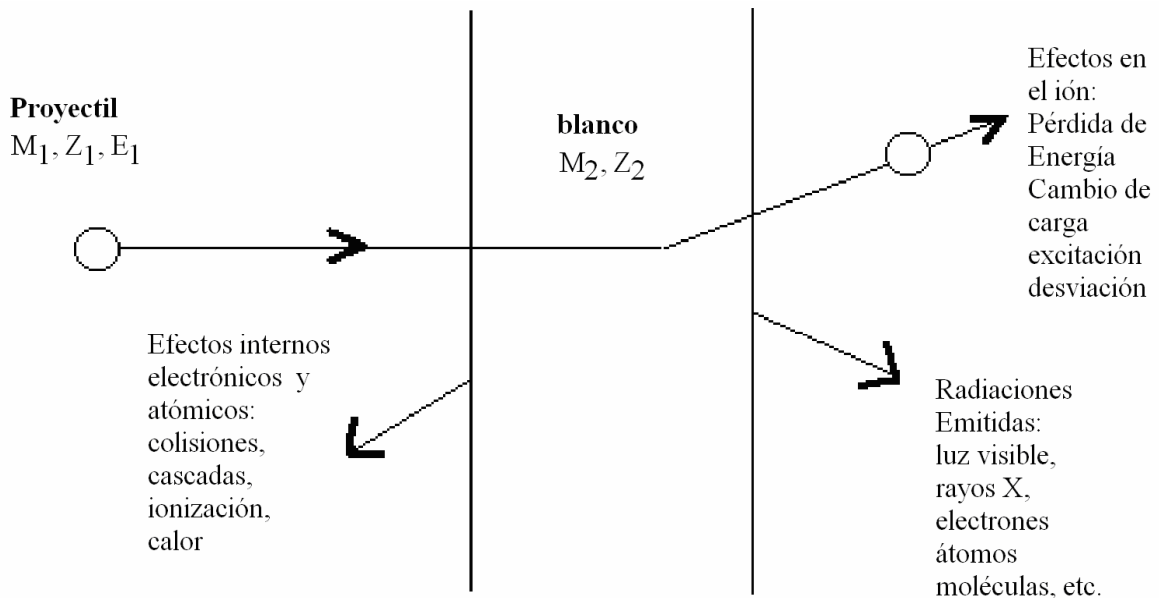


Fig. 2.1. Efectos observados en la interacción de un ión con un sólido

En general cuando una partícula cargada pasa a través de la materia pierde energía y su trayectoria cambia respecto a su dirección original, básicamente debido a dos

procesos:

1. Colisiones inelásticas con los electrones atómicos del material.
2. Dispersión elástica debido a la interacción con los núcleos de la muestra.

Estas reacciones ocurren muchas veces y es el resultado acumulado lo que genera los efectos observados.

De los dos procesos electromagnéticos mencionados, las colisiones inelásticas son las responsables de la mayor parte de la pérdida de energía de las partículas cargadas que atraviesan la materia. En este tipo de colisiones ( $\sigma \sim 10^{-17} - 10^{-16} \text{ cm}^2$ ) la energía es transferida de la partícula a los electrones ligados causando la ionización o excitación de los átomos. La cantidad transferida en cada colisión individual es en general muy pequeña, pero las colisiones son tan numerosas que el efecto puede ser observado aún en materiales delgados. Por ejemplo, un protón a 10 MeV pierde la totalidad de su energía en sólo 0.25 mm de cobre.

Estas colisiones suelen dividirse en dos grandes grupos: 1) las llamadas “suaves”, en las que el átomo se excita y 2) “duras”, en las que la energía transferida es suficiente como para ionizar el átomo. En algunas de las colisiones duras, la energía transferida es tal que el electrón puede causar ionizaciones secundarias. Estos electrones energéticos son denominados electrones  $\delta$  [43].

La dispersión elástica de la partícula incidente debido a la interacción con los núcleos del material también puede ocurrir, pero no tan frecuentemente como las colisiones electrónicas. En general muy poca energía es transferida en este proceso. Cuando la energía del ión disminuye, en forma considerable al final de su recorrido, predomina la dispersión nuclear produciendo una cascada de colisiones y gran cantidad de desplazamientos atómicos.

### 2.1.1. Poder de frenado

La energía perdida por una partícula cargada por unidad de camino recorrido se define como el poder de frenado (stopping power) y se representa con la función  $S(E)$ :

$$S(E) = -\frac{dE}{dx} \quad (2.1)$$

En su camino a través del sólido el ión pierde paulatinamente su energía y el poder de frenado varía continuamente, ya que depende fuertemente de la energía del ión incidente. Debido a que cambia en varios ordenes de magnitud, su valor se representa generalmente en escala doble logarítmica. Como se puede observar en la figura 2.2, el poder de frenado se puede dividir en dos regiones separadas por un máximo, localizado aproximadamente en 1 MeV/uma, denominado pico de Bragg. A mayores energías la función decrece como  $1/E$ , es el rango descrito por la teoría de Bethe-Bloch (BB) [44,45] y para energías menores al pico de Bragg decrece aproximadamente como  $\sqrt{E}$ , es el rango descrito por la teoría de Lindhard-Scharff-Schiott (LSS) [46].

La teoría BB se aplica para altas energías y velocidades no relativistas, en forma más precisa cuando la velocidad del proyectil es  $V_0 \gg v_0 Z_I^{2/3}$ , donde  $v_0 = 2.2 \times 10^8 \text{ cm s}^{-1}$  es la velocidad de Bohr y  $Z_I$  es el número atómico del ión incidente. Esta teoría es válida hasta que la velocidad se aproxima a  $v_0 Z_I^{2/3}$ . A menores velocidades se reduce paulatinamente la carga efectiva, el ión se neutraliza y el poder de frenado electrónico comienza a ser proporcional a la velocidad del ión incidente (LSS). Ambas teorías describen adecuadamente la conducta general del poder de frenado teniendo en cuenta la dependencia con la energía y su magnitud.

La región de transición, alrededor del máximo del poder de frenado solo se describe empíricamente. A muy bajas energías del ión, algunos keV, justo antes del final del camino del ión prevalece el poder de frenado nuclear.

Como los fenómenos de colisiones nucleares y electrónicas se pueden considerar independientes entre sí, el poder de frenado puede expresarse como la suma de los dos términos. Uno proveniente de las colisiones electrónicas y el otro de las nucleares:

$$\frac{dE}{dx} = \frac{dE}{dx} \Big|_{\text{nuclear}} + \frac{dE}{dx} \Big|_{\text{electrónico}} \quad (2.2)$$

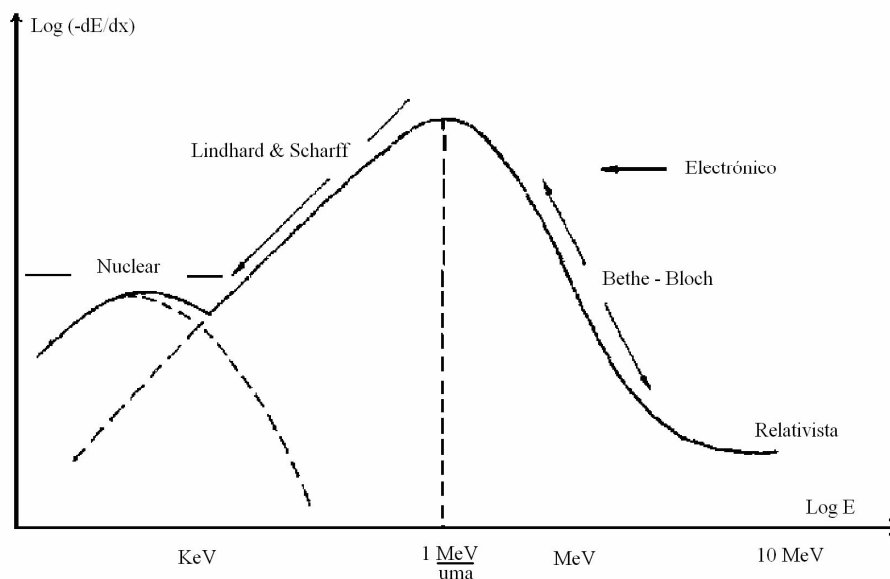


Fig. 2.2. Gráfico doble logarítmico del poder de frenado en función de la energía donde se muestran diferentes zonas que corresponden a distintos mecanismos [47].

### 2.1.1.1. Poder de frenado electrónico

Desde el punto de vista de la interacción con los electrones la pérdida de energía del ión en la materia se puede dividir en tres fases:

*Energía transferida:* Durante la primera fase la energía es transferida del ión a los electrones individuales del material. La energía perdida por el ión y el desvío producido pueden ser determinados a partir de las mediciones del ángulo y la energía antes y después

de atravesar la muestra.

*Cascadas de colisiones electrónicas:* La segunda fase es la emisión de electrones (electrones  $\delta$ ). Estos transportan la energía en forma aproximadamente perpendicular a la dirección del ión y hasta distancias del orden del micrón.

*Cascadas de colisiones atómicas:* Al final del recorrido predomina la interacción atómica y se forman cascadas de colisiones. Este tipo de cascada tiene una dimensión mucho menor que la zona afectada por las electrones  $\delta$ , y es de aproximadamente  $0,01 \mu m$ .

### 2.1.1.2. Poder de frenado nuclear

A medida que disminuye la energía del proyectil las nubes electrónicas que rodean los núcleos del blanco se hacen más difíciles de penetrar por el ión incidente. En un caso límite el proceso de impacto se puede representar por la colisión de dos esferas rígidas de masa comparable. Si se incrementa la energía a valores mayores que  $0,1 \text{ MeV/uma}$ , las nubes electrónicas de ambos átomos se interpenetran y se puede observar un fenómeno del tipo de una dispersión de Rutherford pura. Las colisiones atómicas permiten una transferencia directa de momento y energía del proyectil al átomo de la muestra.

### 2.1.2. La fórmula de Bethe-Bloch

Un tratamiento mecanocuántico fue realizado por primera vez por Bethe y Bloch para obtener una expresión del poder de frenado. En este cálculo la transferencia de energía se parametriza en términos de una transferencia de cantidad de movimiento en lugar de hacerlo mediante el parámetro de impacto, como en el cálculo de Bohr. Esto es más realista pues la transferencia de la cantidad de movimiento es una cantidad medible mientras que el parámetro de impacto no lo es. La fórmula así obtenida es:

$$-\frac{dE}{dx} = 2\pi N_a r_e^2 m_e c^2 \rho \frac{Z_2}{A_2} \frac{Z_1}{\beta^2} \left[ \ln \left( \frac{2m_e \gamma^2 v^2 w_{\max}}{I^2} \right) - 2\beta - \delta - 2\frac{C}{Z_2} \right] \quad (2.3)$$

Donde:

$$2\pi N_a r_e^2 m_e c^2 = 0.1535 \text{ MeV cm}^2/\text{g}$$

$$r_e: \text{ radio clásico del electrón} = 2.817 \times 10^{-13} \text{ cm.}$$

$m_e$ : masa del electrón.

$$N_a: \text{ número de Avogadro} = 6,022 \times 10^{23} \text{ mol}^{-1}.$$

I: potencial de excitación promedio.

$Z_2$ : número atómico del material absorbente.

$A_2$ : número másico del material absorbente.

$\rho$ : densidad del material absorbente.

$Z_1$ : carga de la partícula incidente.

$\beta = v/c$  de la partícula incidente.

$\gamma = 1/\sqrt{1-\beta^2}$ .

$\delta$ : corrección de densidad.

$C$ : corrección de capa.

$W_{max}$ : energía máxima transferida en una colisión.

La energía máxima transferida ocurre cuando la colisión es frontal. Para una partícula incidente de masa  $M$ :

$$W_{max} = \frac{2m_e c^2 \eta^2}{1 + 2s\sqrt{1 + \eta^2 + s^2}} \quad (2.4)$$

Donde  $s = m_e/M$  y  $\eta = \beta\gamma$ . Si  $M \gg m_e$  entonces

$$W_{max} \approx 2 m_e c^2 \eta^2 \quad (2.5)$$

El potencial de excitación promedio  $I$  es el parámetro principal de la fórmula de BB. Es esencialmente la frecuencia orbital promedio de los electrones ligados a los átomos del material absorbente multiplicada por la constante de Planck. En la práctica es una cantidad muy difícil de calcular, por lo tanto se utilizan fórmulas semiempíricas

$$\frac{I}{Z_2} = \begin{cases} 12 + 7Z_2^{-1} & Z_2 < 13 \\ 9.76 + 58.8Z_2^{-1.19} & Z_2 \geq 13 \end{cases} \quad (2.6)$$

Las cantidades  $\delta$  y  $C$  son correcciones importantes a altas y bajas energías respectivamente. La corrección de densidad  $\delta$  surge debido a que el campo eléctrico del ión polariza los átomos a su paso disminuyendo así la interacción con los electrones más lejanos. Este efecto toma importancia con la creciente energía del ión y depende de la densidad del material pues la polarización inducida es mayor. La corrección de capa  $C$  tiene en cuenta los efectos que aparecen cuando la velocidad del ión es menor o igual que la velocidad orbital de los electrones ligados. Esto es debido a que la suposición de que el electrón es estacionario respecto a la partícula incidente deja de ser válida.

### 2.1.3. La curva de Bragg

Se le llama así a la curva obtenida de la representación de la pérdida de energía de partículas cargadas que penetran en un material (Fig. 2.3). Esta curva es prácticamente plana (ligeramente ascendente) al principio, donde la partícula tiene aún una alta velocidad

y está perdiendo energía de forma más o menos constante, y después alcanza un valor máximo, conocido como “*pico de Bragg*” donde el ión está ya muy ralentizado, por lo que empieza a recoger carga y la curva decae. Esta caída puede ser más o menos abrupta, dependiendo del número atómico del ión. Así, las partículas con el número mayor de cargas nucleares empiezan a recoger electrones más temprano en su proceso de desaceleración.

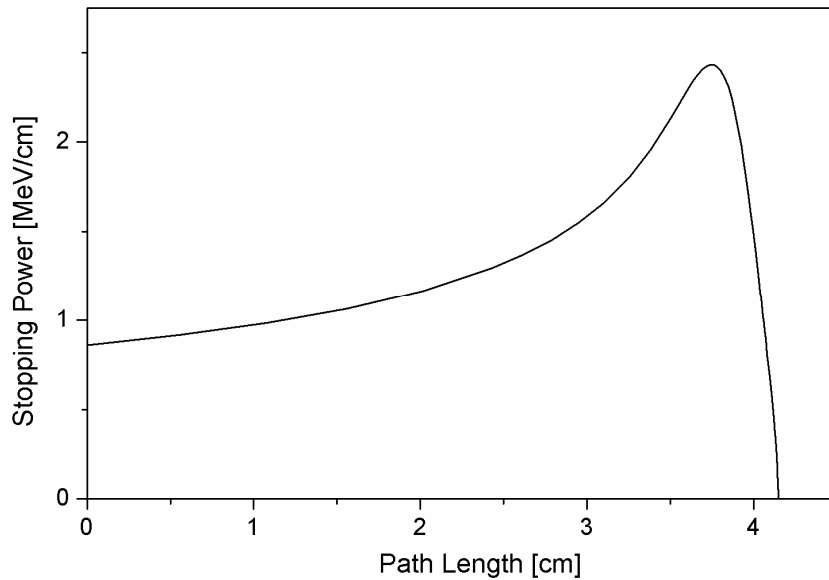


Fig. 2.3. Representación de la curva de Bragg en un gráfico de pérdida de energía frente a la distancia de penetración de partículas alfa de 5.49 MeV en el aire [43].

### 2.1.4. Ley de Bragg

Un valor aproximado del poder de frenado en el caso de tener más de un elemento se puede obtener promediando  $dE/dx$  para cada elemento en el compuesto pesado por la fracción de electrones de cada uno:

$$\frac{1}{\rho} \frac{dE}{dx} = \frac{\omega_1}{\rho_1} \left( \frac{dE}{dx} \right)_1 + \frac{\omega_2}{\rho_2} \left( \frac{dE}{dx} \right)_2 + \dots \quad (2.7)$$

Donde  $w$  es la fracción en peso de los elementos del compuesto

$$\omega_i = \frac{a_i A_i}{A_m} \quad (2.8)$$

$a_i$  es el número de átomos del elemento  $i$ -ésimo en la molécula y  $A_i$  su peso atómico.

$$A_m = \sum a_i A_i \quad (2.9)$$

### 2.1.5. Rango



Dado que las partículas cargadas pierden su energía cuando atraviesan la materia, podemos preguntarnos qué distancia recorrerán antes de perder toda su energía y detenerse. Si suponemos que la energía se pierde de manera continua entonces esta distancia es un cantidad bien definida y será la misma en el caso de partículas idénticas que posean la misma energía inicial y en el mismo material. Esta cantidad se denomina rango de la partícula y depende del tipo de material, de la partícula y de su energía.

En realidad la pérdida de energía no es un proceso continuo, sino que es de naturaleza estadística, ya que dos partículas idénticas con la misma energía no sufrirán exactamente el mismo número de colisiones en el material y por lo tanto no perderán la misma cantidad de energía. Tendremos así una distribución estadística de rangos. En primera aproximación esta distribución es Gaussiana.

Desde un punto de vista teórico, el rango medio de una partícula de energía  $E_0$  se podría calcular como:

$$R(E_0) = \int_0^{E_0} \left( \frac{dE}{dx} \right)^{-1} dE \quad (2.10)$$

Esta expresión no tiene en cuenta la dispersión múltiple de la partícula, lo cual hace que su trayectoria sea distinta a una línea recta. Por lo tanto, el valor calculado de esta forma es menor que la verdadera longitud del camino recorrido por la partícula dentro del material. Como hemos mencionado antes, el efecto de la dispersión múltiple es generalmente muy pequeño para iones pesados, y por lo tanto el camino recorrido puede ser considerado como una línea recta. En la práctica, se hace uso de expresiones semiempíricas de la forma:

$$R(E_0) = R_0(E_{min}) + \int_{E_{min}}^{E_0} \left( \frac{dE}{dx} \right)^{-1} dE \quad (2.11)$$

Donde  $E_{min}$  es el mínimo valor de la energía para la cual la expresión de  $dE/dx$  es válida y  $R_0(E_{min})$  es una constante determinada empíricamente que tiene en cuenta el comportamiento para bajas energías.

### 2.1.6. Dispersión estadística de energía (straggling)

El proceso de pérdida de energía de una partícula que atraviesa un medio material está sujeto a fluctuaciones estadísticas. Por lo tanto un haz de partículas monoenergético presentará una dispersión de energía luego de atravesar un espesor  $\Delta x$  de material homogéneo. Este fenómeno se conoce como dispersión estadística de energía (straggling) y determina un límite finito a la precisión con la cual tanto las pérdidas de energía como las profundidades en una muestra pueden ser determinadas.

### 2.1.7. Dispersión múltiple Coulombiana

Resumiendo, además de las colisiones inelásticas con los electrones atómicos, las partículas que atraviesan un material sufren repetidas dispersiones elásticas coulombianas debido a los núcleos presentes en la muestra, aunque con menor probabilidad que aquellas. Despreciando los efectos debidos al spin y el apantallamiento de la interacción, estas colisiones están gobernadas por la fórmula de Rutherford:

$$\frac{d\sigma}{d\Omega} = \left( \frac{Z_1 Z_2 e^2}{4E} \right)^2 \frac{1}{\sin^4(\theta/2)} \quad (2.12)$$

Esta expresión corresponde al sistema del centro de masa,  $E$  es la energía del proyectil antes de la colisión y  $\theta$  es el ángulo de dispersión. Si la energía la expresamos en MeV, es conveniente tomar  $e^2 = 1,44 \times 10^{13}$  MeV cm.

Debido a la dependencia funcional  $1/\sin^4(\theta/2)$  la mayoría de estas colisiones resultan en una pequeña deflexión angular de la partícula. Pero a medida que la partícula va penetrando se ve expuesta a muchas de estas interacciones y el efecto final neto es un desvío con respecto a su dirección original.

En general, el tratamiento de la dispersión coulombiana en la materia se divide en 3 regiones:

*Dispersión simple.* Si el material es lo suficientemente delgado como para que la probabilidad de sufrir más de una colisión sea muy pequeña, entonces la distribución angular estará dada simplemente por la ecuación anterior.

*Dispersión plural.* Si el número medio de dispersiones es  $N < 20$  entonces tenemos dispersión plural. Es el caso más difícil de tratar pues no hay un método simple que pueda ser aplicado para describirlo.

*Dispersión múltiple.* Si el número medio de dispersiones simples es  $N > 20$  y la pérdida de energía es baja o despreciable, este problema puede ser tratado estadísticamente para obtener una distribución de probabilidad para el ángulo neto de deflexión en función del espesor del material atravesado.

## 2.2. Detectores de huellas nucleares de estado sólido

Los detectores de huellas nucleares son básicamente materiales sólidos dieléctricos en donde el paso de partículas cargadas, crean a lo largo de su trayectoria, zonas de daño a escala atómica llamadas huellas latentes. Un eventual tratamiento químico o electroquímico de esas huellas latentes permite su visualización bajo un microscopio óptico. La ciencia de los detectores de huellas nucleares de estado sólido nació en 1958 cuando D.A Young [1] observó las primeras huellas en un cristal de LiF. Estas cavidades posteriormente llamados "huellas", fueron encontrados en un cristal de LiF que fue colocado en contacto con una hoja de uranio, el cual fue previamente irradiado con neutrones lentos, posteriormente el cristal de LiF fue tratado con una solución químicamente agresiva, los neutrones térmicos llevaron a la fisión de los núcleos de uranio y los fragmentos de fisión bombardearon el cristal dañando ciertas regiones de este.

Las regiones dañadas constituyeron zonas químicamente más activas que las áreas indemnes circundantes. Un año después, Silk y Barnes reportaron haber encontrado regiones dañadas en mica; utilizando para esto el microscopio electrónico de transmisión. Fleisher et al. [7] realizaron diversas investigaciones de huellas en diversos materiales como minerales, plásticos y vidrios.

Los polímeros detectores más comúnmente usados son: el detector CR-39, Cartwright et al [31], basado en el carbonato de polyallyldiglycol, el detector conocido bajo nombre comercial LR-115 basado en nitrato de celulosa y el detector de Makrofol que se basa en el policarbonato de bisphenol-A. Los dos primeros detectores son más sensitivos que el detector de Makrofol, sin embargo el detector LR-115, el cual tiene una capa activa limitada de aproximadamente 13  $\mu\text{m}$ , no permite detectar partículas alfa con energías mayores de 4 MeV, además este detector tiene el inconveniente de tener un complicado conteo de huellas cuando la duración de la exposición es de más de 4 semanas.

### **2.2.1. Mecanismo de formación de una huella**

La operación de un detector de huellas nucleares de estado sólido se basa en los efectos producidos por la interacción de la partícula cargada con el material del detector cuando la partícula cargada pasa a través de él. El principal proceso de interacción es la ionización de las moléculas del material. Por ejemplo, una partícula alfa con la energía de 6 MeV crea cerca de 150 000 pares de iones en nitrato de celulosa. Puesto que el rango de una partícula alfa de 6 MeV en este material es de cerca de 40  $\mu\text{m}$ , un promedio de 3700 pares de iones se crean por micrómetro, o entre 3 y 4 pares de iones por nanómetro. Una partícula alfa ioniza casi todas las moléculas cerca de su trayectoria. Este proceso de ionización primario dispara una serie de nuevos procesos químicos que dan lugar a la creación de los radicales químicos libres y de otras especies químicas. En la figura 2.4 se muestra una representación esquemática de la ruptura de cadenas causada por el paso de una partícula alfa en un polímero. A lo largo de la trayectoria de la partícula alfa, una zona enriquecida con radicales químicos libres y otras especies químicas es creada. Esta zona de daño es llamada huella latente.

Los procesos primarios de interacción de la partícula cargada incidente con el detector son la ionización y la excitación de las moléculas del detector la partícula inicial cargada pierde su energía a través de muchos procesos de interacción. Teóricamente, este interactúa a través de las fuerzas de Coulomb, con las partículas cargadas (electrones y núcleos) en el material. La mayoría de las interacciones ocurre con los electrones y solamente una pequeña cantidad de interacciones son con los núcleos. Ya que la partícula cargada pesada inicial (solamente tales partículas pueden producir huellas) es mucho más pesada que los electrones, la dirección de la partícula no cambia efectivamente y la trayectoria es casi totalmente una línea recta. Esto puede no ser verdad si la partícula interactúa con un núcleo, donde una desviación significativa de la dirección inicial puede ocurrir. Sin embargo, tales interacciones son relativamente raras. Algunas desviaciones de la línea recta pueden suceder cerca del fin de la trayectoria de la partícula, donde la energía de la partícula llega a ser muy baja.

La pérdida de energía es generalmente muy pequeña cuando es comparada con la energía de la partícula. Por ejemplo, la ionización de una molécula en aire en promedio necesita cerca de 32 eV, que es  $10^{-5}$  a  $10^{-6}$  veces la energía de la partícula (si se asume que la energía de la partícula es del orden de MeV). Como resultado de muchos pequeños

procesos de interacción, la partícula pierde su energía continuamente en el material del detector con una tasa lineal de pérdida de energía,  $dE/dx$ . Esta cantidad física recibe el nombre de poder de frenado y es función de las características de la partícula (masa, carga y energía) y del material del detector usado.

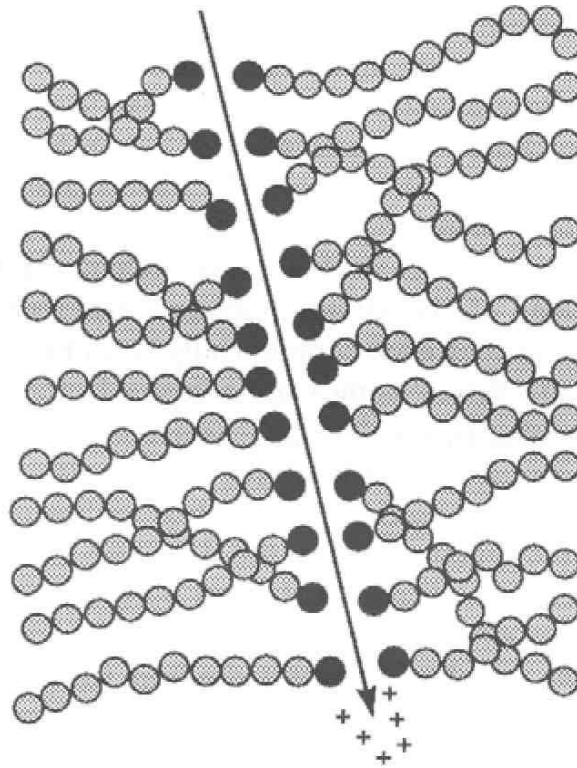


Fig. 2.4. Representación esquemática de la ruptura de cadenas causada por el paso de una partícula alfa en un polímero [2].

### 2.2.2. Ataque químico

Si un pedazo de material que contiene huellas latentes se expone a una cierta solución químicamente agresiva, las reacciones químicas serían más intensivas a lo largo de las huellas latentes. Las soluciones acuosas del NaOH o de KOH son las soluciones químicas más frecuentemente usadas con este propósito. El efecto total es que la solución química graba la superficie del material detector, pero con una tasa más rápida en la región dañada. De esta manera se forma una huella de la partícula, que se puede observar con un microscopio óptico. Este procedimiento se llama ataque químico del detector o visualización de la huella, y el efecto en sí es llamado "Track effect" o efecto huella.

El efecto huella existe en muchos materiales. Es particularmente pronunciado en materiales con las moléculas largas, como los nitratos de celulosa o diversos policarbonatos, tales materiales son los más convenientes para la fabricación de un detector. El efecto es visto también en algunos materiales amorfos como los vidrios. Sin embargo, solamente los materiales dieléctricos demuestran el efecto huella. En materiales conductores y en semiconductores, el proceso de la recombinación ocurre y las huellas latentes no son estables. Un estudio exhaustivo de los materiales que demuestran el efecto huella es dado por Fleischer et al en la referencia [3].

Según lo mencionado anteriormente, el ataque químico de materiales conductores irradiados no produciría huellas visibles. La condición para la formación estable de la huella latente se expresa a veces como un valor límite para la resistividad del material. Sin embargo, no hay un valor único de la resistividad para la cual el efecto huella aparezca siempre. De esta manera, la resistividad del material no puede servir como el único criterio para la formación de la huella.

Aunque el efecto huella sea relativamente bien conocido, y la técnica es algo simple y directa, no hay una teoría única que explique la formación de las huellas. Los procesos físicos básicos a partir de los cuales la partícula cargada inicial pierde su energía son la ionización y la excitación de las moléculas del material. Esta primera fase “física” en la cual la partícula inicial entrega su energía a los átomos que rodean su trayectoria es de muy corta duración; la detención de la partícula ocurre dentro de un periodo del orden de los picosegundos. Los electrones libres creados en estas interacciones primarias pierden su energía, nuevamente, con una serie de ionizaciones y de excitaciones, y crearán cada vez más electrones libres. Una gran cantidad de electrones libres y de moléculas dañadas se crean cerca de la huella de la partícula.

En la segunda fase fisicoquímica, nuevas especies químicas son creadas por interacciones de las moléculas dañadas. Durante el ataque químico, las interacciones de estas nuevas especies químicas con la solución serán más fuerte que con el material no dañado del detector. Sin embargo, no se sabe qué especies químicas se forman después del paso de la partícula a través del material, y la naturaleza del daño tampoco es bien conocida.

### **2.2.3. Geometría de la formación de una huella**

Uno de los desafíos que ha atraído cantidades significativas de atención es el desarrollo de una descripción formal de la formación de una huella. Hay diversas teorías que describen el aspecto físico de la formación de huella. El problema es de naturaleza geométrica. Sin embargo, hasta ahora, no hay una sola teoría completa que pueda explicar satisfactoriamente la formación de la huella y permita calcular los parámetros relacionados con esta. La geometría de formación de una huella ha sido considerada por un gran número de autores [48-54], una revisión completa de los modelos de formación de una huella se puede encontrar en la referencia [55].

La formación de una huella es análoga a la propagación de una onda. Según el principio del Huygens, ver figura 2.5, cada punto en el frente de onda es la fuente de una nueva onda esférica. En el caso de una huella, un hemisferio con un radio  $h = V_b t$  ( $t$  es el tiempo del ataque químico) se forma alrededor de cada punto en la superficie del detector, excepto en la dirección de la trayectoria de la partícula adonde el ataque progresa con la velocidad el  $V_t$ .

La formación de una huella es gobernado por el cociente  $V = V_t/V_b$ , la formación de la huella no es posible si  $V$  es más pequeño que o igual a 1. Es decir la condición  $V > 1$  se debe satisfacer para formar una huella.

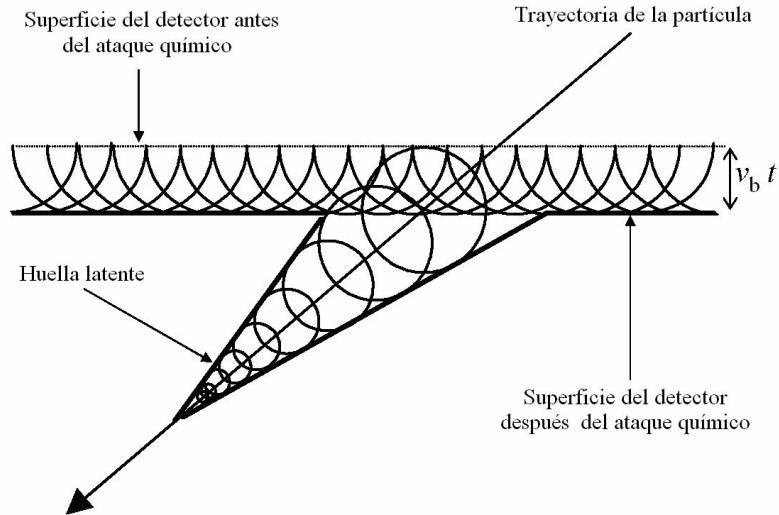


Fig. 2.5. Aplicación de principio Huygens para explicar la formación de una huella latente en un polímero detector [2].

### 2.2.3.1. Formación de una huella con $V_t$ constante e incidencia normal

El caso más simple de formación de una huella es cuando la partícula cargada incide perpendicularmente a la superficie del detector y con una tasa de grabado en la huella constante, según se muestra en la figura 2.6, I es la superficie inicial del detector, I' es la superficie después del ataque químico,  $V_t$  es la velocidad de grabado de la huella a lo largo de la trayectoria de la partícula,  $V_b$  es la velocidad de grabado de las regiones indemnes del detector, O es el punto de la entrada, E es la punto final de una partícula en el material del detector y  $OE = R$  es el rango de la partícula en el material del detector. La distancia entre I e I' es igual a h, es decir, el espesor de la capa removida durante el ataque químico,  $L'$  es la distancia total recorrida por la solución química atacante a lo largo de la trayectoria de la partícula, y L es la profundidad de la huella.

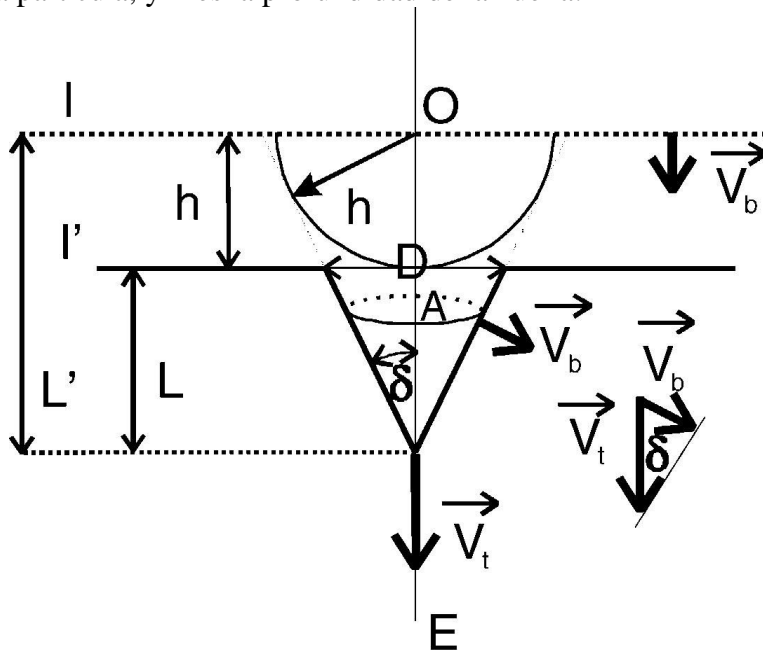


Fig. 2.6. Geometría de formación de una huella. El Angulo de incidencia es normal con respecto a la superficie del detector y  $V_t$  es constante [55].

De la semejanza de triángulos en la figura. 2.6, se obtiene que:

$$\sin \delta = \frac{1}{V} \quad (2.13)$$

Según lo mencionado anteriormente, el ataque químico progresa en todas las direcciones con la taza  $V_b$  excepto a lo largo de la trayectoria de la partícula donde el ataque químico va con una taza  $V_t$ . En tres dimensiones, la huella es un cono con un ángulo de desarrollo  $\delta$ , que es obtenido por la rotación de las paredes de la huella alrededor de la trayectoria de la partícula. El círculo A en figura 2.6 representa la revolución de un punto de la pared de la huella alrededor de la trayectoria de la partícula.

La sección transversal de la superficie post tratada  $L'$  y el área de la base del cono es un círculo de diámetro  $D$ .

De la figura se tiene que profundidad de la huella esta dado por:

$$L = (V_t - V_b)t \quad (2.14)$$

Donde  $t$  es el tiempo del ataque. Además tenemos que:

$$\tan \delta = \frac{D/2}{L} = \frac{h}{\sqrt{L^2 - h^2}} \quad (2.15)$$

Combinando las ecuaciones anteriores encontramos que el diámetro de apertura del la huella es:

$$D = 2h \sqrt{\frac{V-1}{V+1}} \quad (2.16)$$

Si  $V \gg 1$ , de la ecuación anterior tenemos:

$$D \cong 2h \quad (2.17)$$

Basado en la ecuación anterior un método indirecto para la medición de la velocidad de grabado en la región no dañada,  $V_b$ , fue desarrollado. Si la velocidad de grabado de la huella es muy grande, el cual es el caso cuando iones pesados o producto de fisión son usados en la irradiación, la capa removida esta directamente relacionada al diámetro de apertura de la huella el cual es fácilmente medible y dado que  $h = V_b t$ ; es por lo tanto fácil de encontrar  $V_b$ .

Durante el ataque químico, la solución agresiva progresa hacia el punto final E de la trayectoria de la partícula. El extremo de la huella es agudo y la huella es completamente cónica. Sin embargo, en el tiempo  $T_0$  la solución atacante alcanza el punto final E de la trayectoria de la partícula. La superficie del detector en el tiempo  $T_0$  es denotado como superficie 1 en la figura 2.7, la solución atacante después de ese momento

progresa en todas las direcciones con la misma tasa  $V_b$ , y la correspondiente huella se torna una huella sobre atacada. Una esfera es ahora formada alrededor del punto E, y la forma de la huella ha cambiado a la de un cono unido a una esfera (superficie 2 y 3 de la figura). Con un tiempo más prolongado de ataque químico, se agranda la parte esférica y la parte cónica se torna cada vez más pequeña, finalmente con un tiempo suficientemente largo, toda la huella torna esférica (superficie 4 de la figura 2.7). Con una huella esférica el contraste se pierde pudiendo verse con dificultad e incluso tornarse invisible.

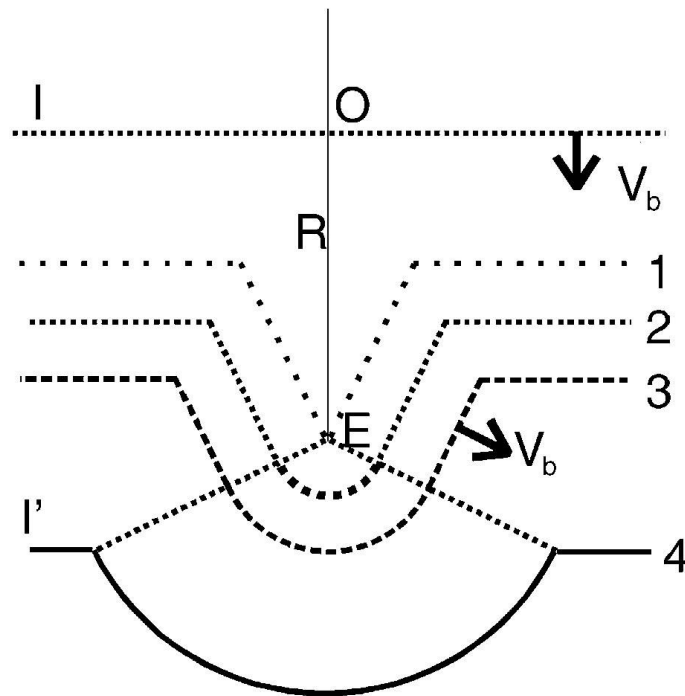


Fig. 2.7. Tres fases en el crecimiento de una huella. I es la superficie inicial del detector, O y E son el punto de entrada y final de la trayectoria de la partícula, R es el rango de la partícula en el material del detector, y  $V_b$  es la velocidad de ataque en la región no dañada. (1) huella cónica; (2 y 3) huella parcialmente cónica y parcialmente esférica; (4) huella totalmente esférica [55].

## 2.3. Procesamiento digital de imágenes

El procesamiento digital de imágenes tiene por objetivo dos aspectos principales de aplicación: de un lado se encuentra el mejorar la información presente en una imagen para la interpretación humana y del otro lado el procesamiento de dicha información para el reconocimiento independiente por un sistema de diagnóstico asistido por Computadora. Las técnicas de procesamiento de imágenes son aplicadas cuando resulta necesario realzar o modificar una imagen para mejorar su apariencia, destacar algún aspecto de la información contenida en la misma o cuando se requiere medir, contrastar o clasificar algún elemento contenido en la imagen. Por otra parte, también son utilizadas cuando es necesario combinar imágenes completas, porciones de ellas o para reorganizar su contenido.

### 2.3.1. Representación Digital de una Imagen

Las imágenes digitales representan información visual asociada a una escena



ambiental real, que correspondería a lo que observamos con nuestro sentido de la vista, de igual manera, representan información no visible pero que puede ser medida mediante el uso de dispositivos tales como: los rayos X, ultrasonidos, radiación infrarroja, etc. El proceso de adquisición de la imagen requiere de un dispositivo apropiado para detectar el tipo de información visual del objeto de estudio y convertirla en una señal eléctrica que posteriormente se convierte en datos binarios, los cuales, se pueden almacenar o procesar a través de una computadora.

Por lo que, una imagen digital puede considerarse como una matriz cuyos índices de fila y columna identifican un punto de la imagen y el valor del correspondiente elemento de la matriz indica el nivel de gris en ese punto, ver figura 2.10, que podría ser denotada como  $f(x, y)$ , donde cada punto se denomina píxel y tiene asociadas las coordenadas espaciales definidas por  $x$  e  $y$  [56].

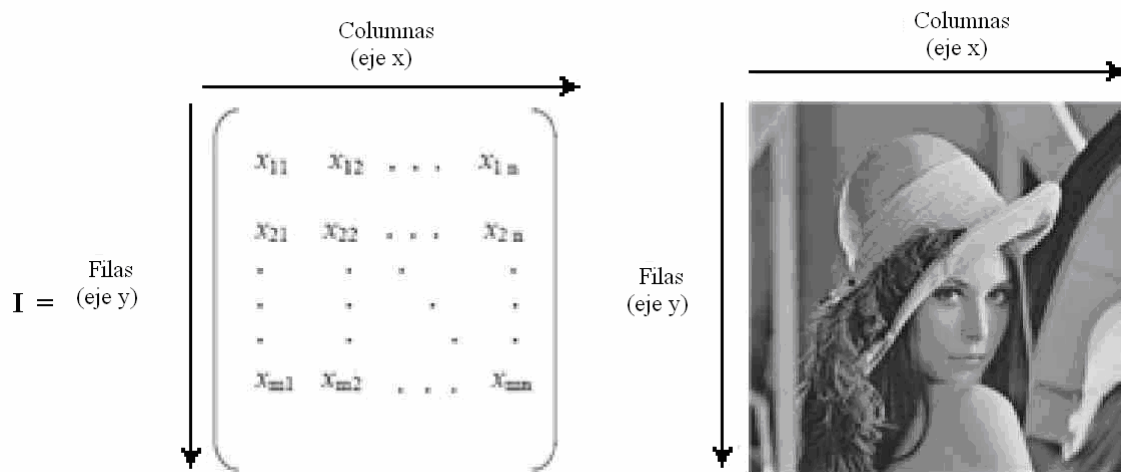


Fig. 2.8. Representación de una imagen digital [56].

La figura 2.8. Muestra la representación de una imagen digital, cuyo tamaño es de  $M \times N$  píxeles donde  $M$  corresponde al largo de la imagen y  $N$  al ancho de la misma.

### 2.3.2. Técnicas de procesamiento digital de Imágenes (PDI)

Existen básicamente dos tipos de técnicas:

- Técnicas en el dominio espacial.
- Técnicas en el dominio de la frecuencia.

El dominio espacial se refiere al propio plano de la imagen, y las técnicas de esta categoría se basan en la manipulación directa de los píxel de la imagen. El procesamiento en el dominio de la frecuencia se basa en la modificación de la transformada de Fourier de una imagen.

#### 2.3.2.1. Técnicas en el dominio espacial

Las funciones de procesamiento de la imagen en el dominio espacial pueden expresarse como:

$$g(x, y) = T[f(x, y)] \quad (2.18)$$

Donde  $g(x, y)$  es la imagen de salida,  $f(x, y)$  es la imagen procesada y  $T$  es un operador que actúa sobre  $f$ .  $T$  puede también operar sobre varias imágenes al mismo tiempo, como cuando se hace la suma píxel por píxel de  $M$  imágenes para reducir el ruido. Dentro de los métodos de dominio espacial puede enfocarse 2 perspectivas:

- Alteración píxel a píxel de los datos en una escala global (individuales).
- Operaciones basadas en múltiples puntos (vecindad).

Las operaciones individuales implican la generación de una nueva imagen modificando el valor del píxel  $(x, y)$  basándose en una regla global aplicada a cada píxel de la imagen original. El proceso consiste en obtener el valor del píxel  $(x, y)$  en la imagen, modificándolo por una operación lineal o no lineal y colocando el valor del nuevo píxel en la correspondiente localización dentro de la nueva imagen.

Las operaciones de vecindad utilizan el mismo procedimiento excepto que el nuevo valor del píxel en la imagen de salida, depende de una combinación de los valores de los píxeles en la vecindad del píxel de la imagen original que esta siendo transformada. Para definir el entorno alrededor del píxel  $(x, y)$  se emplea un área de la imagen cuadrada o rectangular centrada en  $(x, y)$  llamada subimagen. El centro de la subimagen se mueve píxel a píxel comenzando, por ejemplo, en la esquina superior izquierda y aplicando el operador en cada posición  $(x, y)$  para obtener  $g(x, y)$ .

### 2.3.2.1. 1. Operador Identidad

Este operador crea una imagen de salida idéntica a la original. La función de transformación es:

$$g(x, y) = f(x, y) \quad (2.19)$$

### 2.3.2.1.2. Operador inverso o negativo

Este operador crea una imagen de salida que es la inversa de la original. Para una imagen con valores de gris en el rango de 0-255 la función de transformación es:

$$g(x, y) = 255 - f(x, y) \quad (2.20)$$

### 2.3.2.1.3. Alteración global de brillo.

Se logra sumando una constante al valor de niveles de gris de cada píxel:

$$g(x, y) = f(x, y) + K \quad (2.21)$$

Para una imagen con valores de gris en el rango de 0-255, si el nivel de gris excede de 255, se deja el nivel en 255.  $K$  es el brillo que se desea aumentar o disminuir.

#### 2.3.2.1.4. Binarizado

Esta transformación crea una imagen de salida binaria a partir de los niveles de gris de la imagen original, los niveles que están sobre  $p_1$  se hacen iguales a 255 (blanco), y aquellos que están por debajo se hacen cero (negro).

$$g(x, y) = \begin{cases} 255 & \text{si } f(x, y) > p_1 \\ 0 & \text{si } f(x, y) < p_1 \end{cases} \quad (2.22)$$

#### 2.3.2.1.5. Histograma

Es una gráfica de los niveles de gris contra su frecuencia dentro de la imagen. Se logra definiendo un vector con un número de espacios igual al número de tonos de gris que tiene la imagen. El histograma de una imagen es una función discreta que representa el número de píxeles en la imagen en función de los niveles de intensidad,  $g$ . La probabilidad  $P(g)$  de ocurrencia de un determinado nivel  $g$  se define como:

$$P(g) = \frac{N(g)}{M} \quad (2.23)$$

Donde  $M$  es el número de píxeles en la imagen y  $N(g)$  es la frecuencia de  $g$ .

#### 2.3.2.1.6. Ecualizado

Permite modificar el contraste de una imagen: si la imagen está oscura, aumenta el brillo, y si está muy clara, disminuye el brillo.

$$g = \frac{N(g)}{M'} \quad (2.24)$$

Donde  $N(g)$  son las frecuencias de cada nivel de gris, comenzando desde el negro hasta el más brillante y  $M'$  se define como:

$$M' = \frac{r * c}{NG} \quad (2.25)$$

Donde  $r$  es el número de filas y  $c$  el número de columnas de la imagen y  $NG$  es el número de grises de la imagen. El ecualizado no altera las frecuencias de los tonos de gris, sino que reparte los píxeles en todo el ancho de los tonos de gris.

#### 2.3.2.1.7. Filtros de la media y mediana

En estas 2 técnicas, dada una imagen de entrada  $f(x, y)$  dan como resultado una imagen suavizada  $g(x, y)$  cuya intensidad para cada punto  $(x, y)$  se obtienen a partir de los valores de intensidad de los píxeles de  $f$  incluidos en el entorno de vecindad de  $(i, j)$ . El filtro de la media se puede expresar en la siguiente forma:

$$g(x, y) = \frac{1}{P} \sum_{(m,n) \in S} f(m, n) \quad (2.26)$$

Donde  $S$  es un entorno de vecindad de  $m \times n$  donde preferentemente  $m = n$  y  $P$  es el número total de píxeles en la subventana de  $m \times n$ . En lo que respecta al filtro de la mediana este también toma una vecindad alrededor del píxel  $(x, y)$  con la diferencia de que la intensidad del punto  $(x, y)$  de la imagen  $g$  es la mediana del vector de vecindad.

$$g(i, j) = \text{Vec}\left(\frac{P}{2}\right) \quad (2.27)$$

Donde  $\text{Vec}$  es el vector compuesto por los niveles de intensidad de la vecindad formada por la subventana de  $m \times n$  donde preferentemente  $m = n$  y  $P$  es el número total de píxeles en la subventana de  $m \times n$ .

### 2.3.2.1.8. Operador de Sobel

Los bordes en una imagen son píxeles alrededor de los cuales la imagen presenta una brusca variación en los niveles de gris. Existen razones por las cuales los bordes dentro de una imagen son importantes dado que limitan: contornos de objetos sólidos en la imagen, marcas en las superficies. Los contornos y curvas son elementos básicos para aplicaciones como, calibración, movimiento y reconocimiento. El operador de Sobel es un operador de primera derivada empleado para la determinación de los borde de una imagen. Los gradientes  $G_x$  y  $G_y$  están dados por las siguientes máscaras:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (2.28)$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ -1 & 2 & 1 \end{bmatrix}$$

### 2.3.2.2. Técnicas en el dominio de la frecuencia.

La base de las técnicas en el dominio de la frecuencia es el teorema de convolución. Sea  $g(x, y)$  una imagen formada por la convolución de una imagen  $f(x, y)$  y un operador lineal invariante de posición (aquél cuyo resultado depende sólo del valor de  $f(x, y)$  y no de su posición:

$$g(x, y) = h(x, y) * f(x, y) \quad (2.29)$$

Algunas técnicas del dominio de la frecuencia son:

- Transformada de Fourier.
- Transformada del coseno.

- Transformada Walsh-Hadamard.
- Transformada de Wavelets.
- Transformada de Haar.
- Transformada de Slant.

Estas técnicas no son tratadas en esta tesis dado que no son aplicadas en el procesamiento de las imágenes dentro del prototipo computacional.

### **2.3.3. Procesamiento morfológico de imágenes.**

La morfología comúnmente indica una rama de la biología que trata de la forma y estructura de los animales y plantas. En visión computacional se usa esta palabra en el contexto de la morfología matemática, como una herramienta para extraer componentes tales como los límites, los esqueletos y el contorno de una imagen que son útiles en la representación y descripción de la forma de una región. El lenguaje de la morfología matemática es la teoría de conjuntos. Los conjuntos en la morfología matemática representan las formas de los objetos en una imagen. La morfología matemática es una teoría que involucra operaciones lógicas booleanas sobre las imágenes, utilizando teoría de conjuntos adaptada a las imágenes. Las transformaciones resultantes de la morfología matemática pueden clasificarse en tres categorías:

- Unaria, una imagen de entrada produce una imagen de salida.
- Binaria, dos imágenes de entrada producen una imagen de salida
- Extracción de información, una imagen de entrada da como salida información numérica.

En cada una de estas categorías las operaciones sobre las imágenes binarias pueden ser lógicas o geométricas. Las operaciones unarias incluyen el complemento (negación), la reflexión, y la traslación. El complemento es la operación lógica, que cambia todos los píxeles que están activos a inactivos y viceversa. La traslación cambia todos los píxeles en una dirección dada a una distancia especificada. La reflexión asume un origen para la imagen y coordenadas para cada punto, multiplicando por -1 los píxeles que pertenecen a la región de interés, éstos son reflejados a través del origen, para imágenes bidimensionales esto es análogo a rotar la imagen 180 grados. Las operaciones binarias combinan dos imágenes en una.

Las operaciones morfológicas básicas son la dilatación y la erosión mismas que se explicaran enseguida en más detalle.

#### **2.3.3.1. Elemento estructural**

Las transformaciones de erosión y dilatación involucran la interacción entre una imagen  $A$  (el objeto de interés) y un conjunto estructural  $B$ , el cual es llamado *elemento estructural*. Comúnmente el elemento estructural  $B$  es un disco circular en el plano, pero puede tener cualquier forma. La figura 2.9 se muestra algunos de los tipos básicos de elementos estructurales. El elemento estructural puede ser visto de manera intuitiva como una máscara de convolución.

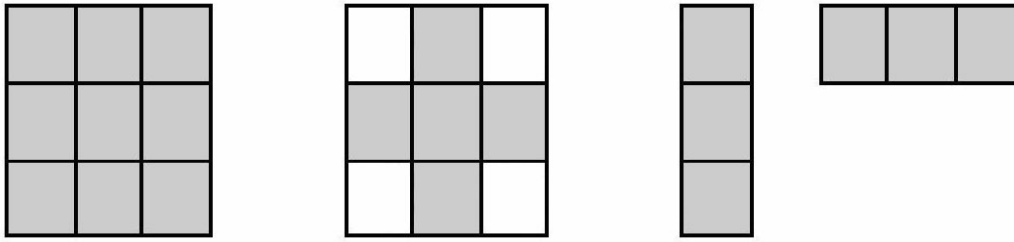


Fig. 2.9. Distintos tipos de elementos estructurales.

### 2.3.3.2. Dilatación

La dilatación de A por B se denota por  $A \otimes B$ , y se define formalmente como:

$$A \otimes B = \{x | (B_x) \cap A \neq \emptyset\} \quad (2.30)$$

Donde B representa al elemento estructural posicionado en x.

La dilatación es el lugar geométrico de los puntos, que al centrar o posicionar el elemento estructural en sus elementos, intersecta al conjunto.

La dilatación cumple con las propiedades asociativa y conmutativa. Por lo tanto en una secuencia de pasos de dilatación el orden de las operaciones realizadas no es importante, lo que permite descomponer una forma compleja en varias más simples, las cuales pueden ser recombinadas como una secuencia de dilataciones.

Considere como un ejemplo de dilatación, el mostrado en la figura 2.10, donde la imagen A es un rectángulo y el elemento estructural B es un disco centrado en el origen.

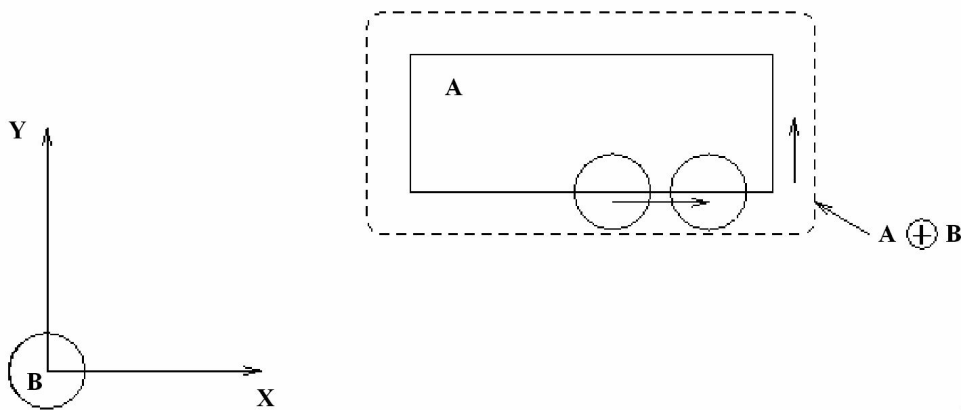


Fig. 2.10. A es dilatada por el elemento estructural B, las líneas punteadas muestran el resultado de la dilatación.

### 2.3.3.3. Erosión

La erosión es lo opuesto a la dilatación. La erosión de una imagen por un elemento estructural resulta en una imagen, que muestra todas las ubicaciones donde el elemento estructural está contenido en la imagen. El principal mecanismo del operador de la erosión, es la comparación local del elemento estructural con el objeto que será transformado, si para un punto dado el elemento estructural está contenido en el objeto, entonces este punto

aparecerá como resultado de la transformación. La erosión se define formalmente como:

$$A \ominus B = \{ z | (B)_z \cap A^c \neq \emptyset \} \quad (2.31)$$

La erosión es el lugar geométrico de los puntos, que al posicionar el elemento estructural en sus elementos, queda completamente contenido en el conjunto. Un ejemplo de erosión de la imagen  $A$  por el elemento estructural  $B$  se aprecia en la figura 2.11.

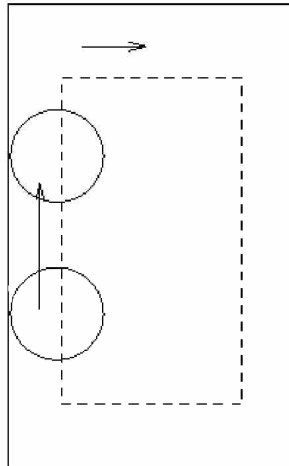


Fig. 2.11.  $A$  es erosionada por el elemento estructural  $B$ , el resultado de la erosión se representa con la línea punteada

Un ejemplo de los procesos de dilatación y erosión sobre una imagen binaria se observa en la figura 2.12.

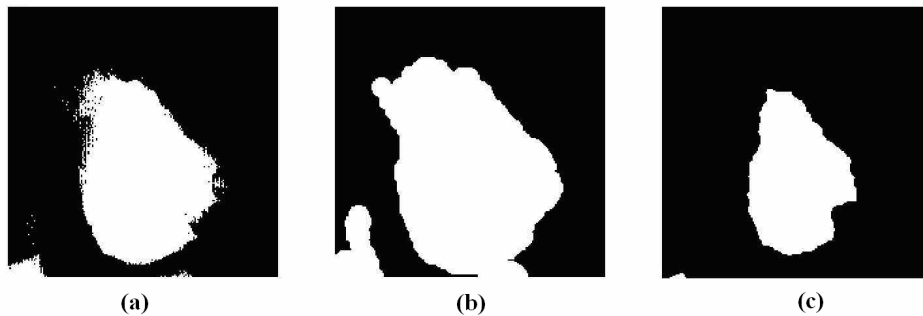


Fig. 2.12. (a) Muestra la imagen original, (b) muestra la imagen dilatada, (c) muestra la imagen erosionada.

La dilatación y la erosión son operaciones duales, porque la dilatación de los píxeles de interés es equivalente a la erosión de los píxeles del fondo. Estas operaciones básicas pueden combinarse, por ejemplo la erosión seguida de una dilatación en la misma imagen, removerá todos los píxeles en una región que son más pequeños que el elemento estructural. Esta secuencia es llamada *opening*. Si se utiliza un elemento estructural circular o de disco, todas las regiones de píxeles más pequeñas que el disco serán eliminadas. Esto forma un filtro que suprime detalles espaciales positivos. La secuencia opuesta, una dilatación seguida de una erosión, operación llamada *closing*, rellena los huecos que son más pequeños que el elemento estructural. Puede observarse que la dilatación expande un objeto en una imagen mientras que la erosión lo contrae.

## 2.4. Redes neuronales artificiales

Los seres vivos poseen gran habilidad para realizar algunas tareas que algorítmicamente serían muy costosas o difíciles de realizar. Un ejemplo de esto es la destreza de un insecto que vuela por una habitación sin chocar con las paredes. Intuitivamente parece poco probable que el insecto calcule la disparidad de lo que ven sus ojos, con esta información genere un mapa con las coordenadas de todos los objetos que lo rodean, calcule su ecuación de movimiento, la derive y calcule la fuerza a aplicar en sus alas para modificar su trayectoria convenientemente. Esto nos hace pensar que hay otra forma de realizar estas tareas.

Las redes neuronales intentan dar un enfoque distinto al modo de resolver los problemas del que le dan los métodos tradicionales. Estas no se programan, se entrenan. El entrenamiento consiste en presentarle a la red un conjunto de patrones de entrada-salida que de alguna forma se conoce *a priori*. Este conjunto de patrones entrada-salida se denomina *conjunto de entrenamiento*. Luego, se desea que la red neuronal infiera la regla que rige estos patrones y sea capaz de *generalizar*. O sea, de obtener la salida correcta ante patrones de entrada que no pertenecen al conjunto de entrenamiento.

Los algoritmos de clasificación basados en redes neuronales representan una importante rama en la disciplina del reconocimiento de patrones debido a su capacidad de resolver problemas complejos de clasificación en los cuales el conocimiento *a priori* de la estructura espacial y probabilística de los patrones presentados es poco o nulo

A continuación se presenta un resumen de los fundamentos sobre redes neuronales artificiales. Este tema está bastante desarrollado en las referencias [57-63], por tal motivo solo pasaremos a hacer alguna breve descripción de algunos aspectos básicos. Una red neuronal es un procesador distribuido en paralelo, que puede filtrar, procesar las informaciones y tomar una decisión. Una red neuronal se parece al cerebro en dos ideas básicas:

- La red necesita una cantidad de información para entrenarse.
- Las conexiones entre las neuronas se usan para almacenar las informaciones

El uso de las redes neuronales ofrece muchas propiedades y capacidades, como el aprendizaje adaptativo, autoorganizativo, funcionamiento en paralelo en tiempo real y tolerancia a fallos por la codificación redundante de la información. Desde el punto de vista de solucionar problemas, las redes neuronales son diferentes de los ordenadores convencionales que usan algoritmos secuenciales, mientras que las redes neuronales actúan como el cerebro humano, procesando la información en paralelo, y también pueden aprender y generalizar a casos nuevos que no estaban incluidos durante el proceso del diseño. Las redes neuronales pueden procesar las informaciones más rápido que los ordenadores convencionales, pero tiene la desventaja de que no podemos seguir su respuesta paso a paso como se puede hacer al ejecutar un programa convencional en un ordenador por lo que no resulta fácil detectar los errores.

### 2.4.1. Neurona Biológica

La base y fuente de inspiración de las redes neuronales artificiales es la célula del



sistema nervioso de los animales, conocida como neurona, y es por tanto importante observar su fisiología para comprender como los investigadores en ingeniería y matemática tratan de imitar los mecanismos de almacenamiento y procesamiento de la información en el cerebro. Un ser humano adulto posee aproximadamente  $10^{12}$  neuronas, cada una de las cuales se conecta en promedio con otras  $10^5$  neuronas. En la Figura 2.13 se puede apreciar un esquema simplificado de algunas neuronas. En ella se puede observar como la morfología de una neurona comprende tres elementos principales: el soma, o cuerpo principal; las dendritas, o terminaciones de la neurona que actúan como contactos funcionales de entrada con otras neuronas; y el axón, una rama más larga que será la encargada de conducir el impulso nervioso y que finaliza también en diversas ramificaciones. La comunicación entre neuronas se realiza a través de las sinapsis, que son los puntos de conexión entre las fibras terminales del axón de una neurona y una dendrita de otra. El impulso nervioso producido por una neurona se propaga por el axón y, al llegar a un extremo, las fibras terminales pre-sinápticas liberan compuestos químicos llamados neurotransmisores. Los neurotransmisores se liberan en la membrana celular pre-sináptica y alteran el estado eléctrico de la membrana post-sináptica. En función del neurotransmisor liberado, el mecanismo puede resultar excitador o inhibitorio para la neurona *receptora*. En el soma de una neurona se integran todos los estímulos recibidos a través de todas sus dendritas. Si como resultado se supera un potencial de activación, la neurona se “dispara”, generando un impulso que se transmitirá a través del axón. Sino, se mantiene en reposo. La repercusión de un estímulo nervioso en el estado excitatorio de la neurona receptora no es constante y se modifica con el tiempo en el proceso de aprendizaje a esto se lo denomina *plasticidad sinóptica*. Es de esperar, por ejemplo, que una neurona ubicada en la parte del cerebro dedicada a la visión, sea capaz de procesar la información o estímulos provenientes del nervio óptico e ignorar aquellos estímulos que provienen de otros centros receptivos.

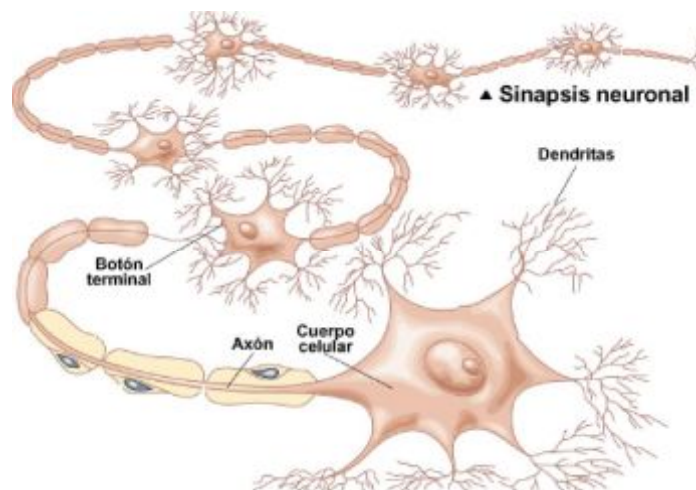


Fig. 2.13. Modelo de neurona biológica

Para construir un modelo formal y simplificado se rescataron los siguientes aspectos:

- La neurona posee sólo dos estados: *excitatorio o de reposo*.
- Existen dos tipos de sinapsis: *excitatorias e inhibitorias*.
- La neurona es un dispositivo integrador, ya que suma los impulsos que le llegan

de las dendritas.

- Cada sinapsis transmite con mayor o menor intensidad los estímulos eléctricos, de acuerdo a su *Capacidad Sinóptica*.
- El aprendizaje consiste en modificaciones en las sinapsis.

## **2.4.2. Neurona Artificial**

Una neurona artificial, pretende simular el modelo y las características más importantes de la neurona biológica; es una unidad de procesamiento constituido por el núcleo, uno o varias entradas y una salida. Las entradas representan señales que provienen de otras neuronas, el núcleo es el encargado de proporcionar un valor de salida basado en una función umbral que la neurona debe sobrepasar para activarse. La salida normalmente se conecta a otras neuronas.

Una neurona es el elemento fundamental para la construcción de cualquier modelo de red neuronal. A partir de ellas, se pueden generar representaciones específicas, de tal forma que un estado conjunto de ellas puede significar una letra, un número o cualquier otro objeto.

### **2.4.2.1. Tipos de Neuronas**

Se pueden encontrar tres tipos de neuronas:

- Las unidades de entrada, que reciben señales (información de entrada) desde el entorno ya sea de sensores o de otros sectores del sistema.
- Dicha información se transmite a ciertos elementos internos que se ocupan de su procesamiento. Es en la sinapsis y neuronas correspondientes a este segundo nivel donde se genera cualquier tipo de representación interna de la información. A estos elementos se los llama unidades ocultas.
- Una vez finalizado el período de procesamiento, la información llega a las unidades de salida, cuya misión es dar la respuesta del sistema.

## **2.4.3. Red Neuronal Artificial**

Las redes neuronales son modelos, generalmente llamados modelos conexionistas, que intentan reproducir el comportamiento del cerebro, referente a su arquitectura y funcionamiento. Las redes neuronales están compuestas de un gran número de unidades de procesamiento conocidas como neuronas artificiales, trabajando al mismo tiempo para la solución de problemas específicos. El diseño de la red está determinado mayormente por las conexiones entre sus elementos.

Para cuestiones de comprensión se referirá a las redes neuronales artificiales como redes neuronales, omitiendo la palabra artificial.

### **2.4.3.1. Elementos de una Red Neuronal**

En la estructura de una red neuronal se distinguen los siguientes elementos:

- Unidades de procesamiento.
- Niveles o capas.
- Estado de activación.
- Patrón de conectividad.
- Regla de propagación.
- Función de transferencia.
- Regla de aprendizaje.

#### 2.4.3.1.1. Unidades de Procesamiento

Las unidades de procesamiento, son básicamente las neuronas. Como se puede ver en la figura 2.14 cada unidad de procesamiento está caracterizada en cualquier instante por un valor numérico denominado *estado de activación*  $a_j(t)$ ; asociado a cada unidad, existe una *función de salida*  $f_j$ , que transforma el estado de activación en una *señal de salida*  $y_j$ , cuando sobrepasa un valor umbral. Dicha señal es enviada a través de los canales de comunicación unidireccionales a otras unidades de la red; en estos canales la señal se modifica de acuerdo con la sinapsis (el peso,  $w_{ji}$ ) asociada a cada uno de ellos según una determinada regla.

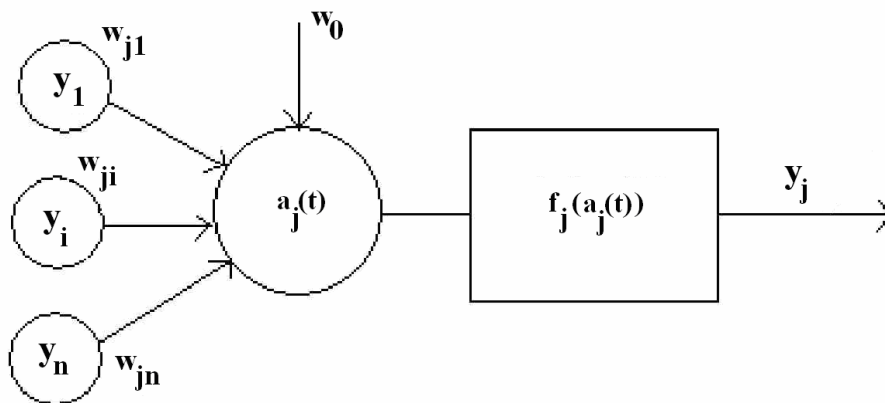


Fig. 2.14. Unidad de proceso (Neurona Artificial)

#### 2.4.3.1.2. Niveles o Capas

Una capa o nivel es un conjunto de unidades de procesamiento cuyas entradas provienen de la misma fuente (que puede ser otra capa de neuronas) y cuyas salidas se dirigen al mismo destino (que puede ser otra capa de neuronas). Existen tres tipos de capas de acuerdo a si tienen unidades de entrada, ocultas o de salida. La información dada a la red neuronal, mediante su capa de entrada, es propagada capa tras capa hasta la capa de salida.

#### 2.4.3.1.3. Estado de activación

El estado de activación  $a_i(t)$  es un valor asignado a la unidad de procesamiento  $U_i$  en un tiempo  $t$ , este estado es calculado de acuerdo a ciertos criterios o reglas. El estado de activación del sistema en un tiempo  $t$ , es representado vectorialmente como:

$$A(t) = (a_1(t), a_2(t), \dots, a_N(t)) \quad (2.32)$$

Cada neurona en un determinado tiempo se encuentra en un cierto estado de activación, es decir, en reposo o excitada, a cada uno de los cuales se le asigna un valor continuo o discreto. En notación binaria, un estado activo estaría indicado por un 1, que indicaría la emisión de un impulso por parte de la neurona; mientras un estado de reposo o pasivo está indicado por un 0. En el caso de valores discretos, se consideran valores en el intervalo  $[0,1]$  o  $[-1,1]$ , siguiendo alguna función conocida, generalmente la sigmoideal.

#### 2.4.3.1.3. Función de Transferencia

La función de transferencia corresponde a una señal transmitida por cada unidad hacia otra, determinando de esta manera una conexión entre estas. Dicho de otra manera, asociado a cada unidad  $U_i$  existe una función de salida  $f_i(a_i(t))$ , que transforma el estado actual de activación  $a_i(t)$  en una señal de salida  $y_i(t)$ .

$$y_i(t) = f_i(a_i(t)) \quad (2.33)$$

El vector que contiene las salidas de todas las neuronas en un instante  $t$  es:

$$Y(t) = (f_1(a_1(t)), f_2(a_2(t)), \dots, f_N(a_N(t))) \quad (2.34)$$

Las funciones de transferencia más comunes son: función escalón, función lineal y mixta, función Sigmoideal y función Gaussiana. En la tabla 2.1, se muestra algunos de los diferentes tipos de funciones de transferencia aplicadas a las redes neuronales artificiales.

#### 2.4.3.1.4. Patrón de conectividad

Las conexiones que unen a las neuronas tienen asociado un peso, que es el que hace que la red adquiera conocimiento. Cada conexión entre la neurona  $i$  y la neurona  $j$  está ponderada por un peso  $w_{ji}$ . La matriz  $W$  de pesos refleja la influencia que la neurona  $i$  tiene sobre la neurona  $j$ .

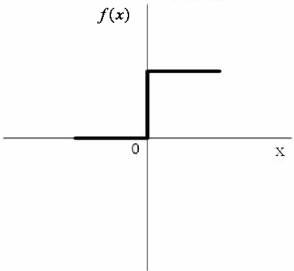
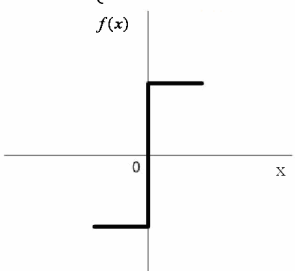
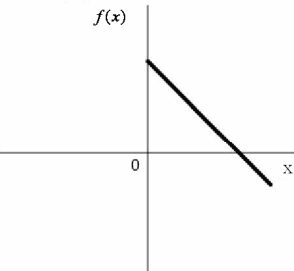
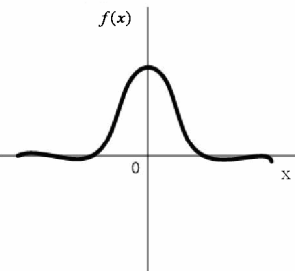
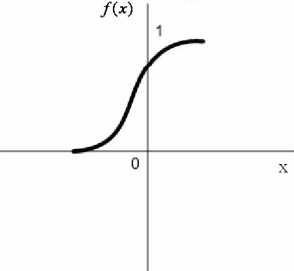
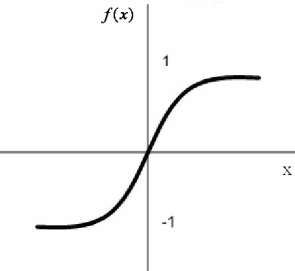
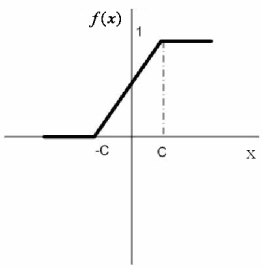
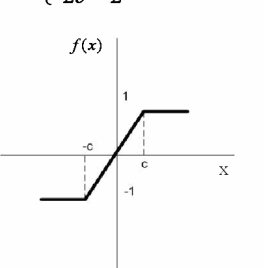
Existen tres tipos de conexiones, las excitadoras de valor positivo que tiende a activar la neurona, las inhibidoras de valor negativo que tienden a desactivar la neurona y las nulas con un peso igual a cero, que indica que no hay conexión entre ambas.

#### 2.4.3.1.5. Regla de propagación

La regla de propagación indica el procedimiento a seguir para combinar los valores de entrada a una unidad con los pesos de las conexiones que llegan a esa unidad. Así se considera una *entrada neta* como la suma del producto de cada señal recibida  $y_i$  por el valor de la sinapsis que conecta ambas neuronas  $w_{ji}$  más el peso umbral de la neurona  $j$   $w_{j0}$  [63].

$$net_j = \sum_i^n w_{ji} \cdot y_i + w_{j0} \quad (2.35)$$

Tabla 2.1. Diferentes tipos de funciones de transferencia aplicadas a las redes neuronales artificiales [63].

<p style="text-align: center;"><b>Función Limitador Fuerte</b></p> $f(x) = \begin{cases} 1 & \text{si } x \geq 0 \\ 0 & \text{si } x < 0 \end{cases}$ 	<p style="text-align: center;"><b>Función Limitador Fuerte Simétrico</b></p> $f(x) = \begin{cases} 1 & \text{si } x > 0 \\ -1 & \text{si } x < 0 \end{cases}$ 
<p style="text-align: center;"><b>Función Lineal</b></p> $f(x) = ax + b$ 	<p style="text-align: center;"><b>Función Gaussiana</b></p> $f(x) = Ae^{-Bx^2}$ 
<b>Función Sigmoideal</b>	
$f(x) = \frac{1}{1 + e^{-\alpha x}}$ 	$f(x) = 1 - \frac{1}{1 + e^{-\alpha x}}$ 
<b>Función Mixta</b>	
$f(x) = \begin{cases} 0 & x < -c \\ \frac{x}{2c} + \frac{1}{2} & \text{otro caso} \\ 1 & x > c \end{cases}$ 	$f(x) = \begin{cases} -1 & x < -c \\ \frac{x}{2c} + \frac{1}{2} & \text{otro caso} \\ 1 & x > c \end{cases}$ 

#### **2.4.3.1.6. Regla de Aprendizaje**

El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada, es decir, inducido por la interacción con el entorno, y como resultado de experiencias conducentes al establecimiento de nuevos modelos de respuesta a estímulos externos.

Biológicamente la información memorizada está relacionada con los valores sinápticos más que con las neuronas mismas. En una red neuronal el conocimiento se encuentra representado en los pesos de las conexiones entre neuronas.

#### **2.4.4. Características de una Red Neuronal**

Una red neuronal está definida por ciertas características, entre las que se encuentran:

- Topología de la red.
- Mecanismo de aprendizaje que utiliza.
- Tipo de asociación realizada entre la información de entrada y de salida.
- Forma de representación de estas informaciones.

##### **2.4.4.1. Topología**

La topología consiste en la organización y disposición de las neuronas en la red formando capas o niveles, en este sentido hay distinción entre redes con una sola capa y las redes con múltiples capas.

###### **2.4.4.1.1. Redes Monocapa**

En las redes monocapa se establecen conexiones laterales entre neuronas pertenecientes a la única capa que constituye la red, pudiendo existir también las conexiones autorrecurrentes. Estas redes se utilizan típicamente en tareas relacionadas con lo que se conoce como autoasociación, por ejemplo para regenerar informaciones de entrada que se presentan a redes incompletas o distorsionadas.

Entre las redes monocapa más importantes están: el modelo de red de Hopfield, la Máquina de Boltzmann, la máquina de Cauchy, mismas que presentan conexiones autorrecurrentes. Entre otros modelos monocapa que no presentan conexiones autorrecurrentes están el modelo Brain-state-in-a-box o el modelo Optimal Linear Associative Memory.

###### **2.4.4.1.2. Redes Multicapa**

En las redes multicapa, las neuronas están agrupadas en varias capas; en este tipo de topologías existen las conexiones hacia adelante o feedforward, ver figura 2.15 en las que las neuronas de una capa solamente pueden ser entradas de las neuronas de la siguiente capa.

El otro tipo de conexiones son las que tienen conexiones hacia atrás (feedback), en éstas las salidas de las neuronas posteriores están conectadas hacia las entradas de las capas anteriores. Este tipo de redes puede tener conexiones entre neuronas de la misma capa, así como conexiones con capas anteriores.

Es por tal razón que las redes multicapa pueden ser de dos tipos: redes multicapa con conexiones hacia adelante (feedforward), y las redes multicapa con conexiones hacia adelante y hacia atrás (feedforward/feedback). En el caso de las redes feedforward, su utilidad es vista en aplicaciones de reconocimiento o clasificación de patrones.

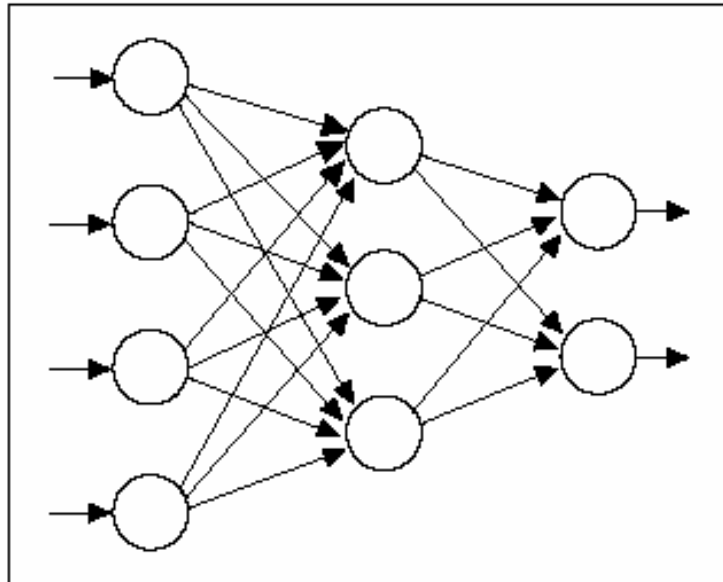


Fig. 2.15. Conexiones feedforward [62]

En las redes feedforward/feedback la información circula tanto hacia adelante como hacia atrás durante el funcionamiento de la red; existiendo por lo general dos conjuntos de pesos, uno para cada tipo de conexión. Habitualmente este tipo de redes tiene una estructura bicapa, teniendo su aplicación para la asociación de una información o patrón de entrada en la primera capa con otra información o patrón de salida en la segunda capa, estas dos capas interactúan entre si hasta alcanzar un estado estable, lo que se conoce como resonancia.

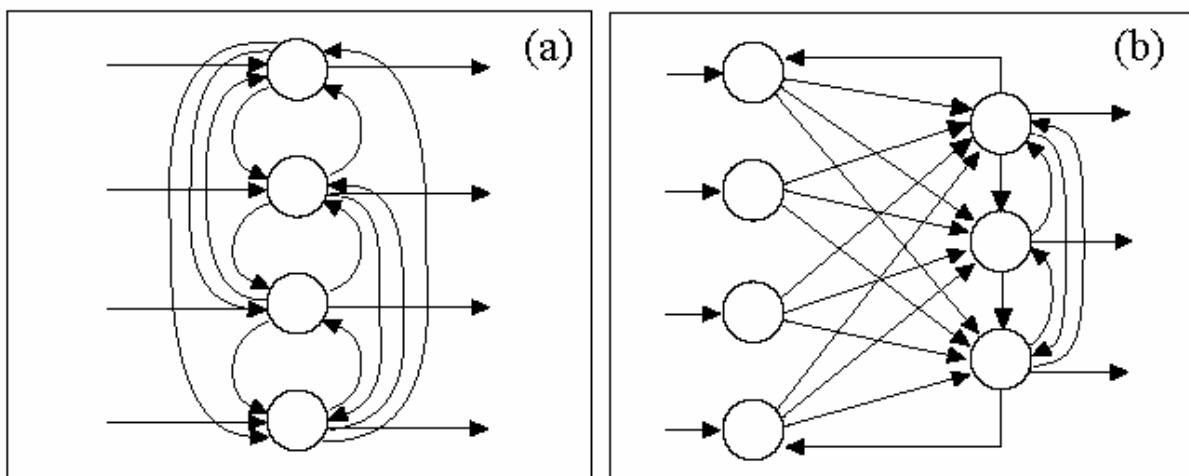


Fig.2.16 (a) Conexiones laterales. (b) Conexiones feedforward/feedback [62]

También en este grupo de redes existen las que tienen conexiones laterales entre neuronas de la misma capa, figura 2.16(a). Estas conexiones se diseñan como conexiones

excitadoras, permitiendo la cooperación entre neuronas, o como inhibidoras, estableciéndose una competición entre las neuronas correspondientes.

Las redes con conexiones feedforward/feedback se las conoce también como redes dinámicas ya que aprenden de la experiencia pasada, esto gracias a sus conexiones hacia atrás, figura 2.16). Que envían la información de salida de un tiempo  $t$  a neuronas de capas anteriores para crear nuevos valores en tiempo  $t + 1$ .

En otro tipo de redes multicapa las neuronas se disponen en planos superpuestos (capas bidimensionales), lo cual permite que puedan eliminarse las variaciones geométricas como el tamaño, giros, desplazamientos o distorsiones que presenten las informaciones o patrones de entrada a la red, éste es el caso de la red Neocognitron.

En la tabla siguiente se muestra un resumen de los modelos más conocidos y el tipo de conexión que presentan.

Tabla 2.2. Clasificación de las redes multicapas.

<b>Nº de Capas</b>	<b>TIPO DE CONEXIONES</b>		<b>MODELO DE RED</b>	
2	Feedforward		ADELINE/MADELINE	
			PERCEPTRÓN	
			LINEAR ASSOCIATIVE MEMORY	
	Feedforward /Feedback	Conexiones laterales implícitas y Autorrecurrentes		LEARNING VECTOR QUANTIZER
				TOPOLOGY PRESERVING MAP (TPM)
		Sin conexiones laterales		BIDIRECTIONAL ASSOCIATIVE MEMORY (BAM)
				TEMPORAL ASSOC. MEMORY (TAM)
Con conexiones laterales y autorrecurrentes		COMPETITIVE ADAPTIVE BAM		
		ADAPTIVE RESONANCE THEORY (ART)		
3	Feedforward	Con conexiones laterales	COUNTERPROPAGATION	
N	Feedforward		BACKPROPAGATION (BPN)	
	Feedforward/Feedback		COGNITRON/NEOCOGNITRON	

#### 2.4.4.2. Mecanismos de Aprendizaje

Durante el proceso de aprendizaje, los pesos de las conexiones de la red sufren modificaciones, en base a los patrones de entrada presentados a la red, de modo que la red ajuste apropiadamente los pesos de las conexiones entre neuronas. El proceso termina cuando los pesos permanecen estables, en este punto se puede decir que la red ha aprendido.

El proceso para hacer que una red neuronal aprenda, se rige por ciertas reglas de aprendizaje, de manera general se consideran el aprendizaje supervisado, el no supervisado, el reforzado y el híbrido.

Las reglas de aprendizaje se fueron creando junto con los modelos de redes



neuronales, cada una distinta de las otras, adecuada para un modelo específico. Puesto que existe gran cantidad de algoritmos de aprendizaje y arquitecturas en la literatura de redes neuronales, es imposible realizar un único esquema de clasificación que pueda capturar las características principales de los diversos paradigmas.

El aprendizaje supervisado significa proveer a la red de la salida deseada, ya sea calificando manualmente a la actuación de la red o proveyendo a la misma con los resultados esperados. En el aprendizaje no supervisado, la red es provista de los datos de entrada pero no los de salida, y el sistema debe “decidir” qué características usar para agrupar los datos.

La diferencia fundamental entre ambos tipos estriba en la existencia o no de un agente externo (supervisor) que controle el proceso de aprendizaje de la red.

#### 2.4.4.2.1. Aprendizaje Supervisado

El proceso de aprendizaje se realiza a través de un agente externo que determina la respuesta que debería generar la red a partir de una entrada determinada. En el caso que la salida no coincida con la entrada, se procede a modificar los pesos con el fin de conseguir una aproximación a la salida obtenida [63].

Según Hilera [63], se consideran tres formas de aprendizajes supervisados:

- Aprendizaje por corrección de error.
- Aprendizaje por refuerzo.
- Aprendizaje estocástico

En el aprendizaje por corrección de error, los ajustes se realizan aplicando la siguiente regla:

$$\Delta w_{ji} = \alpha y_i (d_j - y_j) \quad (2.36)$$

Siendo:

$\Delta w_{ji}$ : Variación del peso entre las neuronas  $i$  y  $j$ .

$y_i$ : Valor de salida de la neurona  $i$ .

$d_j$ : Valor de salida deseado para la neurona  $j$ .

$y_j$ : Valor de salida obtenido en la neurona  $j$ .

$\alpha$ : Factor de aprendizaje ( $0 < \alpha < 1$ ) que regula la velocidad del aprendizaje.

Windrow y Hoff definieron una función para cuantificar el error global cometido en cualquier momento durante el proceso de aprendizaje de la red, este error es calculado por la siguiente fórmula:

$$Error_{global} = \frac{1}{2P} \sum_{k=1}^P \sum_{j=1}^N (y_j - d_j)^2 \quad (2.37)$$

Siendo:

$N$ : Número de neuronas de salida.

$P$ : Número de informaciones que debe aprender la red.

$$\frac{1}{2} \sum_{j=1}^N (y_j - d_j)^2 : \text{ Error cometido en el aprendizaje de la información } k\text{-ésima.}$$

Por tanto, en este aprendizaje se trata de encontrar unos pesos para las conexiones de la red que minimicen esta función de error. Para ello, el ajuste de los pesos se puede hacer de la forma proporcional a la variación relativa del error que se obtiene al variar el peso correspondiente:

$$\Delta w_{ji} = k \frac{\partial Error_{global}}{\partial w_{ji}} \quad (2.38)$$

Esta regla es conocida como Regla Delta o del Mínimo Error Cuadrado (LMS, en inglés). Mediante este procedimiento, se llega a obtener un conjunto de pesos con los que se consigue minimizar el error medio.

En el aprendizaje por refuerzo la función del supervisor es la de indicar mediante una señal de refuerzo si la salida obtenida en la red se ajusta a la deseada (éxito = +1 o fracaso = -1), y en función de ello se ajustan los pesos basándose en un mecanismo de probabilidades. Este aprendizaje es más lento que el anterior, puesto que el supervisor es un crítico más que un maestro.

El aprendizaje estocástico consiste básicamente en realizar cambios aleatorios en los valores de los pesos de las conexiones de la red y evaluar su efecto a partir del objetivo deseado y de distribuciones de probabilidad.

Existen diversas variantes referentes a estos tipos de aprendizaje, cada una de las cuales se va ajustando a cierto tipo de red neuronal.

#### 2.4.4.2.2. Aprendizaje no Supervisado

En el aprendizaje no supervisado, la red no cuenta con salidas esperadas, por lo tanto no utiliza un tutor o supervisor y se basa solamente en la información ingresada. Esto se refiere a una auto-organización en el sentido que la red organiza automáticamente los datos presentados a la red y detecta sus propiedades colectivas emergentes, es por ello que suele decirse que estas redes son capaces de autoorganizarse.

El aprendizaje no supervisado se utiliza generalmente para realizar algunas caracterizaciones iniciales a los datos de entrada. Un modelo básico de esto utiliza los mapas autoorganizativos de Kohonen. En cuanto a la posibilidad de verdadero aprendizaje por parte de estos sistemas, aún es una promesa lejos de lograrse, e incluso de entenderse. Esta posibilidad todavía está relegada al laboratorio.

Dependiendo de la red, la salida puede representar el grado de familiaridad o similitud entre la información de entrada y las informaciones que se le han mostrado hasta

entonces; como también la salida puede referirse a una categorización (clustering), es decir, indicar la categoría a la que pertenece la información presentada en la entrada, en este caso, cada salida es una categoría. Algunas redes lo que realizan con un aprendizaje no supervisado es un mapeo de características, que representan un mapa topológico de las características de los datos de entrada.

#### **2.4.4.2.3. Aprendizaje Reforzado**

En un aprendizaje reforzado por una parte se emplea la información del error cometido pero se sigue sin poseer la salida deseada. Dicho aprendizaje descansa en la idea dual premio castigo, donde se refuerza toda aquella acción que permita una mejora del modelo mediante la definición de una señal crítica. Esta estrategia de aprendizaje permite tratar con patrones de entrada diferidos que aparecen, por ejemplo, en aplicaciones de robótica.

#### **2.4.4.2.4. Aprendizaje Híbrido**

En el aprendizaje híbrido coexisten en el modelo neuronal los dos tipos básicos de aprendizaje, el supervisado y el no supervisado, normalmente en distintas capas de neuronas. Modelos de este tipo son, por ejemplo, Red Counter-Propagation y Red de función en Base Radial.

#### **2.4.4.3. Tipo de asociación entre la información de Entrada/Salida**

De acuerdo a la naturaleza de la información almacenada en la red, la asociación que presenta la red entre la información de entrada y de salida, se puede clasificar en heteroasociativa y autoasociativa.

Una red heteroasociativa se refiere a asociar una salida diferente para cada información de entrada distinta, de tal forma que cuando se presente cierta información de entrada la red deberá responder generando la correspondiente salida asociada. Mientras que en una red autoasociativa, la red aprende ciertas informaciones durante el entrenamiento, de tal forma que cuando se le presenta una información de entrada en la fase de ejecución, realizará una autocorrelación, respondiendo con uno de los datos almacenados, el más parecido al de la entrada.

#### **2.4.4.4. Forma de Representación de la información**

Un estímulo presentado a la red, es representado generalmente en forma de vector, en este caso el vector viene a contener las características más importantes de la información de entrada.

Esto se realiza a causa de que la información que se introduce a la red generalmente no está en el formato que la red utiliza, entonces es necesario abstraer el problema que se modela en una red neuronal para representarlo como información válida.

### **2.4.5. Aspectos para el diseño de la topología**

En la topología de las redes neuronales resulta de interés analizar lo referente a

niveles o capas, así como la cantidad de neuronas por capas que son necesarias. En el nivel inicial existe una unidad sensorial por cada rasgo de entrada a la red, en la capa de salida se colocan tantas neuronas como sean necesarias. Se ha probado que dos capas ocultas son suficientes para resolver cualquier problema, pero aún no existe un criterio riguroso para determinar la cantidad de neuronas en las capas ocultas.

Mientras que existe la propagación de errores hacia atrás para la definición de los pesos, no existe un parámetro general para la definición de la topología. Por consiguiente, se debe considerar: el número de unidades a utilizar, la conectividad y el número de capas ocultas necesarias.

#### **2.4.5.1 Número de Capas**

Ha sido probado que a lo más dos capas de unidades ocultas son suficientes para aproximar cualquier función, y a lo más una capa de unidades ocultas es suficiente para aproximar cualquier función continua.

Sin embargo, se puede o no conocer si la función es capaz de ser aprendida en un intervalo razonable de tiempo. Según el teorema de Kolmogorov: “Cualquier función continua puede ser aproximada con una sola capa oculta, más capas hacen más fácil el aprendizaje, más capas hacen más lento el proceso”.

#### **2.4.5.2 Número de Unidades**

Los algoritmos utilizados para determinar el número de unidades ocultas pueden ser clasificados en dos categorías: las clases destructivas y las clases constructivas. Los algoritmos destructivos empiezan de una gran red y luego se procede a la poda, removiendo enlaces y neuronas. Los algoritmos constructivos comienzan con una red bastante simple y hacen que la misma se transforme en una red más compleja en el curso del proceso.

### **2.4.6. Red Backpropagation**

En 1986, Rumelhart, Hinton y Williams formalizaron un método para que una red neuronal aprendiera la asociación que existe entre los patrones de entrada a la misma y las clases correspondientes utilizando más niveles de neuronas de los que empleó Rosenblatt en su Perceptrón. Este método conocido como Backpropagation está basado en la generalización de la regla delta y ha ampliado considerablemente el rango de aplicación de las redes neuronales.

En una red backpropagation existe una capa de entrada con  $n$  neuronas y una capa de salida con  $m$  neuronas y al menos una capa oculta de neuronas internas. En la figura 2.19 se muestra la arquitectura de una red backpropagation de 3 capas. Cada neurona de una capa (salvo la de entrada) recibe entrada de todas las neuronas de la capa previa y genera salida hacia todas las neuronas de la capa siguiente (salvo las neuronas de salida). No hay conexiones hacia atrás feedback, ni laterales o autorrecurrentes.

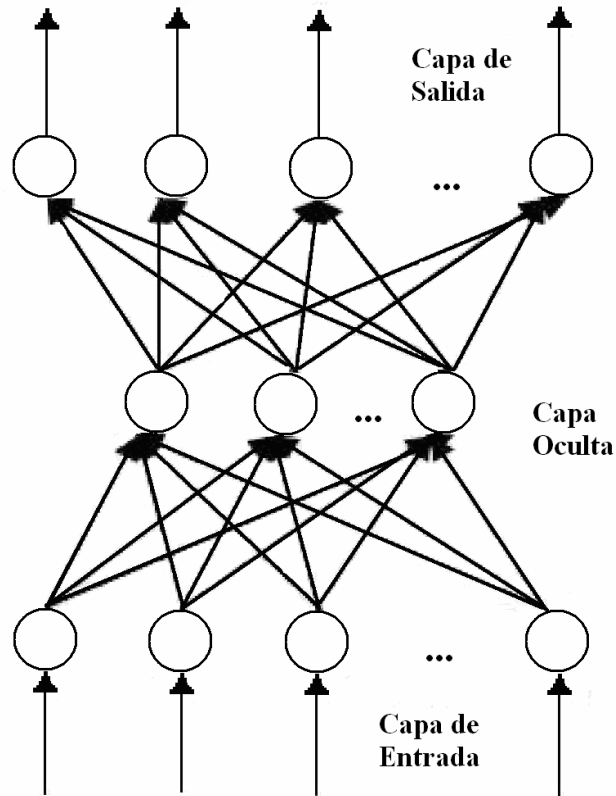


Fig. 2.17. Arquitectura de una red Backpropagation [63].

### 2.4.6.1. Aprendizaje

La regla de aprendizaje utilizada por esta red es el Backpropagation o Propagación hacia atrás. Ésta regla se puede aplicar en modelos de redes con más de dos capas de neuronas. En este tipo de red neuronal la representación interna del conocimiento se organiza en las distintas capas ocultas para lograr cualquier correspondencia entre entradas y salidas, de esta manera la red aprende una relación entre un conjunto de patrones dado como ejemplo y aplica la misma relación con nuevos patrones de entrada.

De forma simplificada, el funcionamiento de una Red Backpropagation consiste en un aprendizaje de un conjunto predefinido de pares entrada-salida dados como ejemplo, empleando un ciclo propagación-adaptación de dos fases: primero se aplica un patrón de entrada en la capa de entrada de la red el cual se propaga hacia las capas superiores hasta generar una salida, se compara el resultado obtenido en cada neurona de salida con el valor deseado para esa neurona y se obtiene un error para dicha unidad. A continuación, estos errores se transmiten hacia atrás hacia todas las neuronas de la capa intermedia que contribuyan directamente a la salida, recibiendo el porcentaje de error aproximado a la participación de la neurona intermedia en la salida original, de esta manera la red ajusta los parámetros de la red para minimizar el error medio cuadrático.

Este proceso se repite capa por capa hasta llegar a la entrada y hasta que cada neurona haya recibido un error que describa su aporte al error total. Según el valor del error recibido, se reajustan los pesos de las conexiones entre cada par de neuronas en la red, de manera de que el error total cometido para ese patrón disminuya.

### 2.4.6.2. Regla Delta Generalizada.

La regla propuesta por Widrow en 1960 conocida entonces como Regla Delta, ha sido extendida a redes con capas intermedias en la llamada Regla Delta Generalizada desarrollada por Rumelhart, con conexiones hacia delante y cuyas células tienen funciones de activación continuas, no decrecientes y derivables generalmente son lineales o sigmoideas, dando lugar al algoritmo de retropropagación del error.

Este algoritmo utiliza una función o superficie de error asociada a la red, buscando el estado estable de mínimo error a través del cálculo del gradiente de la superficie de error. Para ello, realimenta el error del sistema para realizar la modificación de los pesos en un valor proporcional al gradiente decreciente de dicha función de error.

### 2.4.6.3. Algoritmo de la Retropropagación del Error

Dada una neurona ( $U_i$ ) y su salida  $y_i$ , el cambio que se produce en el peso de la conexión que une la salida de dicha neurona con la unidad  $U_j$ ,  $w_{ji}$  para un patrón de aprendizaje  $p$  es:

$$\Delta w_{ij}(t+1) = \alpha \delta_{pj} y_{pi} \quad (2.39)$$

Donde:

- $p$ : Patrón de aprendizaje concreto.
- $\alpha$ : Constante o tasa de aprendizaje.

El punto en el que difieren la regla delta generalizada de la regla delta es el valor concreto de  $\delta_{pj}$ . Por otro lado, en las redes multinivel con capas ocultas no se conoce la salida deseada de las unidades de estas capas para poder calcular los pesos en función del error cometido. Sin embargo, inicialmente se puede conocer la salida deseada de las neuronas de salida. Para la unidad  $U_j$  de salida, se define:

$$\delta_{pj} = (d_{pj} - y_{pj}) f'(net_j) \quad (2.40)$$

Donde:

- $\delta_{pj}$ : Salida deseada para la neurona  $j$  y el patrón  $p$ .
- $net_j$ : Entrada neta de la neurona  $j$ .

Si  $U_j$  no es de salida, se tiene:

$$\delta_{pj} = \left( \sum_k \delta_{pk} w_{pk} \right) f'(net_j) \quad (2.41)$$

Siendo:

$k$ : rango que cubre todas las neuronas conectadas a la salida de  $U_j$ .

El error que se produce en una neurona oculta es la suma de todos los errores cometidos por las neuronas a las que está conectada su salida, multiplicados por el peso de la

conexión correspondiente.

En síntesis, durante la fase de entrenamiento se siguen los siguientes pasos para aplicar el algoritmo de entrenamiento:

- Inicializar los pesos  $w_{kj}$  de la red con valores pequeños aleatorios.
- Presentar un patrón de entrada  $X_p$  y especificar la salida deseada que debe generar la red para el patrón presentado.
- Calcular la salida actual de la red, para ello se aplican las fórmulas detalladas en el apartado 2.4.3.1.5.
- Calcular los términos de error para todas las neuronas.
- Utilizar un algoritmo recursivo para la actualización de los pesos de las neuronas de la red, iniciando por las neuronas de la capa de salida, hasta llegar a las neuronas de la capa de entrada. Para ello utilizar las siguientes fórmulas:

$$\square \quad w_{ij}^h(t+1) = w_{ij}^h(t) + \Delta w_{ij}^h(t+1) \quad (2.42)$$

$$\square \quad \Delta w_{ij}^h(t+1) = \alpha \delta_{ij}^h y_{ij} \quad (2.43)$$

- Repetir el proceso hasta encontrar un término de error que resulte aceptable para los patrones. Este error está calculado por la fórmula:

$$E_p = \frac{1}{2} \sum_{k=1}^M \delta_{pk}^2 \quad (2.44)$$

#### 2.4.6.4. Adición de un Momento en la Regla Delta Generalizada

En el algoritmo de retropropagación del error se utiliza una tasa de aprendizaje. A mayor tasa, mayor es la modificación de los pesos en cada iteración con lo que el aprendizaje será más rápido pero, por otro lado, puede dar lugar a oscilaciones. Para filtrar estas oscilaciones, se añade en la expresión del incremento de los pesos, un término (momento)  $\beta$ , de manera que dicha expresión queda de la siguiente manera:

$$w_{ji}(t+1) = w_{ji}(t) + \alpha \delta_{pj} y_{pi} + \beta (w_{ji}(t) - w_{ji}(t-1)) =$$

$$\Delta w_{ij}(t+1) = \alpha \delta_{pj} y_{pi} + \beta \Delta w_{ji}(t) \quad (2.45)$$

Donde  $\beta$  es una constante que determina el efecto en t+1 del cambio de los pesos en el instante t.

Con este momento se consigue la convergencia de la red en menor número de iteraciones, ya que si en t el incremento de un peso era positivo y en t+1 también, entonces el descenso por la superficie de error en t+1 es mayor. Sin embargo, si en t el incremento era positivo y en t+1 es negativo, el paso que se da en t+1 es más pequeño, lo cual es adecuado

ya que esto significa que se ha pasado por un mínimo y que los pesos deben ser menores para poder alcanzarlo.

#### **2.4.6.5. Método del Gradiente Conjugado**

Otra variante del método clásico del descenso del gradiente es la retropropagación mediante gradientes conjugados. Este algoritmo no involucra el cálculo de las segundas derivadas de las variables y converge al mínimo de la función cuadrática en un número finito de iteraciones, por lo que ofrece una mayor velocidad de aprendizaje, pero no asegura la máxima convergencia, puesto que requiere un ligero incremento en el uso de memoria.

#### **2.4.6.6. Consideraciones sobre el Algoritmo de Aprendizaje**

El algoritmo de backpropagation encuentra un valor mínimo de error (local o global) mediante la aplicación de pasos descendentes (gradiente descendente). Cada punto de la superficie de la función de error corresponde a un conjunto de valores de los pesos de la red. Con el gradiente descendente, siempre que se realiza un cambio en todos los pesos de la red, se asegura el descenso por la superficie del error hasta encontrar el valle más cercano, lo que puede hacer que el proceso de aprendizaje se detenga en un mínimo local de error.

Según se utilice una red neuronal específica, puede también ser distinto el método de estimación de coeficientes utilizado para minimizar la función de error. En estos métodos existe la posibilidad de que el algoritmo de entrenamiento caiga en un mínimo local o algún punto estacionario sin llegar a encontrar el mínimo global. Sin embargo hay que tener en cuenta que no siempre es necesario encontrar el mínimo global, sino que puede ser suficiente con un error menor a un máximo preestablecido.

#### **2.4.6.7. Dimensionamiento de la red**

No se pueden dar reglas concretas para determinar el número de neuronas o de capas de una red para resolver un problema concreto. Lo mismo ocurre con el número de patrones de entrenamiento. En general tres capas son suficientes aunque algunas veces es más fácil que la red aprenda con más de una capa oculta. La cantidad de neuronas ocultas que debe tener cada capa es difícil de determinar, no existe regla que lo haga. Si la red se simula por software, lo mejor siempre es tratar de mantener el número de neuronas por capa oculta lo más bajo que se pueda, ya que cada neurona implica un costo de procesamiento extra.

#### **2.4.6.8. Generalización de la red**

Durante el entrenamiento, la red va asociando las entradas del conjunto de entrenamiento a ciertos patrones de salida deseados mediante un aprendizaje supervisado, intentando de esta manera obtener valores de aproximación y correctos a las salidas reales que se tendrían para aquellos ejemplos de los que no se disponen su correspondiente objetivo.

Esta generalización de resultados es posible mediante tres condiciones:

- Las entradas de la red deben contener información suficiente para poder generar los objetivos correspondientes, es decir, debe existir una relación matemática entre la entrada y la salida. La red no puede generalizar funciones inexistentes.



- La función que la red aprende debe ser continua, al menos continua en trozos.
- La cantidad de ejemplos con los que se ha de entrenar a la red debe ser lo suficientemente significativo y grande.

La causa más común de la pérdida de capacidad de generalización es el sobreaprendizaje. Esto sucede cuando la cantidad de ciclos de entrenamientos tiende a ser muy alta. Se observa que la respuesta de la red a los patrones de entrenamiento es muy buena mientras que la respuesta a nuevos patrones tiende a ser muy pobre. Al aumentar el número de ciclos la red tiende a sobreajustar la respuesta a los patrones de entrenamiento, a expensas de una menor capacidad de generalización.

Para solucionar este problema, se puede aumentar la complejidad de la red. Sin embargo, es imposible determinar a priori cuan grande ha de ser la red para que no se presente sobreaprendizaje en una determinada aplicación. Es por ello por lo que existen otros dos métodos que se pueden utilizar para mejorar el rendimiento de las redes son la regularización y la parada temprana.

La regularización se trata de modificar la función de error, que normalmente es el error cuadrático medio de los errores de entrenamiento, añadiendo la media del error cuadrático medio de los pesos y umbrales, obligando de esta manera a la red a tener pesos y umbrales pequeños y reduciendo el sobreaprendizaje. En la parada temprana los ejemplos del entrenamiento se dividen en dos grupos, el conjunto de entrenamiento y el conjunto de validación que se utiliza para simular la red entrenada con ejemplos nuevos y determinar si la red generaliza correctamente mediante un error de validación, si este valor sube es un indicador de que la red esta sobre aprendiendo y habrá que finalizar el entrenamiento. En esta última técnica se recomienda utilizar un número elevado de neuronas ocultas y algoritmos lentos para evitar que la red pare su entrenamiento sin terminar su aprendizaje.

### 3. Instrumentación

En este capítulo se presenta en la sección 3.1 un resumen de las actividades efectuadas por A. Zuñiga [64] para obtener las imágenes de las huellas de partículas alfa con incidencia normal sobre el polímero detector CR-39 trabajadas en esta tesis. En la sección 3.2 se muestra detalladamente como se determino las energías de incidencia de las partículas alfa sobre el polímero detector.

#### 3.1. Banco de datos de Imágenes

Las imágenes trabajadas en esta tesis forman parte de una amplia base de imágenes obtenidas durante la realización de una tesis doctoral [64], donde se estudio las principales características dejadas por las partículas alfa al incidir sobre el polímero detector CR-39, usando películas de Uranio y Torio como fuentes emisoras de partículas alfa. Las fuentes se montaron de tal forma que se podía fijar la energía y el ángulo de incidencia de las partículas alfa, en este trabajo solo se tomo en cuenta las imágenes de huellas alfa obtenidas por incidencia normal sobre el detector.

La base de imágenes de huellas alfa de incidencia normal esta conformada por un total de 608 imágenes, de las cuales 342 imágenes fueron obtenidas usando fuentes de Uranio y el resto (266 imágenes) usando fuentes de Torio.

En la figura 3.1 se muestra una imagen típica de esta base, etiquetada con el nombre “E4A9032”, que hace referencia a huellas de partículas alfa con energía de referencia de 4 MeV.

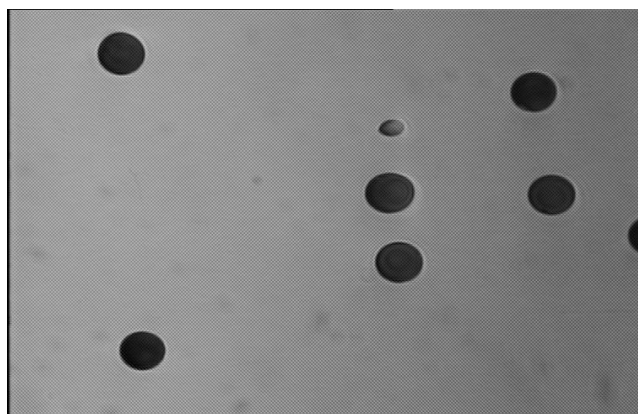


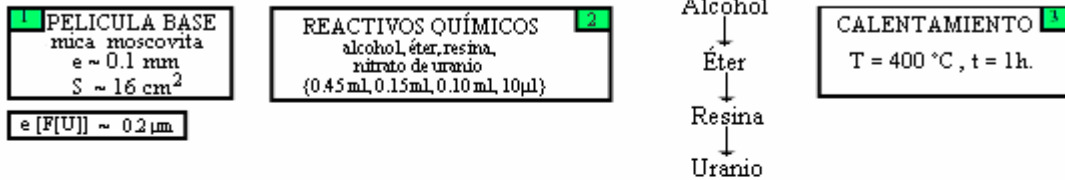
Fig. 3.1. Imagen de huellas nucleares alfa con una energía de referencia de 4 MeV [64].

##### 3.1.1. Fuentes emisoras alfa

Para obtener imágenes de huellas nucleares alfa se usaron fuentes emisoras de partículas alfa compuestas por películas de Uranio y Torio de espesores menores de 0.2  $\mu\text{m}$ . Estas películas fueron fabricadas usando la metodología presentada por Yagoda [65]. Esta consiste en verter una solución de nitrato de uranio o de oxido de torio (que contienen 50 ppm de uranio natural y 100 ppm de oxido de torio natural respectivamente) mezclada con una resina (parlodio disuelto en éter o alcohol en el caso de películas de

uranio o solo en alcohol para el caso de películas de torio) sobre una base de mica muscovita, de un área aproximada de  $4 \times 4 \text{ cm}^2$ , después de eso se evapora el éter o alcohol durante 24 h y finalmente se lleva la película a un horno a  $400 \text{ }^\circ\text{C}$ , por 1 h. La figura 3.2 muestra de forma esquemática el procedimiento seguido y las cantidades correspondientes para producir estas películas.

### FUENTE DE URANIO



### FUENTE DE TORIO

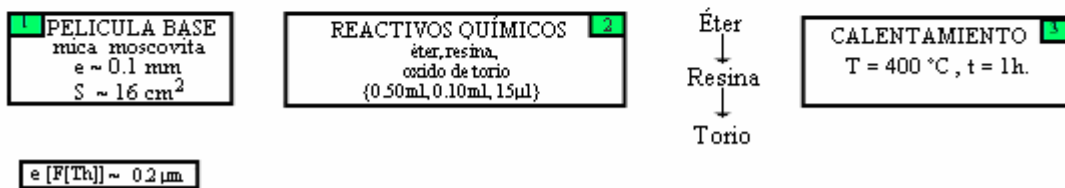


Fig. 3.2. Procedimiento de preparación de las fuentes emisoras alfa de Uranio y Torio, utilizando la metodología de Yagoda [65].

### 3.1.2. Montaje de irradiación.

El montaje de irradiación es una configuración fuente-detector que permite definir las energías y ángulos de incidencia de las partículas alfa que inciden sobre el detector. La figura 3.3 muestra la geometría de este montaje.

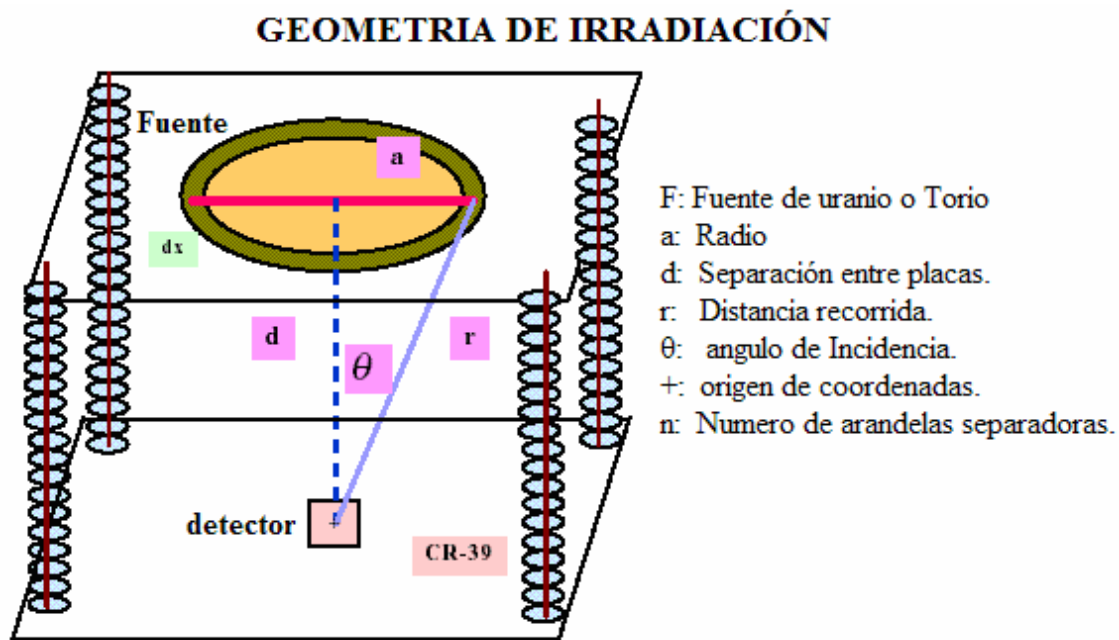


Fig. 3.3. Montaje de irradiación el cual permite establecer la energía y ángulo de incidencia de las partículas alfa que inciden sobre el detector CR-39 [64].

El montaje de irradiación consiste de 2 placas paralelas de acrílico (idénticas), una superior donde se fija la fuente emisora de partículas (en una corona de radio  $a$  y ancho  $dx$ ) y otra inferior que contiene el detector. La fuente y detector están alineados en un eje vertical que pasa por el centro de ambas placas.

La energía queda definida a partir del valor de separación fuente detector,  $r$ , dado por:

$$r = (a^2 + d^2)^{1/2} \quad (3.1)$$

El ángulo de incidencia de las partículas es dado por:

$$\theta = \text{ArcCos}(d/r) \quad (3.2)$$

Para el caso de incidencia normal  $\theta = 0^\circ$ .

El parámetro  $d$ , utilizado en las ecuaciones 3.1 y 3.2, depende de la separación entre las placas.

Para la separación de las placas se utilizó arandelas metálicas de espesor uniforme de  $0.822 \pm 0.0001$  mm. El número de arandelas,  $n$ , es definido a partir de la energía y ángulo que se busca considerando el número entero mas próximo. Se debe notar que el parámetro  $n$  fue utilizado solamente para construir el montaje de irradiación. Las energías y ángulos utilizados fueron obtenidos a partir de mediciones directas de los parámetros  $a$  y  $d$ . En las tablas 9.1 y 9.2 del anexo se muestran las distintas distancias de separación entre la fuente y el detector para el caso de fuentes de uranio y torio respectivamente.

### **3.1.3. Ataque químico de los detectores.**

Los detectores utilizados (CR-39 de espesor  $500 \mu\text{m}$ ) fueron atacados con una solución de hidróxido de sodio  $6.25 \text{ N}$  ( $50 \text{ g}$  de  $\text{NaOH}$ , diluidos en  $200 \text{ ml}$  de  $\text{H}_2\text{O}$ ) a una temperatura de  $71^\circ$  por un tiempo de  $400 \text{ min}$ .

Durante el ataque la solución fue mantenida sin agitación. Después del ataque químico los detectores fueron lavados con agua común inicialmente a la misma temperatura del ataque ( $71^\circ\text{C}$ ). Después a la temperatura ambiente ( $20^\circ\text{C}$ ). El tiempo de lavado fue de  $20$  minutos.

### **3.1.4. Adquisición de las imágenes de las huellas.**

Para la adquisición de las imágenes se utilizó un sistema constituido por un microscopio LEITZ con aumentos de  $20x$ ,  $40x$ ,  $63x$  y  $100x$  en los objetivos y  $10x$  en el ocular y un computador acoplado a una cámara digital (CCD-IRIS; Sony, Modelo: DXC107A).

Antes de capturar las imágenes, El Sistema de digitalización de imágenes fue calibrado con un aumento conveniente, que en este caso fue  $63x10$  para el tiempo de ataque químico de  $400$  minutos, los resultados de la calibración fueron: en la horizontal  $4.71 \text{ pixels}/\mu\text{m}$  y en la vertical  $4.055 \text{ píxel}/\mu\text{m}$ , y un área de campo de  $1.924 \times 10^{-4} \text{ cm}^2$ .

## 3.2. Definición de las energías

El intervalo de energía a estudiar cubre gran parte de las energías de las partículas alfa naturales, cuyo mayor valor proviene de la cadena de Torio (isótopo Po-212, de 8.78 MeV). La tabla 3.1 muestra las energías de las partículas alfa emitidas por las películas de Uranio y Torio.

Tabla 3.1. Energías de las partículas alfa emitidas por las cadenas de decaimiento de Uranio y Torio [66].

Fuentes de Uranio			Fuentes de Torio		
Nucleido	Energía (MeV)	Canal(%)	Nucleido	Energía (MeV)	Canal (%)
U-234	4.777	72.5	Th-232	4.017	77.0
	4.724	27.5		3.957	23.0
U-238	4.196	77.0		3.834	0.2
	4.149	23.0	Th-228	5.421	72.7
U-235	4.397	57.0		5.339	26.7
	4.367	18.0		5.210	0.4
	4.217	5.7		5.172	0.2
	4.598	4.6		5.137	0.1
	4.557	3.7	Ra-224	5.686	94.0
	4.324	3.0		5.449	5.5
	4.345	1.5	5.161	0.0	
4.503	1.2	Rn-220	6.288	99.9	
4.267	0.5		5.747	0.1	
4.158	0.5	Po-216	6.779	99.0	
		5.985	0.0		
		Po-212	8.784	100.0	
		Bi-212	6.090	27.2	
			6.051	69.9	
			5.769	1.7	
			5.626	0.2	
			5.607	1.1	
		5.486	0.0		

En el caso de las fuentes de Uranio para construir los montajes se utilizó como energía de referencia  $E_0 = 4.5$  MeV. Como el Uranio utilizado en la producción de las fuentes está en concentraciones isotópicas, 98% de las partículas alfa son debidas al U-234 y al U-238 (que está en equilibrio secular). En el caso de estas fuentes, despreciaremos las partículas alfa del U-235 y asumimos como energías de salida partículas de energía  $E_1 = 4.2$  MeV y  $E_2 = 4.8$  MeV.

Las energías de salida en el caso de películas de Torio, fueron obtenidas agrupando aquellas que presentan diferencias menores que 100 keV y dejando de considerar a las que poseen rendimiento menor que el 1%. Por lo tanto los grupos resultantes fueron  $E_0 = E_1 = 8.78$  MeV,  $E_2 = 6.77$  MeV,  $E_3 = 6.28$  MeV,  $E_4 = 5.52$  MeV y  $E_5 = 4.0$  MeV. Se tomó como referencia  $E_0 = 8.78$  MeV.

### 3.2.1. Determinación de la energía de incidencia

Para determinar la energía de incidencia de las partículas alfa sobre el polímero detector CR-39 se hizo uso del programa TRIM (“*the TRansport of Ions in Matter*”), el cual es un subprograma del paquete SRIM [67] (“*the Stopping and Range of Ions in Matter*”), ver figura 3.3, con el cual podemos calcular el poder de frenado y rango de iones en un material en un rango de energía entre 10 eV y 2 GeV, usando un tratamiento

mecanocuántico de las colisiones entre iones y átomos. se puede descargar libremente de [68].

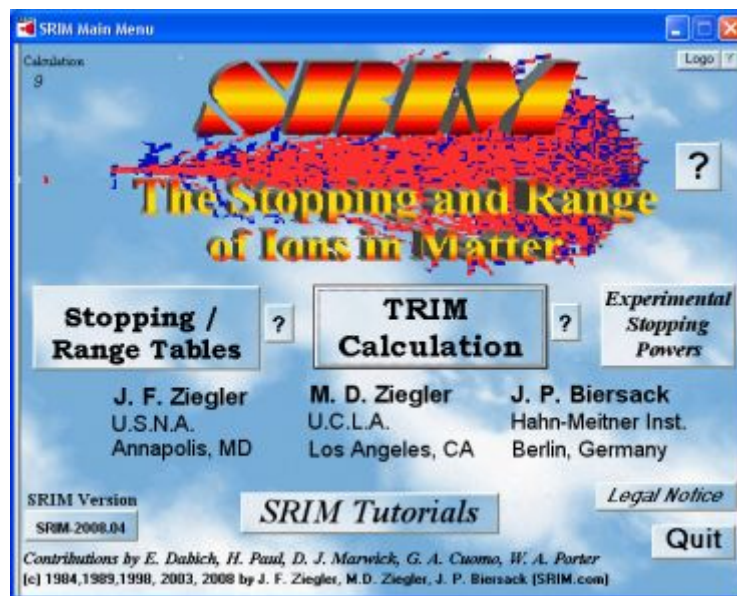


Fig. 3.3. El menú principal del Programa SRIM

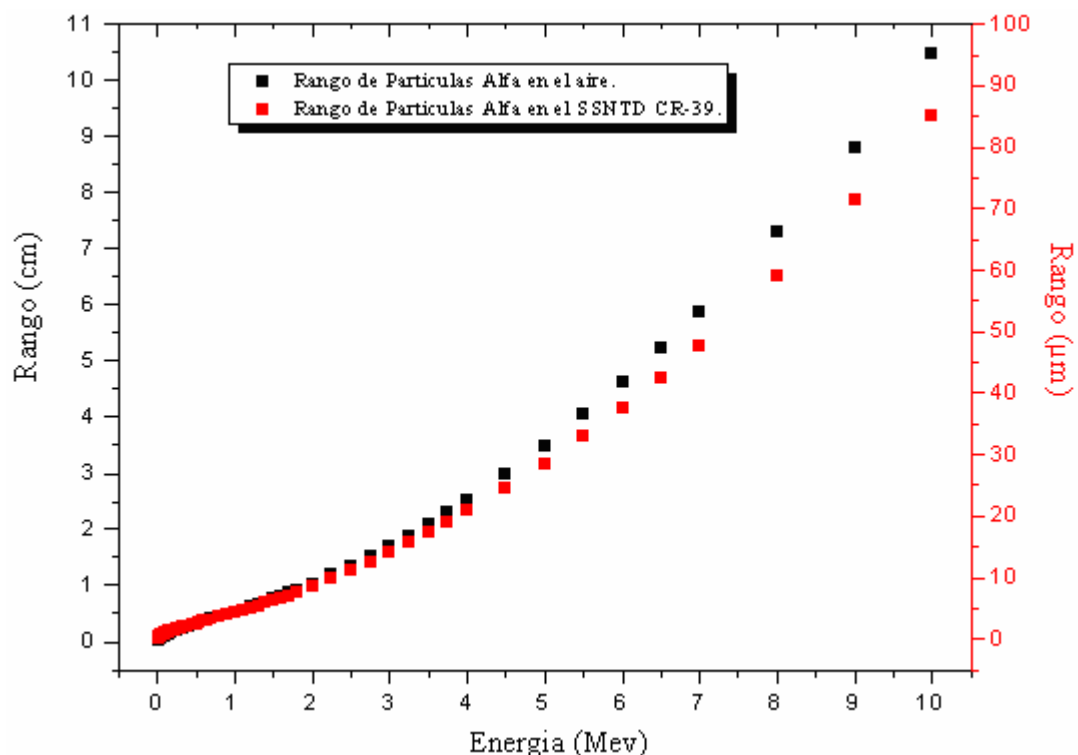


Fig. 3.4. Rango de Partículas alfa en aire y el SSNTD CR-39 determinado usando SRIM.

En la figura 3.4. Se puede apreciar el rango de partículas alfa en aire y en el polímero CR-39 determinado usando el programa SRIM.

En las figuras 3.5 y 3.6 se muestran las pérdidas de energía de partículas alfa de distintas energías emitidas por las fuentes de Uranio y Torio al desplazarse a través de una capa de aire, obtenidas haciendo uso TRIM.

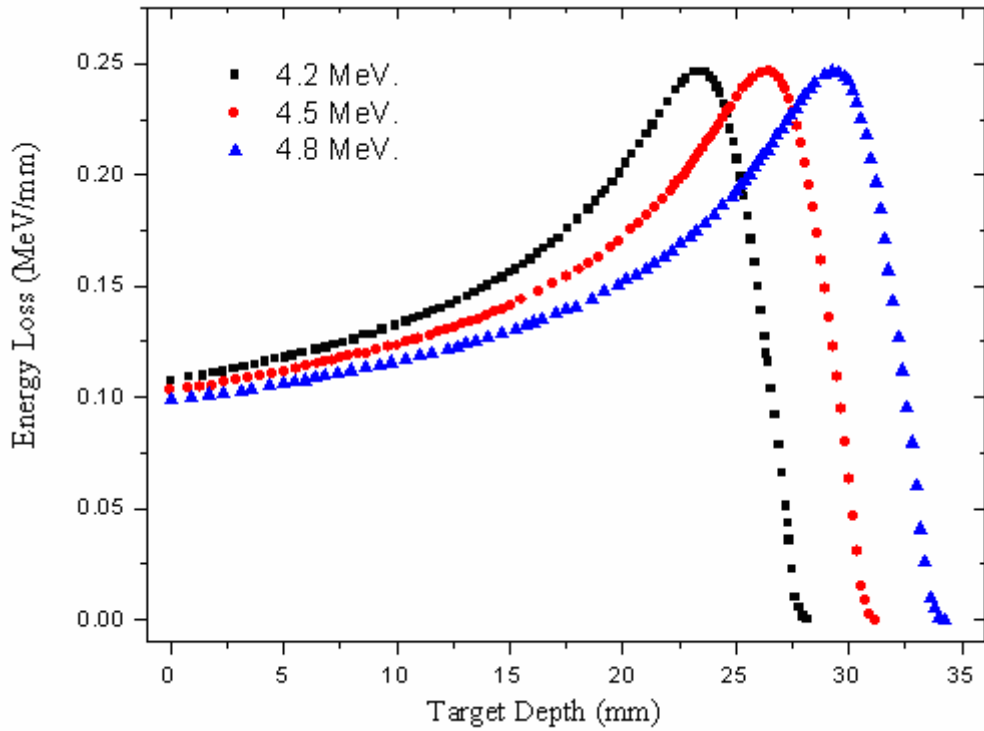


Fig. 3.5. Pérdida de energía de partículas alfa en aire emitidas por las fuentes de uranio.

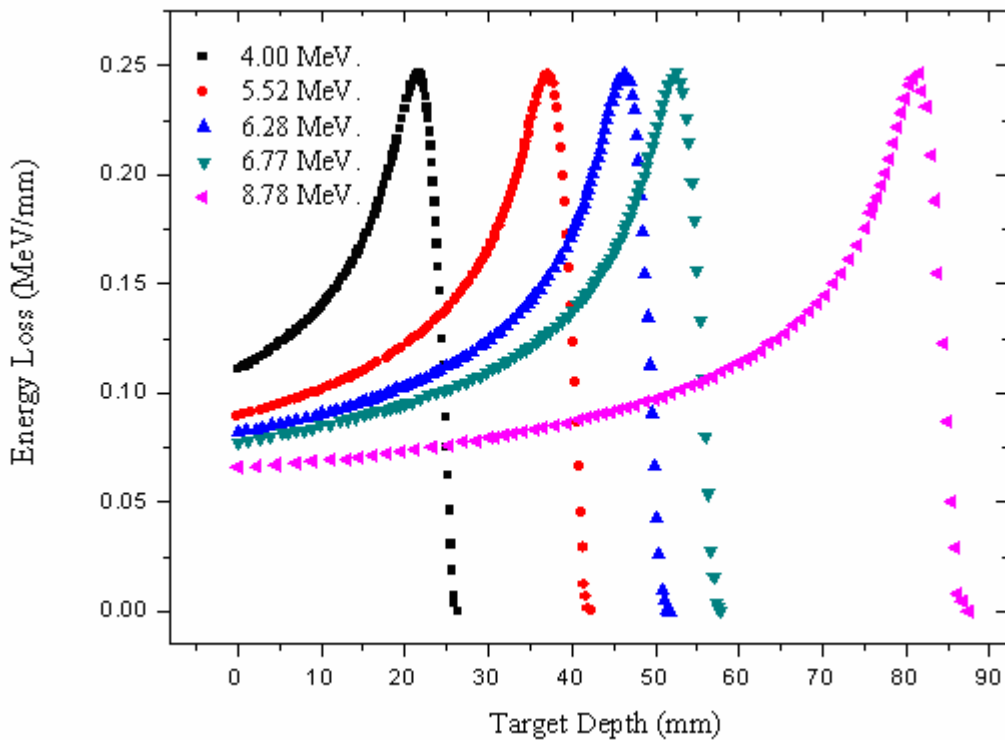


Fig. 3.6. Pérdida de energía de partículas alfa en aire emitidas por las fuentes de torio.

De las curvas de las figuras 3.5 y 3.6 obtenemos la energía de incidencia  $E(x)$  sobre el detector, los resultados se muestran en las figuras 3.7 y 3.8.

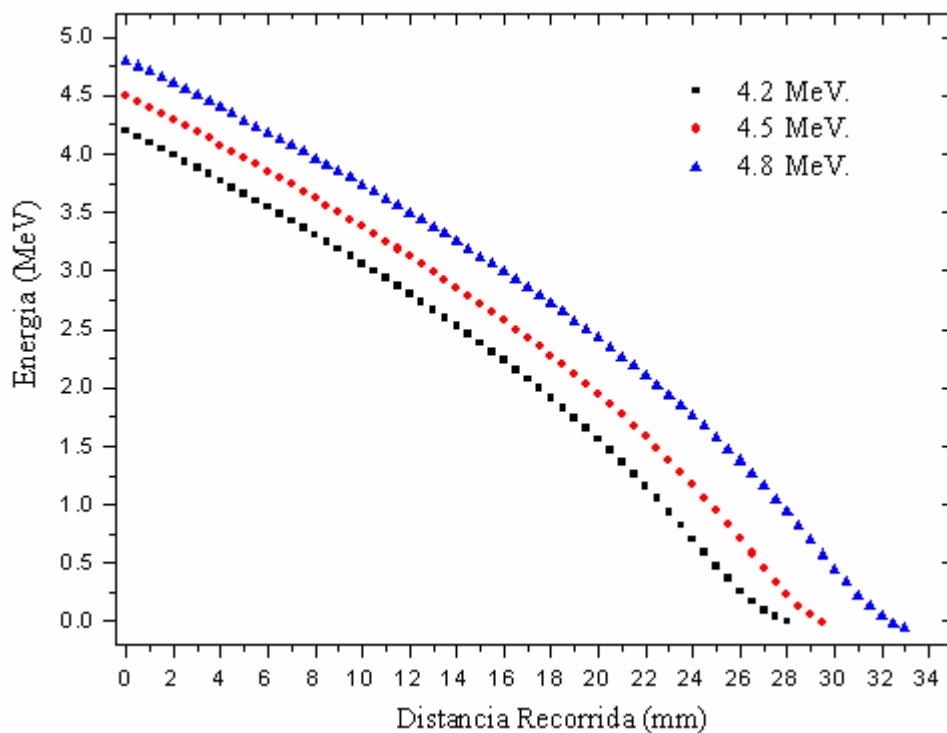


Fig. 3.7. Curvas distancia recorrida-energía residual de partículas alfa en el aire emitidas por las fuentes de uranio.

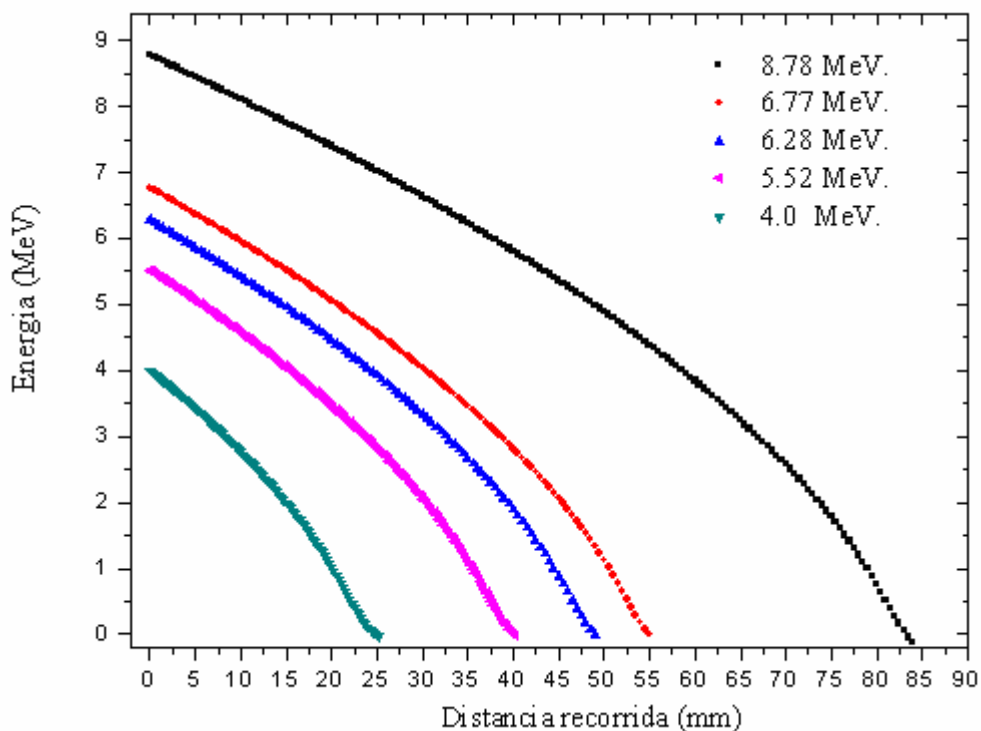


Fig. 3.8. Curvas distancia recorrida-Energía residual de partículas alfa en el aire emitidas por las fuentes de torio.

Las curvas mostradas en las figuras 3.7 y 3.8 fueron obtenidas al resolver la ecuación:



$$E(x) = E_0 - \int_0^x \frac{dE}{dx} dx \quad (3.3)$$

Donde  $E_0$  es energía inicial de las partículas alfa. La integral de la ecuación anterior fue determinada numéricamente.

Finalmente a partir de las curvas mostradas en la figuras 3.7 y 3.8 y con los datos de distancias fuente detector indicados en el anexo 9.1 podemos hallar las energías de incidencia de las partículas alfa sobre el polímero detector, los resultados se muestran en las tablas 3.2 y 3.3

Tabla 3.2. Energías de incidencia de partículas alfa provenientes de las fuentes de Uranio.

<b>E(MeV)</b>	<b>r(mm)</b>		<b>E<sub>i</sub>(MeV)</b>
4.39	1.07	0.01	4.09 4.69
4.22	2.66	0.03	3.91 4.53
3.95	5.13	0.02	3.62 4.28
3.57	8.46	0.06	3.22 3.91
3.27	10.9	0.03	2.90 3.63
2.84	14.14	0.06	2.43 3.23
2.49	16.61	0.07	2.04 2.91
1.95	19.95	0.09	1.43 2.43
0.80	25.63	0.11	0.15 1.45

Tabla 3.3. Energías de incidencia de partículas alfa provenientes de las fuentes de Torio.

<b>E(MeV)</b>	<b>r(mm)</b>		<b>E<sub>i</sub>(MeV)</b>
8.0	11.61	0.03	7.99 5.82 5.27 4.38 2.47
6.9	25.72	0.05	6.94 4.47 3.80 2.64
5.9	39.04	0.04	5.89 2.94 1.90 0.16
4.8	50.41	0.16	4.82 0.78

Donde para ambas tablas tenemos:

- $E_i$  : Energía de las partículas que llegan al detector.
- $E$  : Energía de referencia para construir el montaje de irradiación
- $r$  : Distancia fuente-detector.

## 4. Conteo y clasificación de huellas nucleares alfa.

En este capítulo se discute en detalle la metodología propuesta para el conteo y clasificación de huellas nucleares alfa.

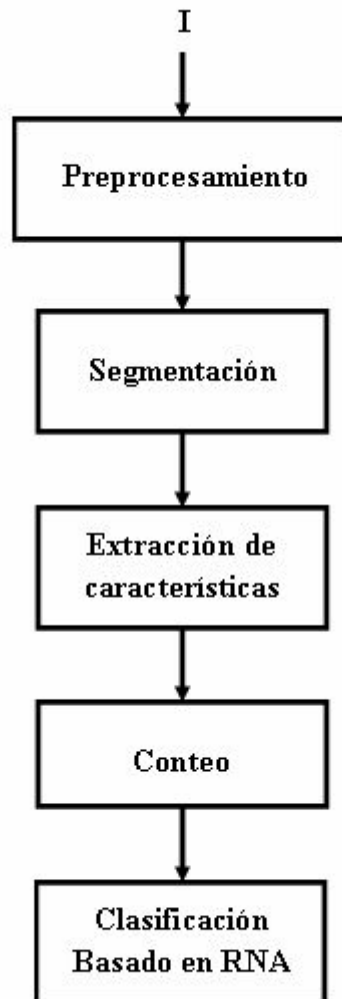


Fig. 4.1. Diagrama a bloques del Sistema propuesto

El procedimiento general inicia cuando se recibe como entrada una imagen digital de huellas nucleares alfa (figura 4.2), la cual es procesada en las primeras tres etapas (Preprocesamiento, segmentación y extracción de características), como se muestra en la figura 4.1, para obtener como resultado los patrones de las huellas, los cuales serán introducidos a una red neuronal artificial en la etapa de clasificación.

A continuación se explicará en detalle cada una de las etapas del sistema propuesto

### 4.1. Preprocesamiento

La finalidad de esta etapa es eliminar aquellos elementos en la imagen que puedan distorsionar el proceso de identificación de las huellas alfa. Se puede adquirir ruido durante el procedimiento de captura y digitalización de la imagen dado que se pueden presentar variaciones en iluminación, mala configuración o mal funcionamiento del sistema de captura etc. distorsionando las propiedades de la imagen digitalizada.

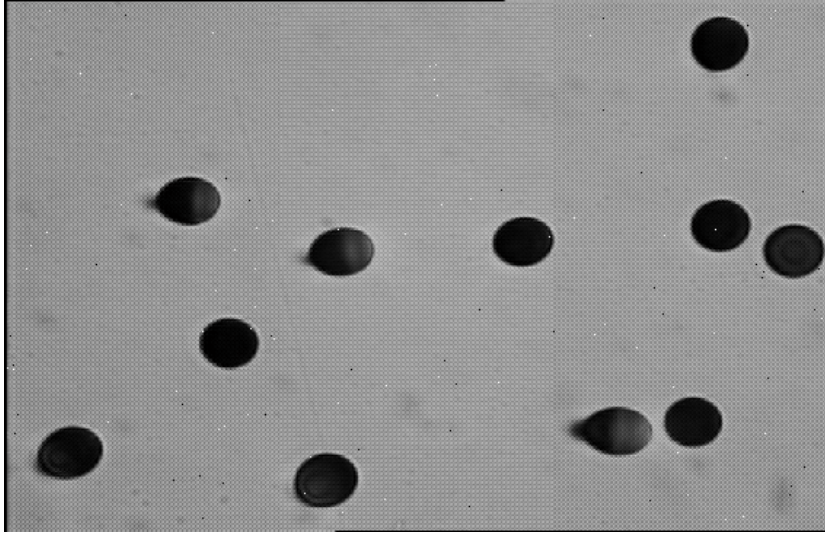


Fig. 4.2. Imagen Original de entrada

En la figura 4.2 se muestra una de las 608 imágenes que conforman la base de datos de imágenes analizadas, todas las imágenes tienen las mismas dimensiones [756,486] píxeles y están digitalizadas con 8 bits de resolución.

#### 4.1.1. Eliminación del ruido

En la figura 4.3 se muestra el procedimiento seguido con la finalidad de eliminar el ruido presente en la imagen. Se aplicó 2 tipos de filtros: el filtro de la mediana y el filtro de gauss.

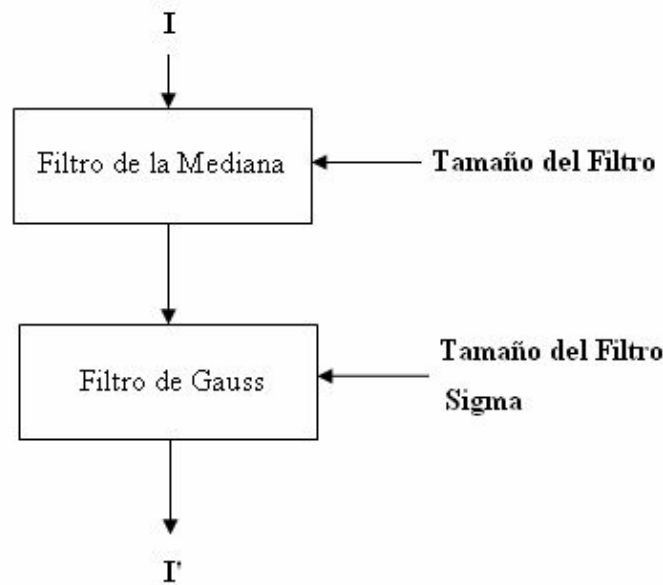


Fig. 4.3. Diagrama de flujo del procedimiento seguido en la eliminación de ruido.

El filtro de la mediana es un filtro no lineal, empleado frecuentemente para eliminar el ruido de alta frecuencia sin eliminar las características significativas de la imagen. En este trabajo se emplea una ventana (máscara) de 3x3, la cual, se centra en cada píxel de la imagen, reemplazando el píxel central por la mediana de los nueve píxeles que cubre la ventana. El tamaño de la ventana nos permite preservar las características propias de la imagen y a su vez eliminar las altas frecuencias.

De igual manera el filtro Gaussiano es un filtro lineal utilizado para remover el ruido, además de suavizar la imagen, para aplicarlo es necesario generar un espacio Gaussiano, el cual es convolucionado con la imagen original, tal que:

$$W(i, j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}} \quad (4.1)$$

Donde  $\sigma$  es la desviación estándar. A mayor  $\sigma$  mayor suavizado. Sin embargo, si el valor de  $\sigma$  es muy grande pueden perderse detalles en la imagen por lo cual se elige un valor que filtre lo suficiente, sin que elimine información de la imagen,  $W(i, j)$  es el valor del píxel en la posición  $(i, j)$ .

La figura 4.4 se muestra la aplicación sucesiva de los filtros de la mediana y de gauss. En (a) se muestra el filtro de la mediana de la imagen original, ver figura 4.2, con un tamaño de ventana de 3x3, mientras en (b) se muestra el filtro de gauss de la imagen en (a) con un tamaño de ventana de 3x3 y  $\sigma = 0.5$ . Se puede observar que la imagen de salida final, imagen (b), preserva los detalles después de la aplicación sucesiva de los dos filtros.

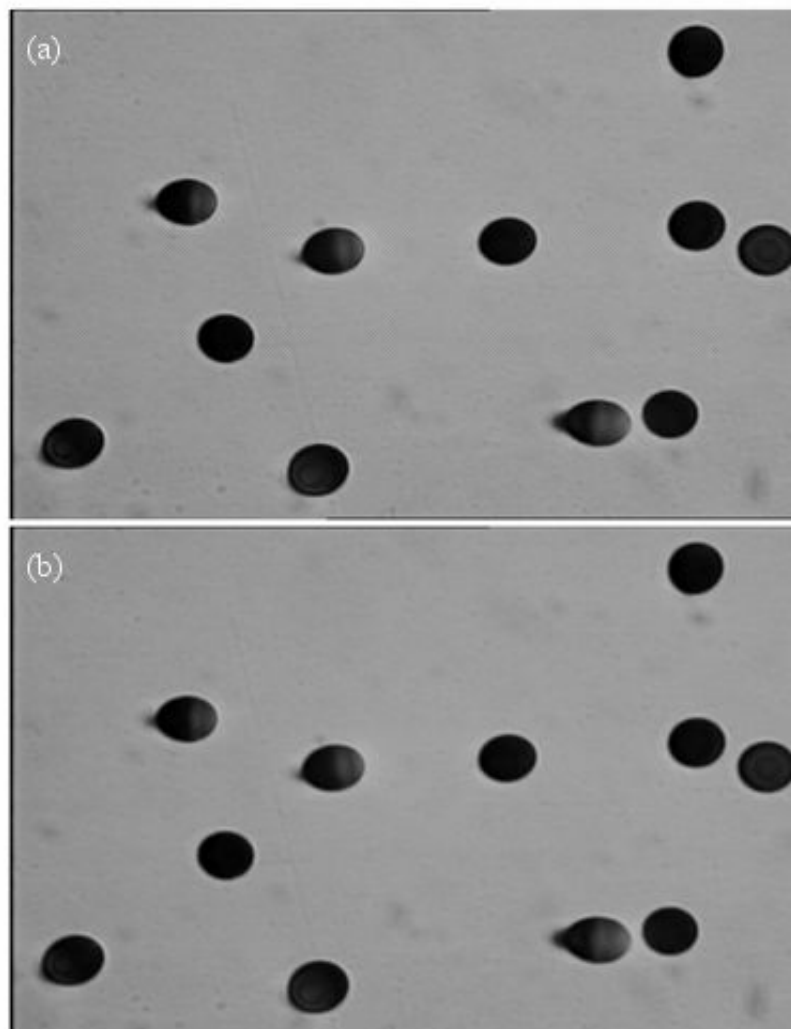


Fig. 4.4. Eliminación del ruido, usando: (a) Filtro de la mediana. (b) Filtro de gauss.

### 4.1.2. Mejorando el contraste de la imagen.

El procedimiento seguido para mejorar el contraste de la imagen consta de 2 pasos como se muestra en la figura 4.5. La primera parte se hace uso de una transformación lineal para expandir los píxeles usados en la representación de la imagen en todo el rango de valores posibles de píxeles y la segunda parte usa una combinación de las transformaciones morfológicas top-hat y bottom-hat.

El procedimiento recibe como imagen de entrada la imagen  $I'$  que es la salida del procedimiento de eliminación de ruido del apartado anterior.

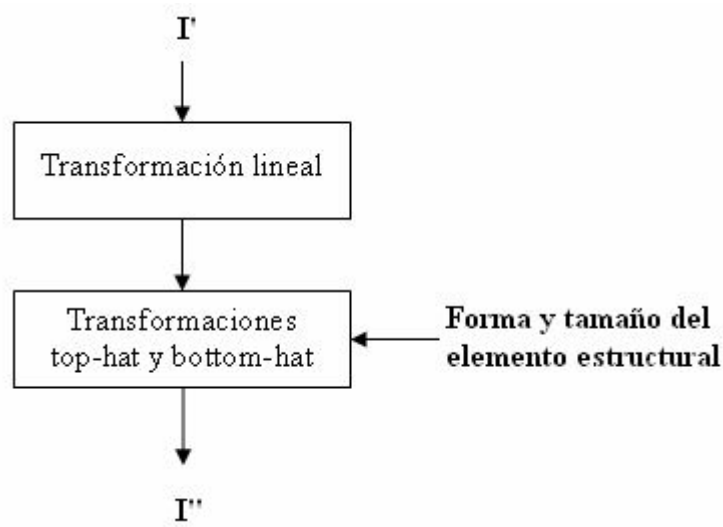


Fig. 4.5. Diagrama de flujo del procedimiento seguido para la mejora del contraste de la imagen.

El contraste de la imagen de entrada se puede mejorar si logramos expandir los valores de intensidad sobre todo el rango de valores posibles, como vemos en la figura 4.6 (a), del histograma de la imagen, solo una parte de todos los valores posibles de intensidad de los píxeles son usados en la representación de la imagen, para expandir a todo el rango posible usamos transformación lineal como la mostrada en la figura 4.6 (b), donde los límites  $Low\_in$  y  $High\_in$  especifican el 1% del menor y mayor valor de todos los valores de los píxeles. Las intensidades se expanden de acuerdo a cierto factor o pendiente; el factor para este aumento lineal de contraste en este caso es:

$$C(x, y) = \left( \frac{I(x, y) - Low\_in}{High\_in - Low\_in} * 255 \right) \quad (4.2)$$

Donde  $I(x,y)$  es la imagen a procesar y  $C(x,y)$  es la imagen con un aumento lineal de contraste. El resultado de la aplicación de la transformación lineal se muestra en la figura 4.7 (a).

La otra técnica empleada en este trabajo para mejorar el contraste combina el uso de las transformaciones morfológicas top-hat y bottom-hat. El resultado de la aplicación de las transformaciones morfológicas se muestra en la figura 4.7 (b).

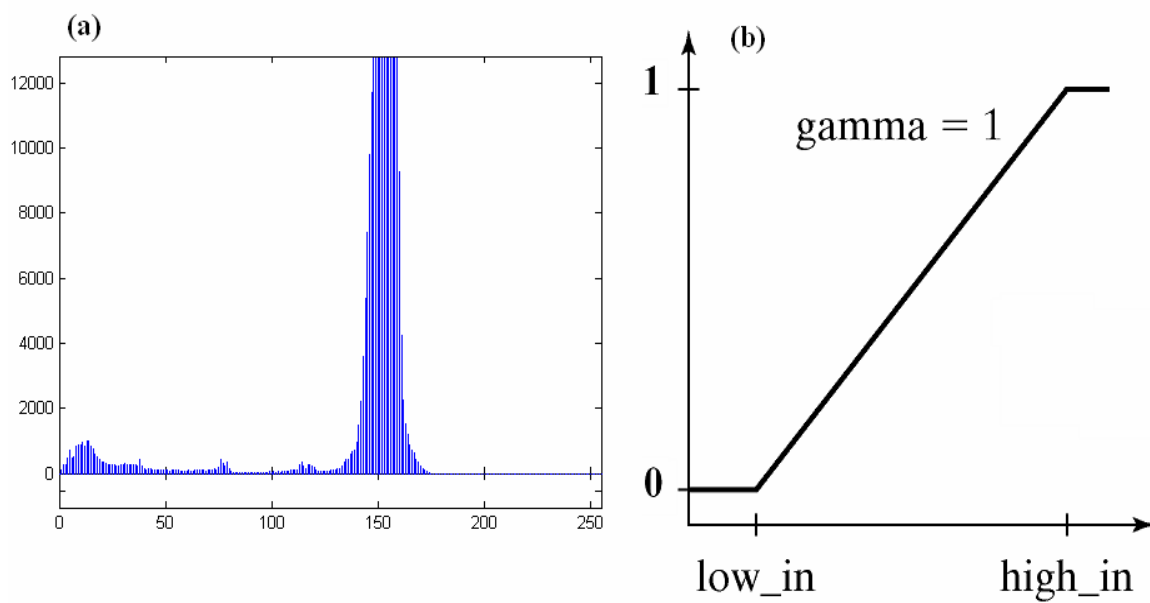


Fig. 4.6. (a) Histograma de la Imagen  $I'$ . (b) Transformación lineal.

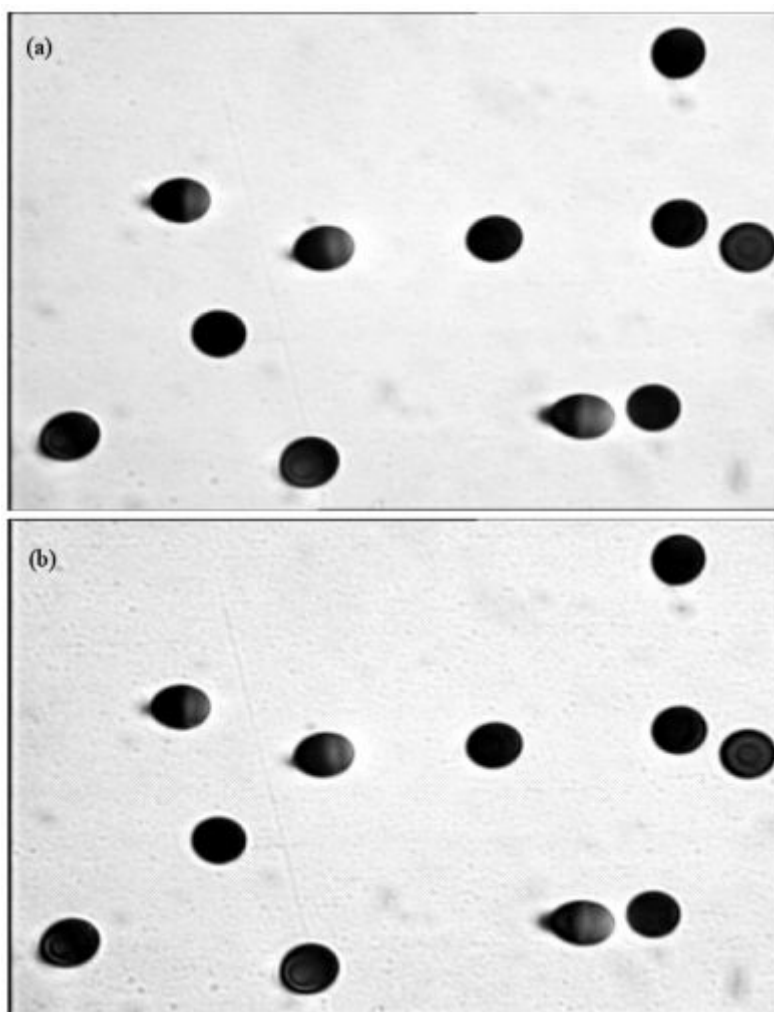


Fig. 4.7. Imagen con el contraste mejorado (a) Transformación lineal, (b) Usando las transformaciones top-hat y bottom-hat

La transformación top-hat es definida como la diferencia entre la imagen original y el *opening* de la imagen original por otro lado la transformación bottom-hat es definida como la diferencia entre *closing* de la imagen original y la imagen original. Para maximizar el contraste entre los objetos de interés y el entorno que los separan unos de otros, la imagen “bottom-hat” es restada de la suma entre la imagen original con la imagen top-hat. Ya que los objetos de interés en nuestra imagen tienen la forma de discos, utilizamos un elemento estructural en forma de disco con un radio de 5 píxeles

Finalmente en la figura 4.8 se puede ver el histograma de la imagen de salida  $I''$  del procedimiento de mejora del contraste, se puede observar que los valores de los píxeles que representan la imagen están distribuidas a lo largo del todo el rango posible de representación.

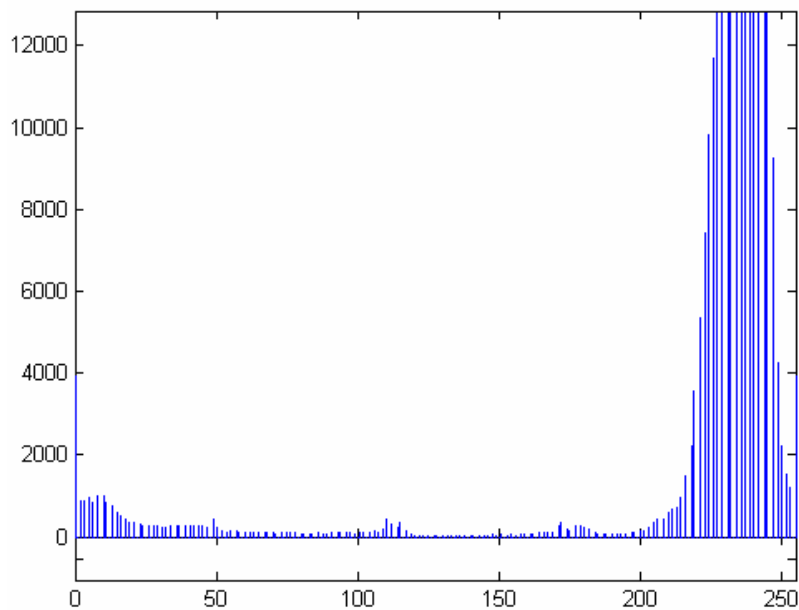


Fig. 4.8. Histograma de la imagen  $I''$

## 4.2. Segmentación

La segmentación es la división de la imagen en sus partes constituyentes u objetos [56]. El propósito de la segmentación de una imagen consiste en dividirla en varias áreas significativas, donde el objeto de interés esté contenido, de modo que el objeto de interés puede ser sujeto de un mejor análisis o una mejor interpretación, y puede extraerse de escenas más complejas. Por lo que la segmentación representa una de las primeras etapas en el análisis de identificación de una imagen. En general, la segmentación automática es una de las tareas más difíciles en el procesamiento digital de imágenes. Una buena segmentación puede conducir al éxito mientras que una segmentación errónea casi siempre garantiza el fracaso en algún problema de visión.

En la figura 4.9 se muestra el procedimiento seguido en la segmentación de la imagen. En este caso el procedimiento toma la imagen  $I''$ , imagen de salida de la etapa de preprocesamiento, como imagen de entrada y devuelve como salida ventanas (subimágenes) conteniendo los objetos o huellas detectadas. Las ventanas se extraen desde la imagen original  $I$  y desde la imagen binarizada IBF. La imagen Binaria IBF es resultado de la adición de las 2 imágenes binarias obtenidas siguiendo 2 de los



procedimientos de segmentación más usados como son la binarización y la detección de contornos, como se muestra en la figura 4.9.

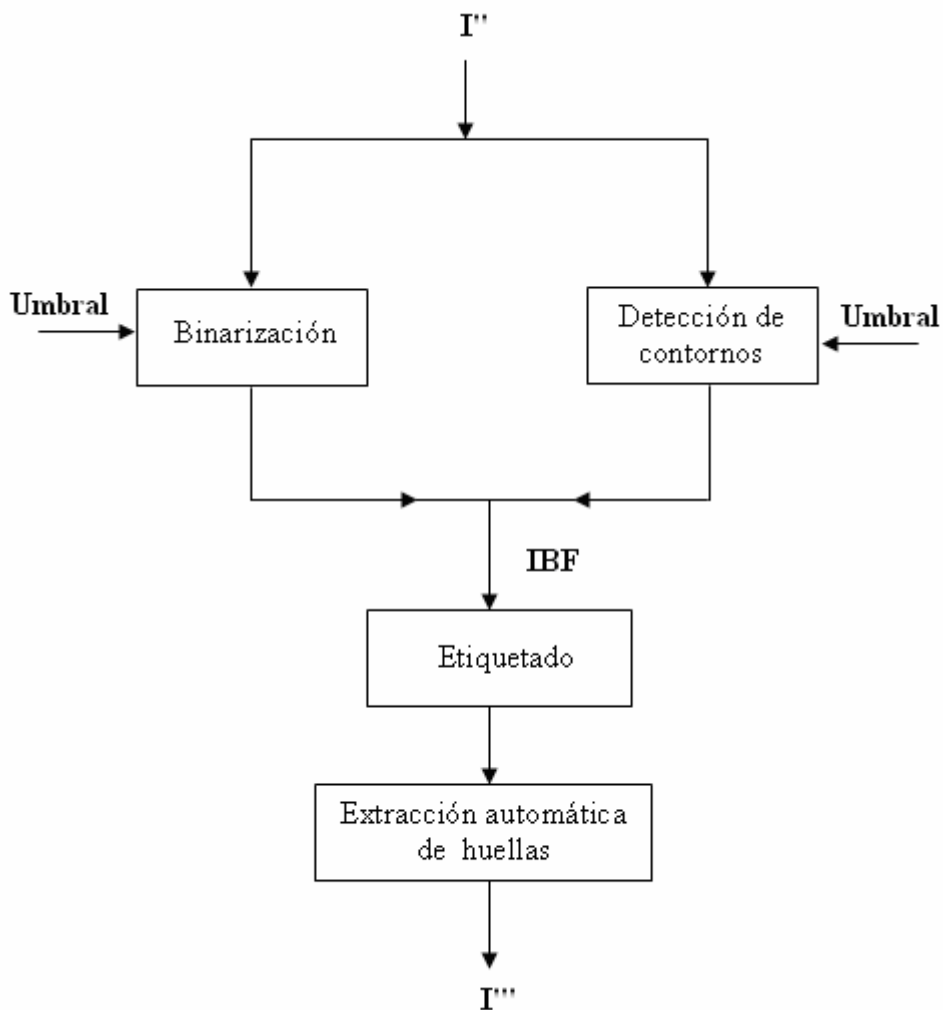


Fig. 4.9. Diagrama de flujo del procedimiento seguido para segmentación de la imagen.

#### 4.2.1. Binarización

El objetivo del proceso de binarización es la de obtener una imagen en blanco y negro, a partir de una imagen con niveles de gris. El problema central en el proceso de binarización es ubicar un umbral adecuado que permita separar los píxeles de tal manera que aquellos que toman valores menores que el umbral correspondan a 0 y aquellos que toman valores mayores o iguales que el umbral correspondan a 1.

Existen diferentes formas de asignar el umbral, tanto manual como automáticamente. En este trabajo el umbral fue calculado automáticamente usando el método de Otsu.

El método de Otsu [69] es un algoritmo para la determinación automática de un umbral de binarización a partir del histograma de una imagen. Si consideramos que el conjunto de niveles de gris que conforman una imagen son  $G = \{0, 1, 2, \dots, L\}$ , siendo  $L$  el valor máximo de niveles de gris, el umbral que vamos a obtener se considera el punto que parte los píxeles de una imagen en dos clases de niveles de gris  $C_0 = \{0, 1, \dots, t\}$  y  $C_1 = \{t + 1, t + 2, \dots, L\}$ , donde  $t$  es el umbral óptimo que maximiza la separabilidad de estas dos

clases.

En este trabajo se desarrollaron 2 algoritmos para la binarización de la imagen, donde la diferencia entre estos 2 algoritmos radica en la forma en la que se obtiene el umbral de binarización. En el primer algoritmo se desarrolla la binarización tomando el umbral global de la imagen mientras que el segundo algoritmo la imagen es dividida en 16 subimágenes y se binariza cada subimagen tomando como umbral el promedio entre el umbral global y el umbral de cada subimagen finalmente se reagrupa las 16 subimágenes para obtener la imagen binarizada final, en la figura 4.10 se puede ver el la imagen binarizada al aplicar este último algoritmo.

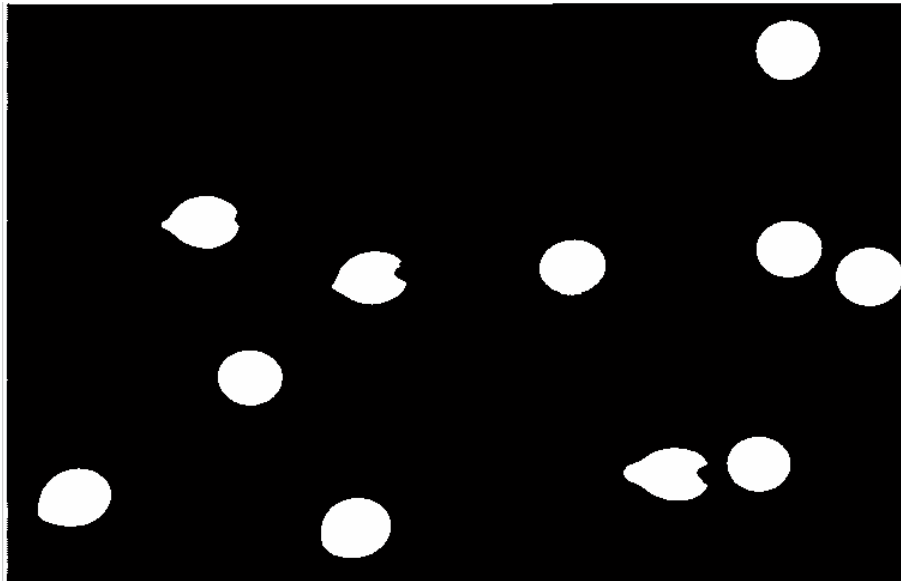


Fig. 4.10. Imagen Binarizada, usando un umbralización ponderada.

#### 4.2.2. Detección de contornos

Un borde en una imagen, es un límite o contorno en el cual ocurren cambios significativos en algún parámetro físico de la imagen, tal como, la reflectancia superficial, la iluminación o la distancia de la superficie visible al observador. Los cambios en los parámetros físicos de la imagen se manifiestan de diversas formas, incluyendo cambios en la intensidad, color, y textura.

La detección de contornos en una imagen se realiza mediante el uso de filtros detectores de contornos (bordes), cuyo objetivo, es el de destacar los detalles finos de una imagen o intensificar detalles que hayan sido difuminados, bien sea por la naturaleza de la imagen o bien por el mecanismo de adquisición en la misma. En general, los bordes de objetos en una imagen se pueden diferenciar por los cambios más o menos abruptos de un valor entre dos o más píxeles adyacentes.

En este trabajo para determinar los bordes de la imagen se usó el método de Sobel con nivel de sensibilidad es decir el algoritmo ignora los píxeles del borde encontrados con nivel de gris menores que un valor prefijado, en este caso el valor de sensibilidad es dado por una fracción del umbral global de la imagen (0.15 veces el umbral). En la figura 4.11 se puede ver los bordes encontrados.

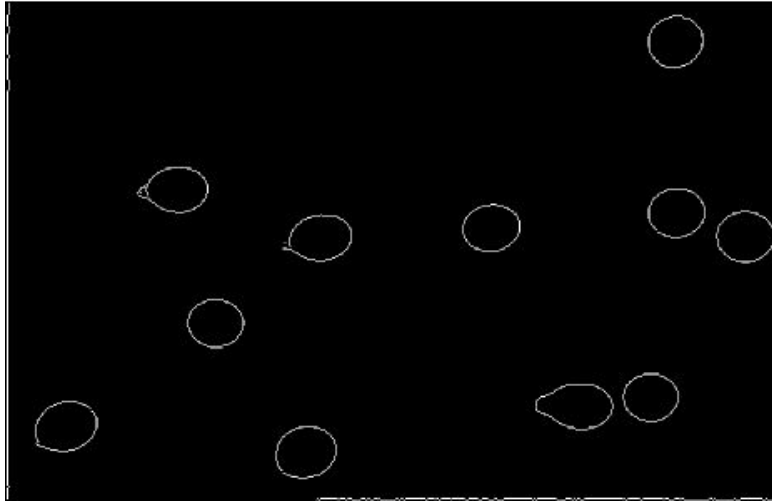


Fig. 4.11. Bordes encontrados usando el método de Sobel con sensibilidad.

### 4.2.3. Etiquetado

El etiquetado permite identificar aquellas regiones conectadas en la imagen que pueden ser huellas alfa, usando el concepto de vecindad entre píxeles, se usa cuando se requiere un análisis exhaustivo de la imagen. Este procedimiento toma una imagen binarizada y retorna una matriz L, del mismo tamaño que la imagen binarizada de ingreso, conteniendo las etiquetas de los objetos conectados, por algún de los 2 tipos de vecindad, encontrados en la imagen binarizada de ingreso. Los elementos de la matriz L son números enteros mayores o iguales a cero donde los píxeles etiquetados con 0 corresponde al fondo, los píxeles etiquetados con 1 corresponden al primer objeto, los píxeles etiquetados con 2 corresponden al segundo objeto y así sucesivamente.

Una región puede tener sus píxeles conectados si estos se encuentran adyacentes o vecinos. La vecindad se puede definir de dos formas:

1. Vecindad 4 ( $V_4$ ). Cuando los píxeles adyacentes tienen un lado en común (ver la figura 4.12 (a)).
2. Vecindad 8 ( $V_8$ ). Cuando los píxeles adyacentes tiene un lado en común o un vértice en común (ver la figura 4.12 (b)).



Figura 4.12. Tipos de vecindad: (a). Vecindad  $V_4$ , (b). Vecindad  $V_8$ .

La imagen binarizada IBF mostrada en la figura 4.13 (a), utilizada como imagen de entrada para el procedimiento de etiquetado se obtuvo de la siguiente forma: sobre la imagen con los bordes de los objetos, determinados en la sección 4.2.2 (figura 4.11), aplicamos una dilatación usando un elemento estructurante en forma de disco, con la finalidad de que los borde encontrados en esta imagen sean curvas de trayectoria cerrada.

Con las curvas cerradas obtenidas en el paso anterior procedemos a rellenar el interior de estas. Al dilatar la imagen, en el paso anterior, aumentamos el tamaño de los objetos en la imagen, para tratar de regresar estos a su tamaño original aplicamos la operación morfológica erosión con el mismo elemento estructurante con el que se aplicó la dilatación en el paso anterior finalmente sumamos a la imagen anterior la imagen binarizada obtenida en la sección 4.2.1 (figura 4.10).

En la figura 4.13 (b) se muestra el resultado del proceso de etiquetado donde cada color representa un objeto (huella) encontrada, en este trabajo se consideró una vecindad  $V_4$  para identificar las regiones conectadas.

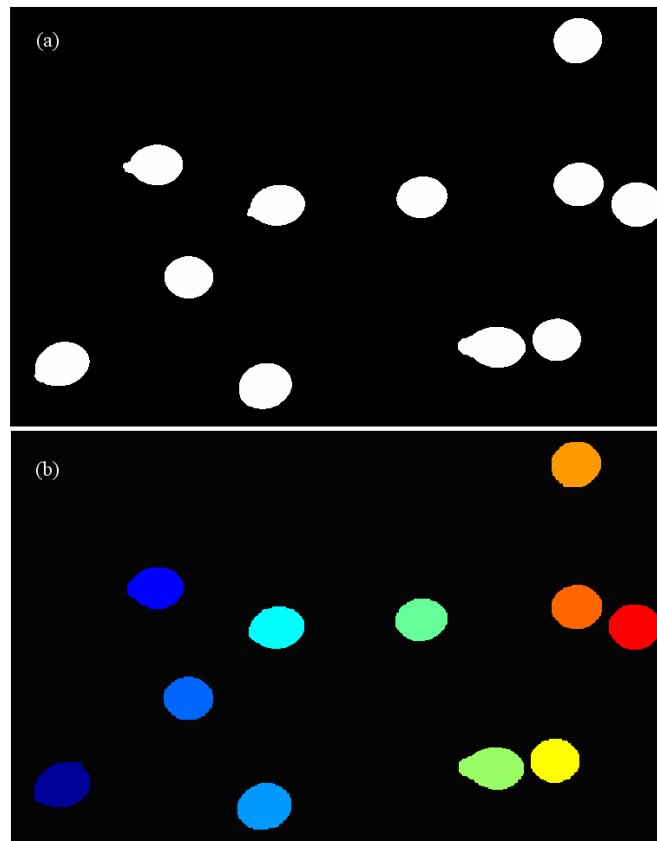


Fig. 4.13. (a) Imagen Binarizada IBF. (b) Imagen etiquetada.

#### 4.2.4. Extracción automática de huellas

En este procedimiento se extraen pequeñas ventanas (subimágenes) conteniendo los objetos etiquetados en el procedimiento anterior. Las ventanas se extraen desde la imagen original  $I$  y desde la imagen binarizada IBF. Las ventanas extraídas serán usadas en las siguientes etapas para extraer características desde cada una de ellas.

En la figura 4.14 se muestra el procedimiento seguido para la extracción automática de las huellas y su contorno, este procedimiento toma como entrada la imagen etiquetada, determinada en el procedimiento anterior, a partir de esta imagen se determina el 'BoundingBox' (Menor rectángulo que contiene completamente al objeto de interés y cuyo lados son paralelos a los ejes), la redondez, el área, la opacidad (nivel de gris medio) y la elongación para cada huella de la imagen etiquetada. Luego tomando las coordenadas de

la esquina izquierda (f,c) y de las dimensiones (largo, ancho) del 'BoundingBox' y extraemos las ventanas de la imagen original y de la imagen binarizada IBF, restando 10 píxeles al número de filas y columnas y agregando 15 píxeles al largo y ancho del 'BoundingBox', todo esto con la finalidad de extraer junto con la huella parte del entorno. Finalmente tomamos solo aquellas huellas que cumplen con ciertos criterios de restricción como son: la redondez debe de ser mayor que 0.85, el nivel de gris medio u opacidad menor que 0.45 y la elongación mayor a 0.7.

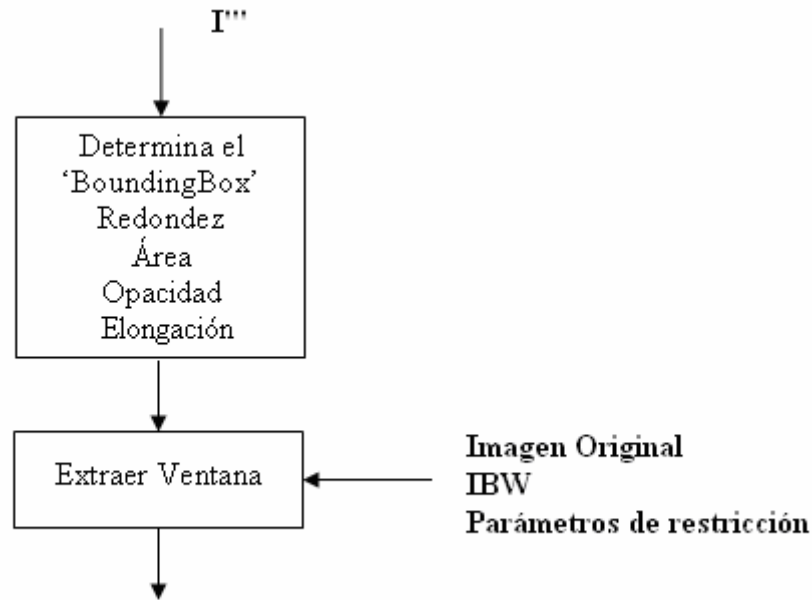


Fig. 4.14. Procedimiento seguido para la extracción de huellas.

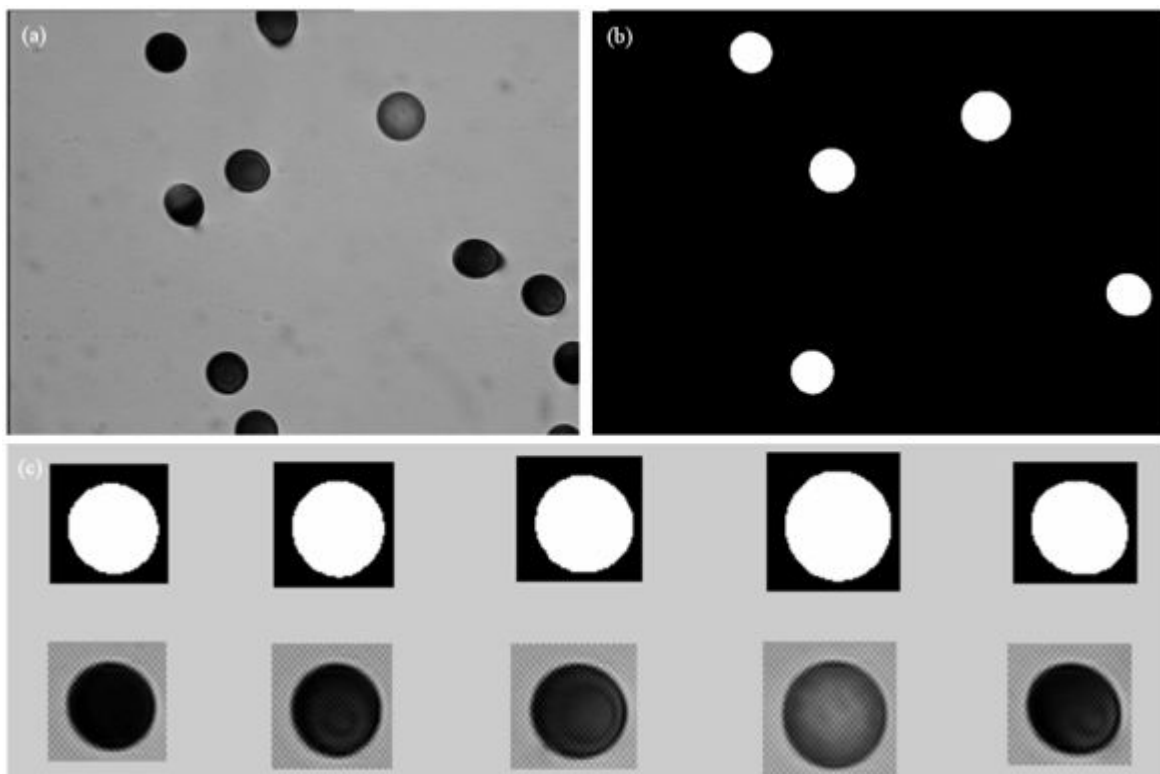


Fig. 4.15. (a) Imagen Original. (b) Imagen Binarizada IBF. (c) Huellas y entorno extraídas automáticamente de las imágenes (a) y (b).

En la figura 4.15 se muestra el resultado de la extracción automática de huellas y su entorno.

### 4.3. Extracción de Características

La extracción de características se realiza como paso previo para la tarea de clasificación. La idea subyacente de la extracción de características es la de caracterizar una huella para ser reconocida por medidas cuyos valores son muy similares para huellas en la misma categoría y diferentes para huellas en diferentes categorías. Esto nos lleva a la tarea de encontrar características fácilmente reconocibles que sean invariantes a transformaciones irrelevantes de la entrada.

Hasta este punto se ha descrito como detectar una huella nuclear alfa a través del procesamiento de imágenes en una imagen bidimensional; una vez logrado esto, es importante seleccionar características a través de las cuales podamos distinguir la categoría o clase a la que pertenece un elemento. En este apartado se discute las características extraídas para este propósito.

Se hizo una compilación de características frecuentemente utilizadas en el reconocimiento de patrones en imágenes binarias e imágenes en escala de grises las cuales las podemos clasificar en 4 grupos:

1. Características de forma  
Características que describen la forma de la huella.
2. Contraste de la huella  
Características relacionadas con el nivel de gris de los píxeles que conforman una huella.
3. Contraste del fondo  
Características relacionadas con el nivel de gris de los píxeles que conforman el fondo de la ventana que contiene la huella.
4. Contraste relativo  
Características que relaciona el promedio de gris de la huella con la del fondo.

Las características de contraste extraen atributos desde el nivel de gris, las características de forma describen las propiedades estructurales de las huellas. Las características extraídas pasarán luego por el proceso de selección de características donde se determinará el subconjunto que tiene mayor significancia en el proceso de clasificación.

Para facilitar el desarrollo de las fórmulas en este capítulo se usaran las variables:

1. I. Imagen en escala de grises de la huella y su entorno.
2. B. Imagen binaria de la huella y su entorno.
3. I'. Imagen en escala de grises de la ventana que contiene a la huella, con ceros en los píxeles que no corresponde a la huella, se calcula de la siguiente manera:

$$I'(f, c) = I(f, c) * B(f, c) \text{ para } f = 1 \dots N_f \text{ y } c = 1 \dots N_c \quad (4.3)$$

Donde  $N_f$  y  $N_c$  son el número de filas y de columnas respectivamente de la ventana que contiene a la huella.

4.  $F'$ . Imagen en escala de grises de la ventana que contiene a la huella, con ceros en los píxeles que corresponden a la huella, se calcula de la siguiente manera:

$$F'(f, c) = I(f, c) * (255 - B(f, c)) \text{ Para } f = 1 \dots N_f \text{ y } c = 1 \dots N_c \quad (4.4)$$

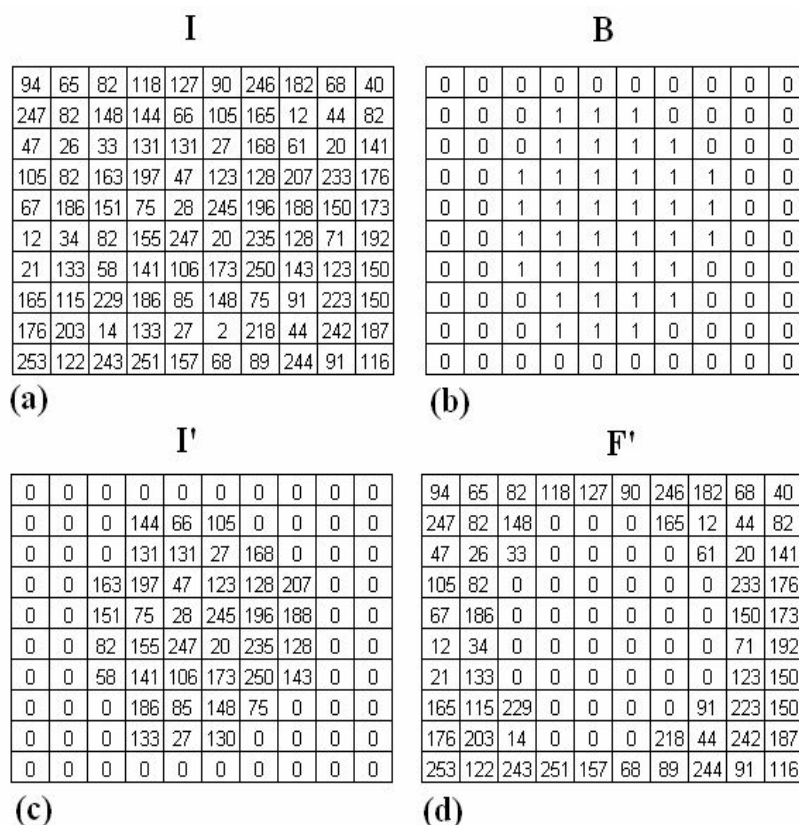


Fig. 4.16. (a) Imagen en escala de grises, b) Imagen binaria, c) Imagen en escala de grises, con ceros en los píxeles del fondo y d) Imagen en escala de grises con ceros en los píxeles de la huella.

### 4.3.1. Características de forma

Estas características intentan describir las propiedades geométricas y estructurales de una huella. En todos los casos los cálculos de estas características se realizan sobre la imagen binarizada B. Un punto importante a recalcar es que el *Píxel Aspect Ratio* (relación que hay entre el tamaño vertical y el tamaño horizontal del píxel) de las imágenes analiza es diferente de uno la sección, de la sección 3.1.4 se tiene:

$$\Delta x = 0.2123 \mu\text{m} \quad (4.5)$$

$$\Delta y = 0.2466 \mu\text{m} \quad (4.6)$$

Dando un *Píxel Aspect Ratio* de 1.16.

### 1. Área.

Viene a ser el número de píxeles que ocupa la huella en la ventana multiplicado por el área de cada píxel.

$$\text{Area} = \sum_{f=1}^{f=N_f} \sum_{c=1}^{c=N_c} B(f, c) * \text{AreaPixel} \quad (4.7)$$

Donde  $N_f$  y  $N_c$  corresponden al tamaño del lado de la ventana que contiene a la huella,  $B$  es la ventana y  $B(f, c)$  es un píxel de la ventana para todo  $B(f, c) \neq 0$  y

$$\text{AreaPixel} = \Delta x * \Delta y \quad (4.8)$$

### 2. Eje Mayor.

Es el diámetro máximo en la huella, esto es la distancia entre los extremos más lejanos.

Este valor es encontrado calculando la distancia entre todos los puntos del perímetro y buscando el par de puntos  $(x_1, y_1)$  y  $(x_2, y_2)$  con la mayor distancia. La distancia está dada por la relación.

$$\text{Dist} = \sqrt{(x_2 - x_1)^2 \Delta x^2 + (y_2 - y_1)^2 \Delta y^2} \quad (4.9)$$

### 3. Eje Menor.

Es el diámetro mínimo en la huella. Este valor es calculado buscando la menor distancia que existe entre 2 pares de puntos del perímetro de la huella que formen una línea perpendicular con el Eje mayor.

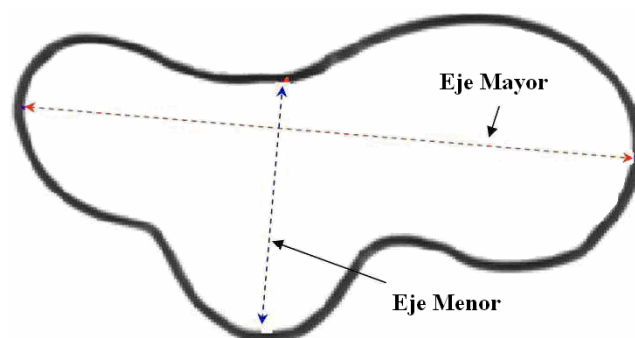


Fig. 4.17. Eje mayor y eje menor.

Las rectas que contienen a los ejes mayor y menor son conocidas como los ejes principales [70] de la imagen.

### 4. Elongación

Se calcula como el cociente entre el eje menor y el eje mayor. Este valor puede estar



entre 0 y 1. Si la elongación es 1, la huella tendrá la forma de un círculo. Este valor decrece a medida que el objeto se alarga (ver la figura 4.17).

$$Elongacion = \frac{Eje\ Menor}{Eje\ Mayor} \quad (4.10)$$

#### 5. Perímetro.

Es la longitud del perímetro de la *huella*. En este trabajo el perímetro se obtiene removiendo todos los píxeles del centro de la *huella*, y luego contando el número de transiciones entre dos píxeles sucesivos del perímetro, en las direcciones de los ejes X e Y y sobre la diagonal XY.

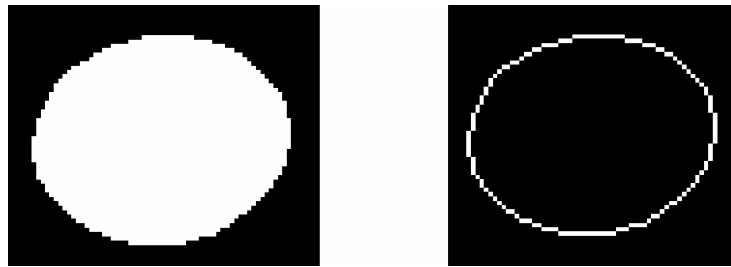


Fig. 4.18. Perímetro de una huella.

#### 6. Diámetro equivalente.

Este indicador es el equivalente al diámetro de un círculo con la misma área a la que ocupa la *huella*, se calcula con la siguiente fórmula:

$$Deq = \sqrt{\frac{4}{\pi} Area} \quad (4.11)$$

#### 7. Redondez.

Este es un indicador en cuanto a la forma de la huella. Este indicador para un círculo es 1, para un cuadrado es 0.78 y para una huella muy delgada es cercano a 0.0.

$$Redondez = \frac{4\pi Area}{Perimetro^2} \quad (4.12)$$

### 4.3.2. Contraste de la huella

Estas características se extraen sólo de los píxeles que conforman la huella sin considerar los píxeles del fondo.

#### 1. Nivel gris máximo.

Mayor nivel de gris de la huella.

$$Gris\ máximo = \max(I') \quad (4.13)$$

2. Nivel gris mínimo.

Menor nivel de gris de la huella.

$$Gris\ mínimo = \min(I') \quad (4.14)$$

3. Mediana del nivel gris.

Mediana de los niveles de gris de la huella.

$$Gris\ mediana = mediana(I') \quad (4.15)$$

4. Nivel promedio de gris.

Es el promedio de los niveles de gris en la huella, es un buen indicador de su nivel de brillo, se usará más adelante para ser comparado con el promedio de gris del fondo.

$$\bar{x} = \frac{1}{N} \sum_{f=1}^{f=N_f} \sum_{c=1}^{c=N_c} I'(f, c) \quad (4.16)$$

Donde  $I'(f, c) \in a la huella$  y  $N$  es el número de píxeles de la huella.

5. Desviación estándar del nivel de gris.

La desviación estándar de los niveles de gris de la huella, es un buen indicador del nivel de ruido. Valores bajos en la desviación estándar indicarán una superficie suave.

$$\sigma^2 = \frac{1}{N} \sum_{f=1}^{f=N_f} \sum_{c=1}^{c=N_c} (I'(f, c) - \bar{x})^2 \quad (4.17)$$

Donde  $I'(f, c) \in a la huella$  y  $N$  es el número de píxeles de la huella.

6. Asimetría del nivel de gris (Skewness).

Este estadístico es una medida del nivel de asimetría que presenta el histograma (ver la figura 4.19). Indica si la cola más larga del histograma se encuentra desviada hacia la derecha, centrada o desviada hacia la izquierda. Si está a la izquierda, el coeficiente es negativo y si está desviada a la derecha es positivo, la fórmula analítica es la siguiente:

$$Sk = \frac{1}{N\sigma^3} \sum_{f=1}^{f=N_f} \sum_{c=1}^{c=N_c} (I'(f, c) - \bar{x})^3 \quad (4.18)$$

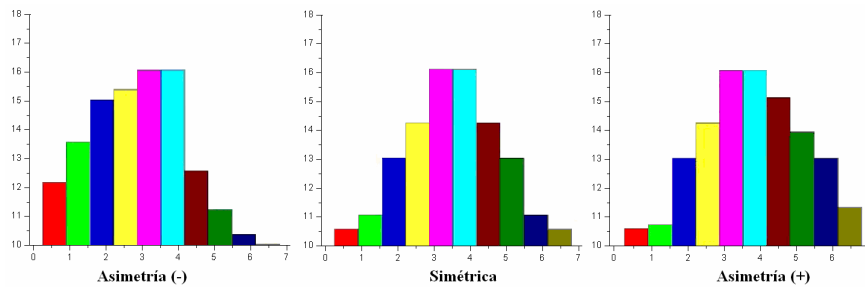


Fig. 4.19. Asimetría de una distribución de frecuencias.

### 7. Kurtosis del nivel de gris.

Este estadístico indica si las colas del histograma tienen una altura superior, igual o inferior a la de una distribución normal. Si el coeficiente es negativo se le llamará platicúrtica y los extremos estarán por debajo de la curva normal. Si el coeficiente es igual a cero, se le llamará mesocúrtica. Si el coeficiente es mayor que cero se le llamará leptocúrtica y los extremos estarán por encima de la curva normal, ver figura 4.20.

$$k = \frac{1}{N\sigma^4} \sum_{f=1}^{f=N_f} \sum_{c=1}^{c=N_c} (I'(f, c) - \bar{x})^4 - 3 \quad (4.19)$$

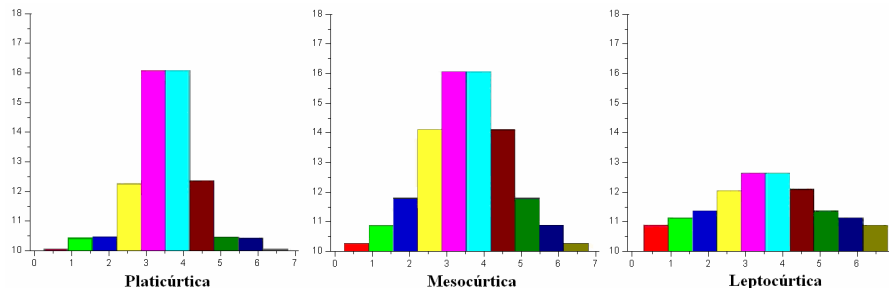


Fig. 4.20. Kurtosis del nivel de gris

### 4.3.3. Contraste del fondo

Estas características se extraen sólo de los píxeles que conforman el fondo de la ventana que contiene la huella. Corresponde a los indicadores del grupo anterior.

1. Nivel gris máximo del fondo

Mayor nivel de gris del fondo.

$$\text{Gris máximo} = \max(F') \quad (4.20)$$

2. Nivel gris mínimo del fondo.

Menor nivel de gris del fondo.

$$\text{Gris m\u00ednimo} = \min(F') \quad (4.21)$$

3. Mediana del nivel gris.

Mediana de los niveles de gris en del fondo.

$$\text{Gris mediana} = \text{mediana}(F') \quad (4.22)$$

4. Nivel promedio de gris.

Promedio de los niveles de gris del fondo

$$\bar{Y} = \frac{1}{M} \sum_{f=1}^{f=N_f} \sum_{c=1}^{c=N_c} F'(f, c) \quad (4.23)$$

Donde  $F'(f, c) \in$  al fondo  $M$  es el n\u00famero de p\u00edxeles del fondo.

5. Desviaci\u00f3n est\u00e1ndar del nivel de gris.

Desviaci\u00f3n est\u00e1ndar de los niveles de gris del fondo

$$\sigma^2 = \frac{1}{M} \sum_{f=1}^{f=N_f} \sum_{c=1}^{c=N_c} (F'(f, c) - \bar{Y})^2 \quad (4.24)$$

#### 4.3.4. Contraste relativo

Estas caracter\u00edsticas intentan encontrar relaciones entre el promedio de gris de la huella con el promedio de gris del fondo. Miden el contraste.

1. Contraste absoluto.

Viene a ser el promedio de gris en la huella menos el promedio de gris del fondo

$$\text{Contraste absoluto} = \text{Gris promedio huella} - \text{Gris promedio fondo} \quad (4.25)$$

2. Contraste Relativo.

El contraste relativo se calcula de la siguiente forma:

$$\text{Contraste relativo} = \frac{\text{Gris promedio huella} - \text{Gris promedio fondo}}{\text{Gris promedio huella} + \text{Gris promedio fondo}} \quad (4.26)$$

3. Contraste proporcional.

Se calcula de la siguiente forma:

$$\text{Contraste proporcional} = \frac{\text{Gris promedio huella}}{\text{Gris promedio fondo}} \quad (4.27)$$

#### **4.4. Conteo de Huellas Nucleares alfa**

Una vez que se ha aplicado el procesamiento de imágenes para detectar huellas en un conjunto de imágenes, el conteo de las huellas no es más que la suma de todas las huellas detectadas en la secuencia de imágenes analizada. Es fácil darse cuenta que un proceso de detección eficiente reportará un conteo de huellas aceptable. Así mismo, una vez que se ha alcanzado el final de la secuencia de imágenes se procede a almacenar las características extraídas en una matriz de características, el cálculo del número de huellas alfa es el tamaño de esta matriz, de tal manera que el conteo de huellas depende directamente de la eficiencia del algoritmo de detección.

#### **4.5. Clasificación de las huellas nucleares alfa a través de un clasificador basado en RNAs**

En esta sección se discute los pasos que se seguirán para obtener la clasificación de las huellas nucleares alfa usando un clasificador basado en redes neuronales. Se describirán los bloques del diagrama de flujo de la figura 4.21 para describir el proceso de clasificación de las huellas a través de RNAs.

##### **4.5.1. Definir la RNA a utilizar**

Esta es la parte medular del proceso de clasificación. En este paso se crea la red neuronal artificial que aprenderá a clasificar las huellas a partir de las características y respuestas deseadas. En este punto se define el número de neuronas en la capa de entrada, número de neuronas en la capa oculta, número de neuronas en la capa de salida, función de transferencia en la capa oculta, función de transferencia en la capa de salida, medida de desempeño de la red; en otras palabras la arquitectura de la RNA.

##### **4.5.2. Definir patrones de entrenamiento**

Es necesario definir patrones para el entrenamiento de la RNA ya que para llevar a cabo la clasificación de las huellas nucleares alfa se debe entrenar la red con el comportamiento deseado.

Para llevar a cabo el entrenamiento de la RNA y que ésta presente un buen desempeño, es necesario separar el conjunto principal de entrada en tres subconjuntos: de entrenamiento, de validación y de prueba. Así, es necesario dividir la matriz de características arrojada por el proceso de detección de huellas en tres subconjuntos: huellas que pertenecen al subconjunto de entrenamiento, los que forman parte del subconjunto de validación y los que constituyen el subconjunto de prueba.

Una vez preparada la RNA, ésta está lista para su entrenamiento. En este proceso se

van presentando las diferentes entradas y respuestas deseadas, preparadas previamente, al clasificador para que a través de la retropropagación del error encuentre un conjunto de pesos y sesgos que minimice la medida de desempeño del clasificador neuronal.

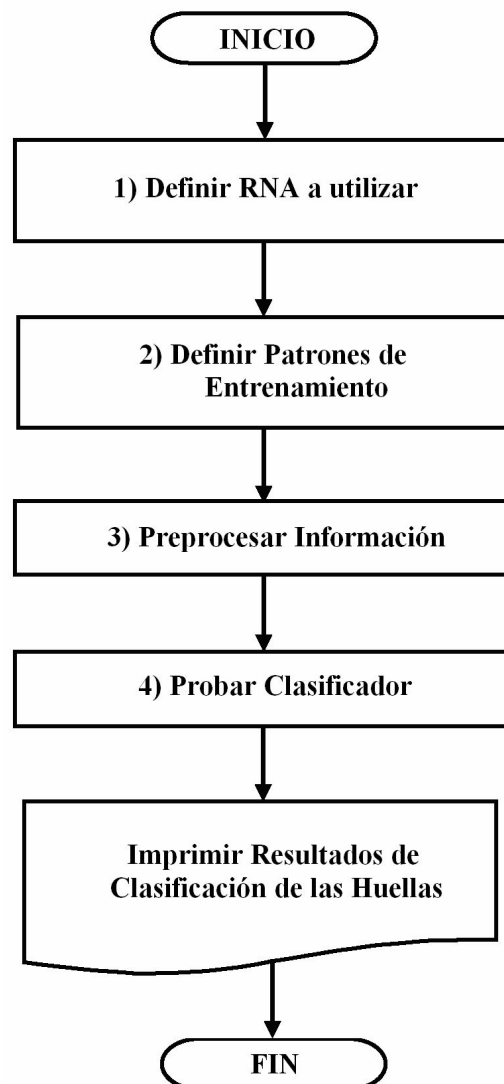


Fig. 4.20. Diagrama de flujo del procedimiento de clasificación de Huellas

### 4.5.3. Preprocesar la información

En la práctica casi siempre es ventajoso aplicar transformaciones de preprocesamiento a las entradas antes de que sean presentadas a la red. Para aplicaciones prácticas, el preprocesamiento es uno de las etapas más importantes en el desarrollo de la solución, y las opciones de preprocesamiento pueden tener un efecto significativo en el desempeño de generalización.

Además, se sabe que el preprocesamiento reduce el grado de dispersión y complejidad de los datos en el espacio de búsqueda bajo el cual se lleva a cabo el entrenamiento, proceso en el cual la RNA trata de aprender los datos que se le presentan, siendo por tanto un factor importante si se busca mejorar su desempeño. En el capítulo 6 se presenta en detalle el proceso de preprocesamiento aplicado tanto a las entradas (matriz

de características) como a las respuestas deseadas de la red.

#### **4.5.4. Probar el clasificador.**

La red una vez entrenada puede ser probada con el subconjunto de prueba que se seleccionó. El conjunto de prueba no interviene en el proceso de entrenamiento y provee así una idea general del comportamiento de la red ante información nueva y desconocida.

## 5. Diseño del software

Este capítulo se muestra el diseño final del software desarrollado para el conteo y clasificación de huellas nucleares alfa, además se describen las herramientas de desarrollo empleadas.

### 5.1. Descripción general del Software

*Alfa Track Análisis*, nombre dado a la aplicación desarrollada, es un entorno gráfico, GUI, (Graphical User Interface) desarrollado en Java [76] usando las librerías de componentes gráficos Swing [82] y SwingX [83], consta de aproximadamente 6000 líneas de código, e integra los algoritmos desarrollados en Matlab [72], gracias al toolbox Matlab builder for java [75], en una sola aplicación, proporcionando una manera cómoda e intuitiva de manejar la aplicación de algoritmos de procesamiento de imágenes desarrollados y descritos en el capítulo anterior y el modelo de red neuronal que se describirá en el siguiente capítulo. En definitiva, se trata de un interfaz gráfico que nos permite seleccionar las imágenes a procesar mediante el clásico “browser” de Windows, y pasar al tratamiento de estas sin necesidad de ejecutar la aplicación en modo consola. El uso de este interfaz gráfico también hace posible automatizar el proceso de manera cómoda. De esta manera, se obtienen los resultados directamente de una forma más visible para el usuario final del programa. La aplicación se ejecuta bajo Windows al hacer doble clic sobre el archivo *AlfaTrackAnalysisFinal.jar*, que se entrega con la tesis en un CD.

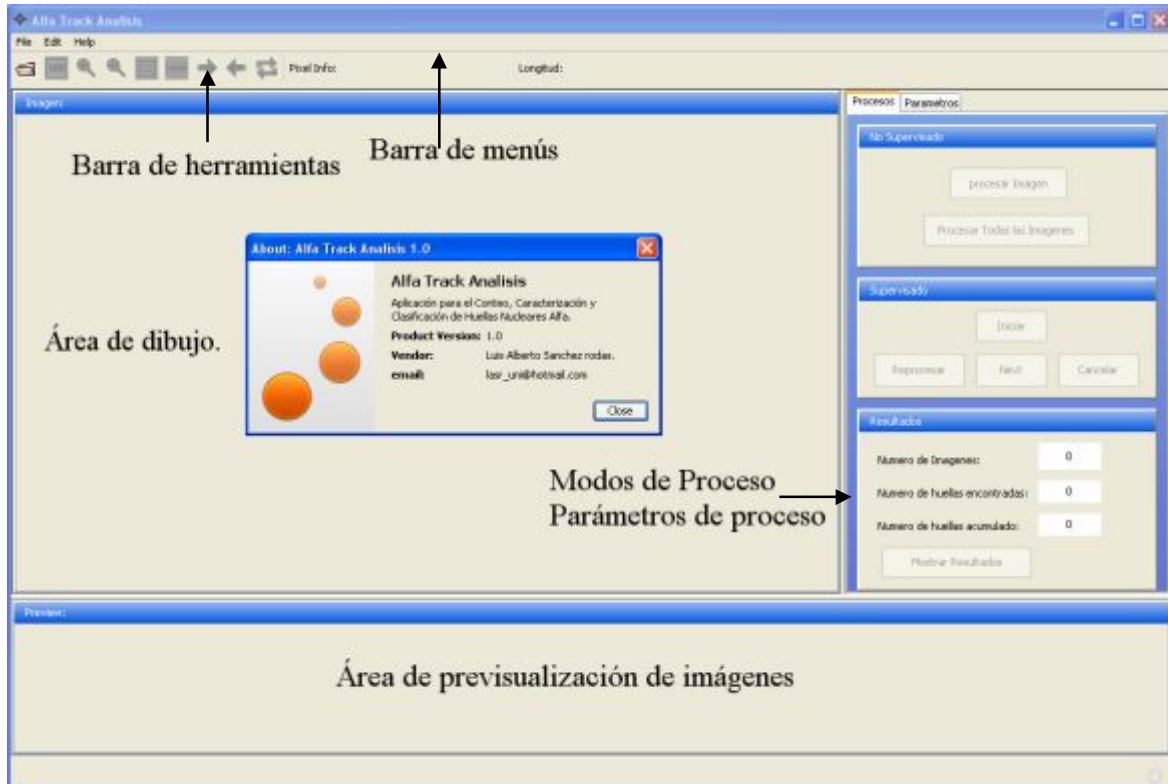


Fig. 5.1. Vista principal del software desarrollado.



En la figura 5.1 se puede apreciar la vista principal del software desarrollado, es en esta vista donde se sitúa el usuario a la hora de iniciar la ejecución del programa, como se puede apreciar en esta figura, la aplicación esta dotada con todos los elementos distintivos de una aplicación Windows: barra de menús, barra de herramientas, barra de estado, barra de progreso, etc. El área principal de la aplicación esta dividido en 3 regiones:

- Área de dibujo, es en esta zona donde se cargará la imagen a procesar y se mostrará algunos resultados del procesamiento de esta imagen.
- Modos de proceso y parámetros de proceso, ubicado en el lado izquierdo de la ventana principal del software, en 2 pestaña etiquetadas con los nombres procesos y parámetros.
- Área de previsualización de imágenes, ubicada la parte inferior de la ventana principal, muestra las imágenes cargadas en memoria pero en pequeñas dimensiones.

### 5.1.1. Barra de herramientas.










La barra de herramientas, mostrada en la figura 5.2, fue dotada con todas las funcionalidades básicas para el procesamiento de imágenes.



Fig. 5.2. Barra de herramientas

La tabla 5.1 muestra las funcionalidades de los distintos botones de la barra de herramientas

Tabla 5.1. Botones de la barra de herramientas

	Carga una o varias imágenes.
	Muestra la imagen en su tamaño original.
	Zoom In.
	Zoom Out.
	Permite navegar a través de una imagen cuando esta es de mayor tamaño que el panel contenedor.
	Permite medir distancias.
	Muestra la imagen siguiente.
	Muestra la imagen anterior
	Refresca una imagen.

Adicionalmente en la parte derecha de la barra de herramientas, ver figura 5.2, se

muestra 2 etiquetas, la primera de ellas muestra las coordenadas de la imagen por donde pasa el cursor y su correspondiente valor de intensidad de gris (Número entero entre 0 y 255). La segunda etiqueta muestra la longitud medida usando el botón para medir distancias.

### 5.1.2. Modos de operación

*Alfa Track Análisis* está diseñado para trabajar en 2 modos de operación:

- Modo 1: Conteo y caracterización, con este modo podemos contar y extraer automáticamente los parámetros de las huellas alfa detectadas.
- Modo 2: Clasificación, en este modo de operación al igual que el modo 1 podemos contar y extraer automáticamente los parámetros de las huellas, adicionalmente este modo permite hacer uso de la red neuronal implementada para clasificar las huellas nucleares alfa.

Para seleccionar el modo de trabajo hay que ir a la barra de menús, opción Edit y luego en Set Mode, en la figura 5.3 se muestra las 2 opciones para los modos de operación.



Fig. 5.3. Modos de operación del software

### 5.1.3. Tipos de proceso

Una de las mejores características con las que se ha dotado a este software es la posibilidad de hacer el procesamiento de las imágenes tanto de una manera supervisada como no supervisada.

En el modo No Supervisado el procesamiento de una o de varias imágenes se realiza sin la intervención del usuario, una vez iniciado el proceso el hilo de ejecución no termina hasta que la última de las imágenes es procesada.



Fig. 5.4. Procesamiento no supervisado

En la figura 5.4 se muestra las opciones para este tipo de proceso:

- Botón *Procesar Imagen*, procesa la imagen actual seleccionada.
- Botón *Procesar Todas las Imágenes*, procesa todas las imágenes cargadas en memoria.

La imagen actual seleccionada es la imagen encerrada en un rectángulo cuyo perímetro esta resaltado con color rojo en el área de previsualización de imágenes como se puede apreciar en la figura 5.8.

Por otro lado en el procesamiento supervisado, el hilo de ejecución se detiene después de que cada imagen es procesada permitiendo al usuario cambiar los parámetros de proceso hasta obtener el resultado deseado. En la figura 5.5 se muestran los botones que permiten controlar el procesamiento supervisado de las imágenes

- Botón *Iniciar*, inicia el procesamiento de las imágenes.
- Botón *Reprocesar*, reprocesa la imagen actual.
- Botón *Next*, procesa la siguiente imagen.
- Botón *Cancelar*, cancela el procesamiento de las imágenes.

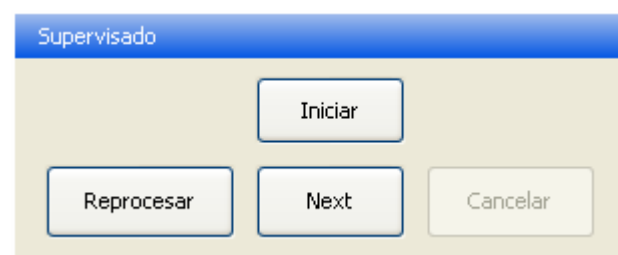


Fig. 5.5. Supervisado

La figura 5.7 muestra una instantánea de la ventana principal de software desarrollado durante el procesamiento de 16 imágenes en el modo supervisado. Para cada imagen procesada el centro de cada huella detectada es etiquetada con un número encerrado en un pequeño rectángulo de amarillo, al pasar el cursor sobre este rectángulo emerge una pequeña ventana con los valores de los principales parámetros de las huellas encontradas, de igual manera el perímetro de cada huella detectada es encerrada por una línea en rojo; que tan bien se ajusta esta línea al perímetro verdadero de la huella da al usuario una idea de que tan bien a resultado el algoritmo de procesamiento, pudiendo así modificar los parámetros de proceso con la finalidad de que estas líneas coincidan.

#### 5.1.4. Resultados.

Durante el procesamiento de las imágenes la ventana *Resultados*, ver figura 5.6, muestra algunos de los resultados del procesamiento de las imágenes

- Etiqueta *Número de Imágenes*, muestra el número de imágenes a procesar cargadas en memoria.
- Etiqueta *Número de huellas encontradas*, muestra el número de huellas encontradas en la imagen actual analizada.

- Etiqueta *Número de huellas acumulado*, muestra el número de huellas totales.
- Botón *Mostrar Resultados*, permite cargar la ventana *Alfa Track Analisis: Resultados*, mostrada en la figura 5.9, este botón permanece inactivo hasta que todas las imágenes cargadas en memoria son procesadas.

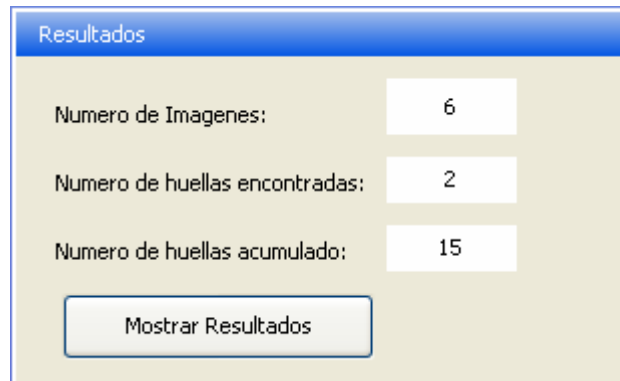


Fig. 5.6. Ventana resultados.

Los resultados obtenidos mostrados en la figura 5.9 pueden ser gravados directamente en un documento Excel para su posterior tratamiento así mismo el software permite generar histogramas de los resultados obtenidos.

### 5.1.5. Parámetros de preprocesamiento.

Todos los parámetros de ingreso necesarios para el preprocesamiento de las imágenes mencionadas en el capítulo anterior:

- Eliminar ruido presente en la imagen, sección 4.1.1.
- Mejorar del contraste de la imagen, sección 4.1.2.

Se pueden modificar en esta ventana, la figura 5.7 muestra los valores por defecto para estos parámetros.

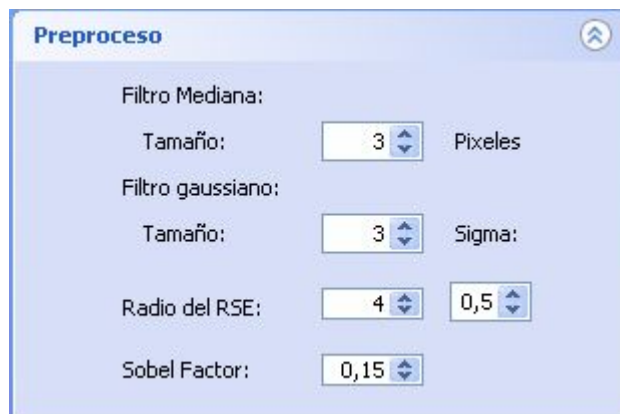


Fig. 5.7. Ventana parámetros de preproceso



Fig. 5.8. Vista principal del software procesando algunas imágenes.

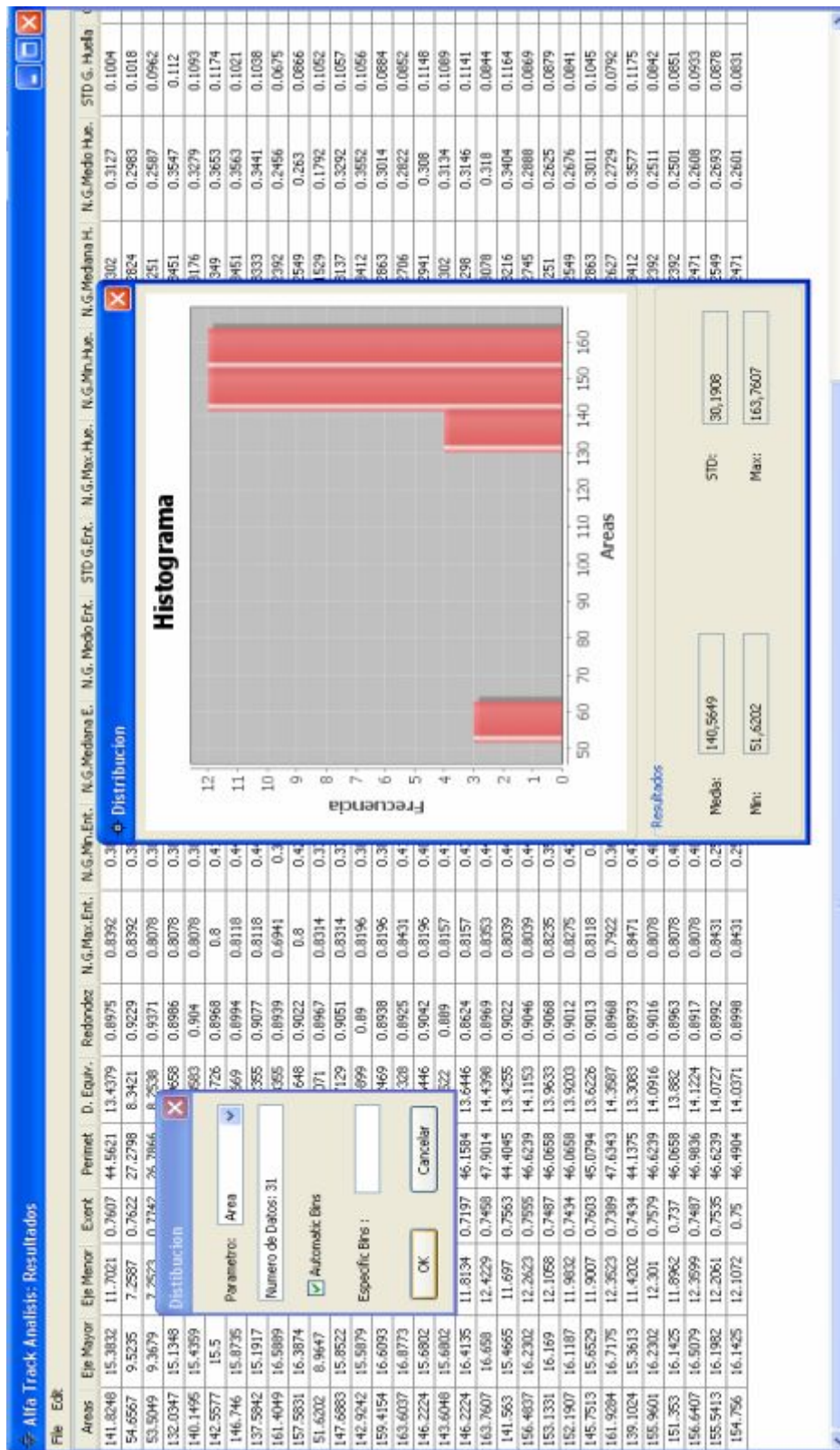


Fig.5.9. Alfa Track Analysis: Resultados

### 5.1.6. Opciones de segmentación

Todas las opciones para la segmentación de las imágenes introducidas en la sección 4.2 del capítulo anterior se pueden modificar desde esta ventana, mostrada en la figura 5.10.

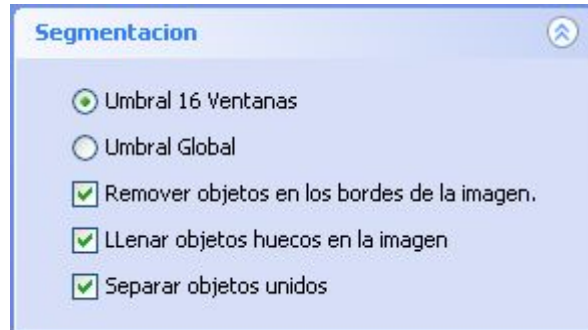


Fig. 5.10. Opciones para la segmentación de imágenes

### 5.1.7. Parámetros de restricción.

Esta ventana nos permite modificar los parámetros de restricción para las huellas alfa encontradas. Hay que mencionar que dependiendo del modo de operación del software los valores por defecto parámetros de restricción cambian, la figura 5.11 muestra los valores por defecto para los parámetros de restricción para el modo de clasificación del software.



Fig. 5.11. Parámetros de restricción, para el modo de clasificación del software desarrollado

El modelo GUI desarrollado hace posible que un usuario que no sea experto en programación usar los algoritmos fácilmente, ya que no necesita saber el funcionamiento del algoritmo, ni necesita saber qué tipo de entrada se espera ni el proceso que se produce. Sólo es necesario saber cómo trabaja el propio interfaz gráfico.

## 5.2. Herramientas de desarrollo

En esta sección se dará una breve descripción de las herramientas utilizadas en el desarrollo del software.

### 5.2.1. Matlab®

*MATLAB*® [72] (abreviatura de *MATrix LABoratory*, "laboratorio de matrices") es un software para cálculo técnico y científico que ofrece un entorno de desarrollo integrado (IDE) interactivo para el desarrollo de algoritmos, visualización de datos, análisis de datos y computación numérica. Cuenta un lenguaje de programación propio (lenguaje M) de alto nivel fácil de utilizar que aumenta significativamente la productividad de los programadores respecto a otros lenguajes de desarrollo tradicionales tales como C, C++, Fortran. Está disponible para las plataformas Windows, Unix y Apple Mac OS X.

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaz gráfica de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. Además, se pueden ampliar las capacidades de *MATLAB* mediante las cajas de herramientas (*toolboxes*).

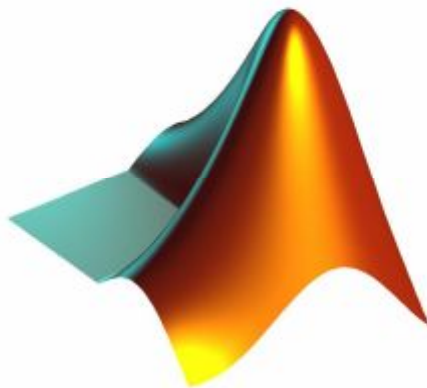


Fig. 5.12. Logo de Matlab®.

A continuación se dará una breve descripción de los *Toolboxes* utilizados en el desarrollo de nuestro software.

#### 5.2.1.1. Image Processing Toolbox

*Image Processing Toolbox* [73] proporciona a Matlab de un conjunto funciones y herramientas gráficas para el procesamiento, análisis y visualización de imágenes que amplía las capacidades del producto para realizar desarrollo de aplicaciones y nuevos algoritmos en el campo del proceso y análisis de imágenes. El entorno matemático y de creación de Matlab es ideal para el procesado de imágenes, ya que estas imágenes son, al fin y al cabo, matrices. Este *toolbox* incorpora funciones para: Realce de imágenes, imagen *deblurring*, detección de características, reducción del ruido, segmentación de imágenes, Transformaciones espaciales y registro de imágenes.

*Image Processing Toolbox* soporta una gran diversidad de tipos de imágenes, entre ellas tenemos: Imágenes de alto rango dinámico (*high dynamic range*), gigapixel resolution, ICC-compliant color e imágenes tomográficas, además las herramientas gráficas incluidas en este *toolbox* nos permite explorar una imagen, examinar una región de píxeles, adjuntar el contraste, crear perfiles o histogramas y manipular regiones de interés (ROIs).



### 5.2.1.2. Neural Network Toolbox

*Neural network toolbox* [60] proporciona funciones para el diseño, inicialización, simulación y entrenamiento de los modelos neuronales de uso más extendido en la actualidad: Perceptrón, redes lineales, redes de retropropagación, redes de base radial, aprendizaje asociativo y competitivo, aplicaciones autoorganizativas, aprendizaje de cuantización vectorial, redes de Elman y redes de Hopfield.

Mediante la inclusión de un amplio abanico de funciones y procedimientos escritos para Matlab, el usuario puede mediante el *Neural Network Toolbox* efectuar el diseño de arquitecturas complejas, combinando los modelos que ya están proporcionados por defecto en el *toolbox*. Asimismo, el usuario puede definir sus propias funciones de transferencia e inicialización, reglas de aprendizaje, funciones de entrenamiento y estimación de error para usarlas posteriormente con las funciones básicas.

El *toolbox*, aporta las facilidades y prestaciones gráficas de *MATLAB* para el estudio del comportamiento de las redes: visualización gráfica de la matriz de pesos y vector de desplazamiento mediante diagramas de Hinton, representación de errores a lo largo del entrenamiento, mapas de superficie de error en función de pesos y vector de desplazamiento, etc. Estos gráficos resultan muy útiles en el estudio de la convergencia y estabilidad de los algoritmos de aprendizaje.

### 5.2.1.3. Matlab Compiler.

*Matlab Compiler* [74] permite compartir aplicaciones Matlab como ejecutables o bibliotecas compartidas. Los ejecutables y bibliotecas desarrollados con Matlab utilizan un motor en tiempo de ejecución llamado *Matlab Component Runtime* (MCR) componente Matlab en tiempo de ejecución. El MCR se incluye en el compilador Matlab para distribuirse con su aplicación y puede instalarse libremente en cualquier equipo sin pagar licencias adicionales.

*Matlab Compiler* permite ejecutar aplicaciones *Matlab* fuera del entorno de *Matlab*. Esta arquitectura reduce significativamente el periodo de desarrollo ya que no resulta necesario traducir manualmente el código a un lenguaje diferente. Puede utilizarse el lenguaje *MATLAB* de alto nivel, optimizado para matrices y las funciones matemáticas, gráficas y de análisis de datos para crear prototipos, desarrollar y probar aplicaciones y funciones de manera rápida. Cuando haya completado las aplicaciones y funciones, se puede utilizar *Matlab Compiler* para empaquetarlas como ejecutables o bibliotecas compartidas que pueden llamarse desde C o C++, sin necesidad de recodificarlas en otro lenguaje. Si construye una aplicación autónoma, *Matlab Compiler* creará un ejecutable para los usuarios finales. Si necesita la integración con C o C++, *Matlab* ofrece una interfaz para utilizar su código como biblioteca compartida.

Algunas de las principales funciones de este *toolbox* se indican a continuación:

- Empaqueta las aplicaciones de *MATLAB* como ejecutables y bibliotecas compartidas.
- Permite distribuir libremente ejecutables autónomos y componentes de software.

- Permite integrar los algoritmos basados en *MATLAB* con aplicaciones desarrolladas con otros lenguajes y tecnologías.
- Encripta el código *MATLAB* para que no pueda visualizarse ni modificarse.

#### 5.2.1.4. Matlab Builder JA.

*Matlab Builder JA* [75] es una extensión de Matlab Compiler junto con el cual nos permite compartir aplicaciones *Matlab* en bibliotecas compartidas (archivos .Jar) que agrupan clases Java creadas a partir de funciones escritas en Matlab. Estas clases Java pueden ser integradas en programas Java y distribuidas libremente al usuario final o a un servicio web.

*Matlab Builder JA* encripta funciones Matlab y genera un envoltorio Java alrededor de estos de tal forma que se comporten de la misma manera como clases nativas Java. La clase Java creada con *Matlab Builder JA* son portables y corren en todas las plataformas soportadas por *Matlab*. Después de instanciar una clase Java basado en un componente *Matlab* creado usando *Matlab Builder JA*, podemos acceder a las funciones *Matlab* contenidas en ella de la misma forma a como se accede a los métodos nativos Java. Los tipos de datos nativos de Java pasados a las funciones *Matlab* son convertidos automáticamente a tipos de datos *Matlab*. Alternativamente podemos explícitamente seleccionar el tipo de dato a usar. En el anexo 9.2.2 se muestra el código java que permite hacer uso de las funciones .m desarrolladas en *matlab* y que están agrupadas en la librería *ProcesImage.jar* generada usando *Matlab Builder JA*.

### 5.2.2. Java

Java [76] surgió en 1991 cuando un grupo de ingenieros de *Sun Microsystems* [77] trataron de diseñar un nuevo lenguaje de programación destinado a electrodomésticos. La reducida potencia de cálculo y memoria de los electrodomésticos llevó a desarrollar un lenguaje sencillo capaz de generar código de tamaño muy reducido. Sun describe al lenguaje Java como “*simple, orientado a objetos, distribuido, interpretado, robusto, seguro, de arquitectura neutra, portable, de altas prestaciones, multitarea y dinámico*”.

Debido a la existencia de distintos tipos de CPUs y a los continuos cambios, era importante conseguir una herramienta independiente del tipo de CPU utilizado. Desarrollaron un código “neutro” que no dependía del tipo de electrodoméstico, el cual se ejecutaba sobre una “*máquina hipotética o virtual*” denominada Java Virtual Machine (JVM). Era la JVM quien interpretaba el código neutro convirtiéndolo a código particular de la CPU utilizada. Esto permitía lo que luego se ha convertido en el principal lema del lenguaje: “*Write Once, Run Everywhere*”.

Al programar en Java no se parte de cero. Cualquier aplicación que se desarrolle se apoya en un gran número de clases preexistentes. Algunas de ellas las ha podido hacer el propio usuario, otras pueden ser comerciales, pero siempre hay un número muy importante de clases que forman parte del propio lenguaje (el API o Application Programming Interface de Java). Java incorpora en el propio lenguaje muchos aspectos que en cualquier otro lenguaje son extensiones propiedad de empresas de software o fabricantes de ordenadores (threads, ejecución remota, componentes, seguridad, acceso a bases de datos, etc.). Por eso muchos expertos opinan que Java es el lenguaje ideal para aprender la

informática moderna, porque incorpora todos estos conceptos de un modo estándar, mucho más sencillo y claro que con las citadas extensiones de otros lenguajes. Esto es consecuencia de haber sido diseñado más recientemente y por un único equipo.

### 5.2.2.1. Entorno de desarrollo de Java

Existen distintos programas comerciales que permiten desarrollar código *Java*. La compañía Sun, creadora de Java, distribuye gratuitamente el *Java Development Kit* [78] (JDK). Se trata de un conjunto de programas y librerías que permiten desarrollar, compilar y ejecutar programas en *Java*. Existe también una versión reducida del *JDK*, denominada *JRE (Java Runtime Environment)* [78] destinada únicamente a ejecutar código *Java* (no permite compilar).

Los IDEs (*Integrated Development Environment*), tal y como su nombre indica, son entornos de desarrollo integrados. En un mismo programa es posible escribir el código Java, compilarlo y ejecutarlo sin tener que cambiar de aplicación. Algunos incluyen una herramienta para realizar Debug gráficamente, frente a la versión que incorpora el *JDK* basada en la utilización de una consola bastante difícil y pesada de utilizar. Estos entornos integrados permiten desarrollar las aplicaciones en forma mucho más rápida, incorporando en muchos casos librerías con componentes ya desarrollados, los cuales se incorporan al proyecto o programa. Como inconvenientes se pueden señalar algunos fallos de compatibilidad entre plataformas, y ficheros resultantes de mayor tamaño que los basados en clases estándar. Para el desarrollo de *Alfa Track Análisis* se utilizó el IDE Netbeans en su versión 6.9, se puede descargar libremente de [79].

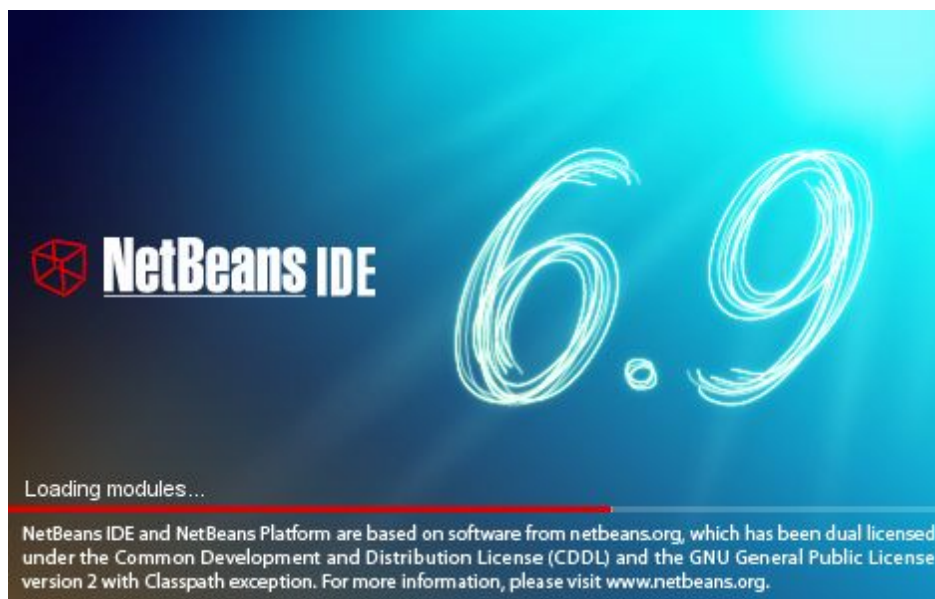


Fig. 5.13. IDE NetBeans usado para el desarrollo del software.

### 5.2.2.2. Librerías Java Utilizadas

Adicionalmente a las librerías estándar fue necesario importar otras librerías para dotar a nuestra aplicación con las funcionalidades deseadas:

#### 1. Java Advanced Imaging JAI

*Java Advanced Imaging* JAI [80] es una API que permite el manejo de funciones de entrada, salida para el procesamiento de imágenes. Se puede descargar libremente de [81].

## **2. JfreeChat**

*JFreeChart* [82] es una librería libre para gráficos escrita 100% en Java que facilita a los desarrolladores mostrar gráficos de calidad profesional en sus aplicaciones, ya sean web o de escritorio.

## **3. SwinX**

*SwinX* [84] nos ofrece una gran biblioteca de componentes gráficos personalizados los cuales los podemos utilizar para dar una mejor presentación a nuestras aplicaciones de escritorio.

## **4. Java Excel API**

*Java Excel* [85] es un API libre que permite a los desarrolladores leer, escribir y modificar hojas de cálculo Excel dinámicamente.

## 6. Resultados y discusiones

En este capítulo se muestra en la sección 6.1 los resultados de la aplicación de los algoritmos de procesamiento digital de imágenes descritos en el capítulo 4 de esta tesis e implementados en el módulo 1 del software desarrollado. En la sección 6.2 se muestra el diseño y evaluación de un modelo de red neuronal artificial para la clasificación de huellas nucleares alfa de acuerdo a la energía de las partículas alfa que las causan el cual es implementado en el módulo 2 del software desarrollado.

Se analizó el total de 608 imágenes que conforman la base de datos de imágenes características de huellas nucleares alfa de incidencia normal sobre el polímero detector CR-39. Las corridas se hicieron en una laptop con un procesador Intel Core Duo con una frecuencia de procesamiento de 2.0 GHz, 2.0 GB de memoria RAM y 160GB de disco duro. El sistema operativo utilizado fue el Microsoft Windows XP.

### 6.1. Análisis de Imágenes de Huellas nucleares alfa

A continuación presentamos el procedimiento general empleado en el análisis de las imágenes características de huella nucleares de partículas alfa con energías que van de 0.15 a 7.8 MeV. Con los montajes de irradiación se fija la energía de referencia y como para cada energía de referencia existen más de un grupo de energías las que inciden sobre el detector, es necesario identificar estos grupos de energías para ello haremos uso de los histogramas de los parámetros de las huellas características para cada energía.

Para cada detector analizado empleamos el siguiente análisis:

1. Construcción de los histogramas de los parámetros de las huellas alfa: área, eje menor, eje mayor.
2. Efectuamos un primer análisis cualitativo usando los histogramas para cada parámetro, identificando:
  - Grupo de datos (picos), valores centrales y valores de corte.
  - Coherencia con las imágenes características (número de grupos y dispersión de los datos).
  - Efectividad de algún parámetro para la separación.
  - Posibilidad de uso de uno de los parámetros para la separación de energías.
3. Efectuamos un análisis cuantitativo, utilizando los valores de corte sugerido en el ítem anterior, los valores medios de los picos procurando la correspondencia de los parámetros con la energía.
4. Con los datos resultantes de la separación hecha en el paso anterior, vamos agregando puntos a la curva variación de parámetros con la energía.

#### 6.1.1. Fuentes de Uranio

A continuación se muestran los histogramas construidos a partir del análisis de las imágenes características para las distintas energías de referencia. Para cada energía de

referencia se muestra 2 histogramas el primero del área de las huellas, el segundo la del eje menor o eje mayor, dependiendo de cual de estas es mas efectivo para la separación de energías.

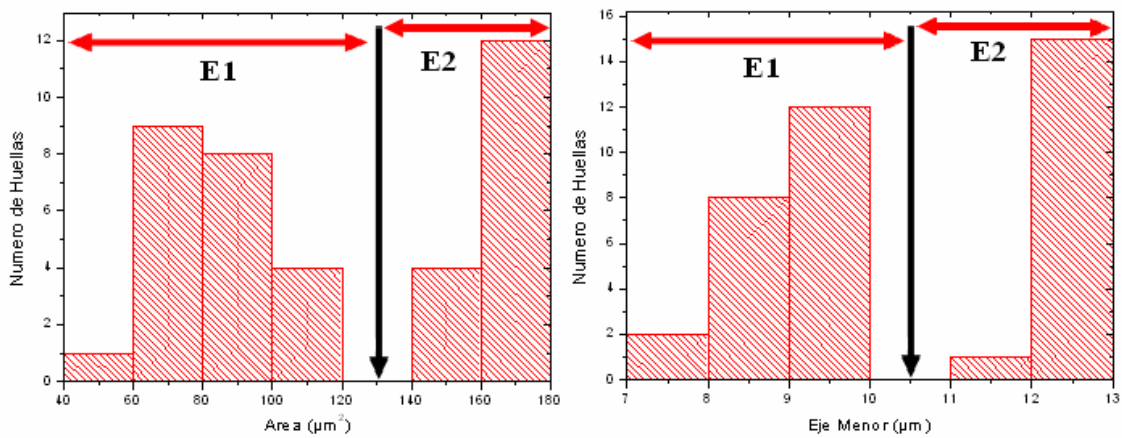


Fig. 6.1. Correlación de los grupos de energía en la configuración  $E_{ref} = 0.8$  MeV con  $E1 = 0.15$  MeV y  $E2 = 1.45$  MeV.

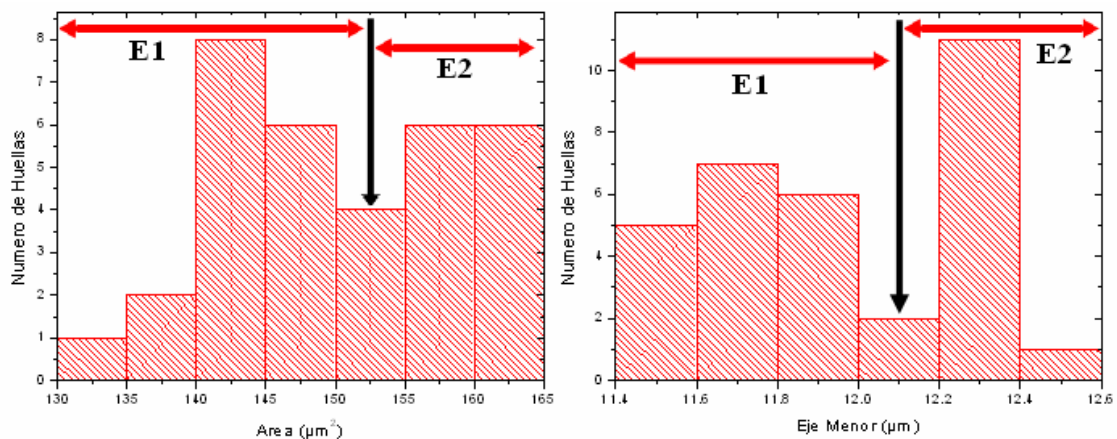


Fig. 6.2. Correlación de los grupos de energía en la configuración  $E_{ref} = 1.5$  MeV con  $E1 = 1.06$  MeV y  $E2 = 2.04$  MeV.

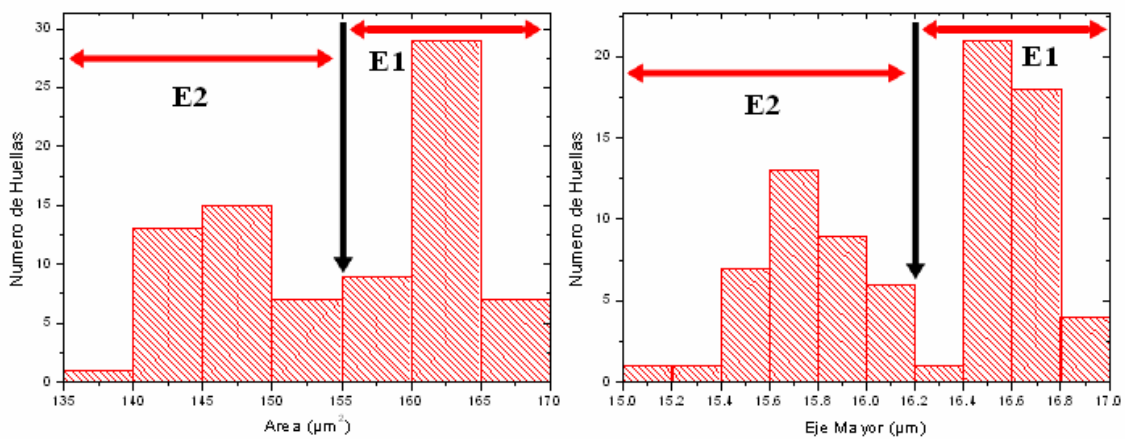


Fig. 6.3. Correlación de los grupos de energía en la configuración  $E_{ref} = 1.95$  MeV con  $E1 = 1.43$  MeV y  $E2 = 2.43$  MeV.

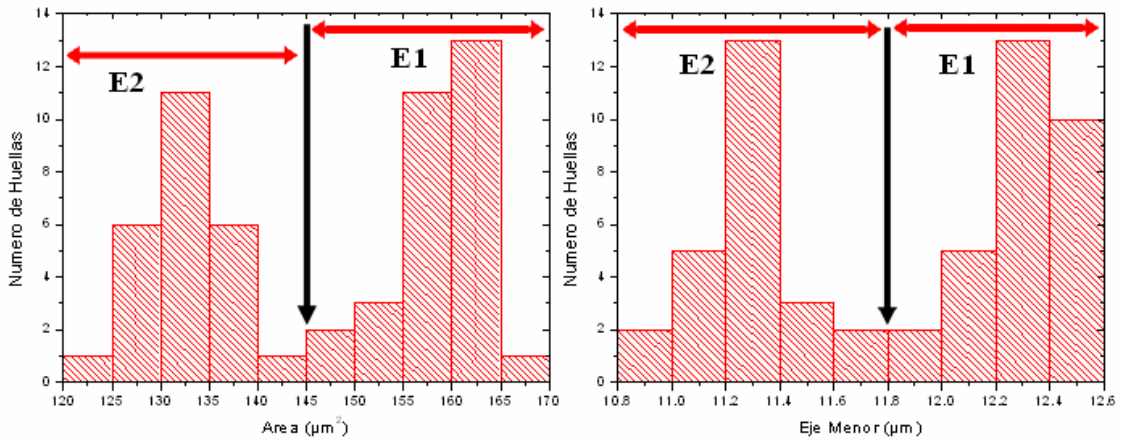


Fig. 6.4. Correlación de los grupos de energía en la configuración  $E_{ref} = 2.49$  MeV con  $E_1 = 2.04$  MeV y  $E_2 = 2.91$  MeV.

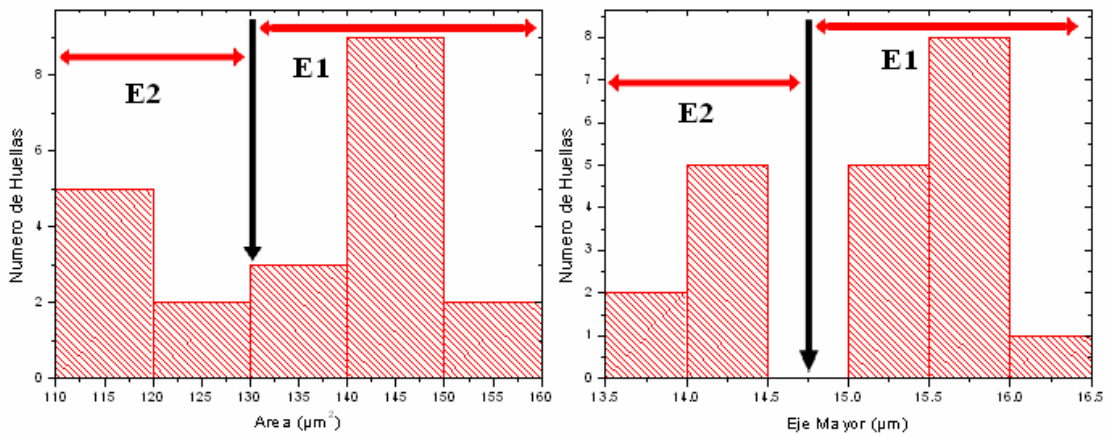


Fig. 6.5. Correlación de los grupos de energía en la configuración  $E_{ref} = 2.84$  MeV con  $E_1 = 2.43$  MeV y  $E_2 = 3.23$  MeV.

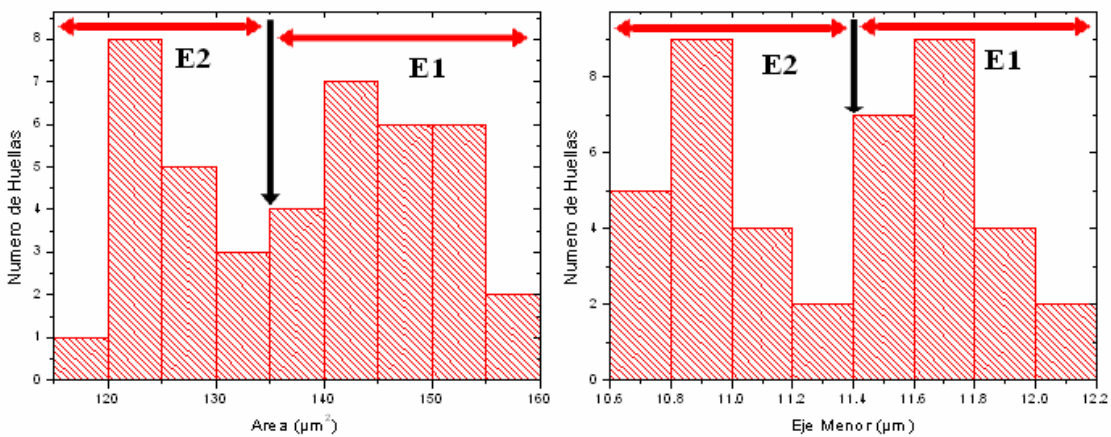


Fig. 6.6. Correlación de los grupos de energía en la configuración  $E_{ref} = 3.27$  MeV con  $E_1 = 2.90$  MeV y  $E_2 = 3.63$  MeV.

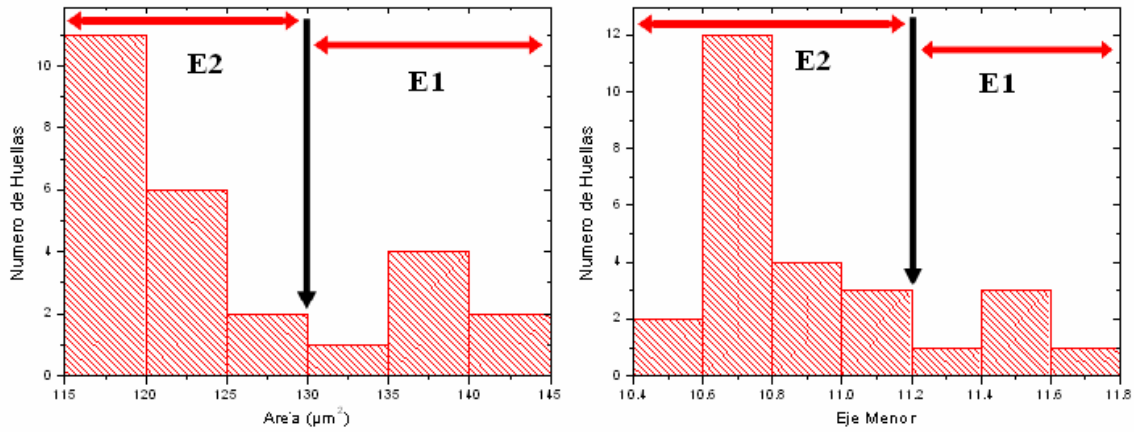


Fig. 6.7 Correlación de los grupos de energía en la configuración  $E_{ref} = 3.57$  MeV con  $E1 = 3.22$  MeV y  $E2 = 3.91$  MeV.

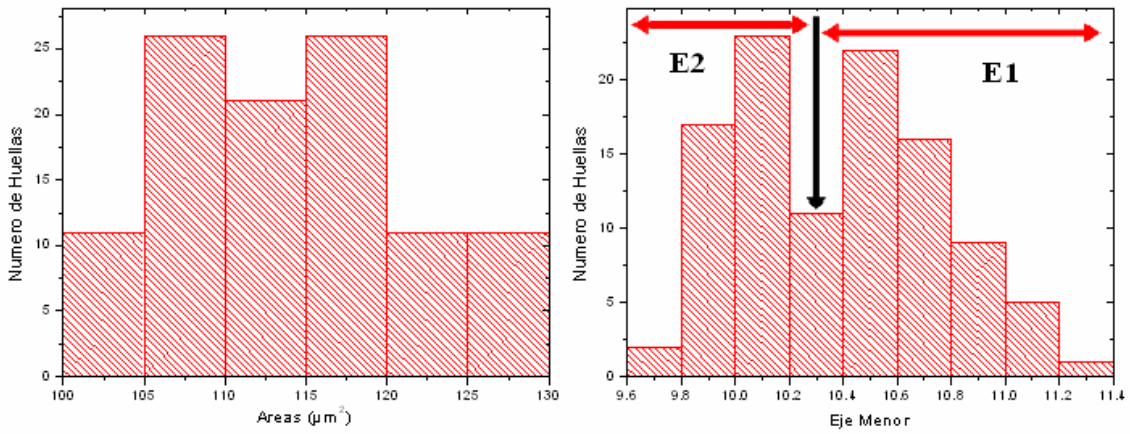


Fig. 6.8. Correlación de los grupos de energía en la configuración  $E_{ref} = 3.95$  MeV con  $E1 = 3.62$  MeV y  $E2 = 4.28$  MeV.

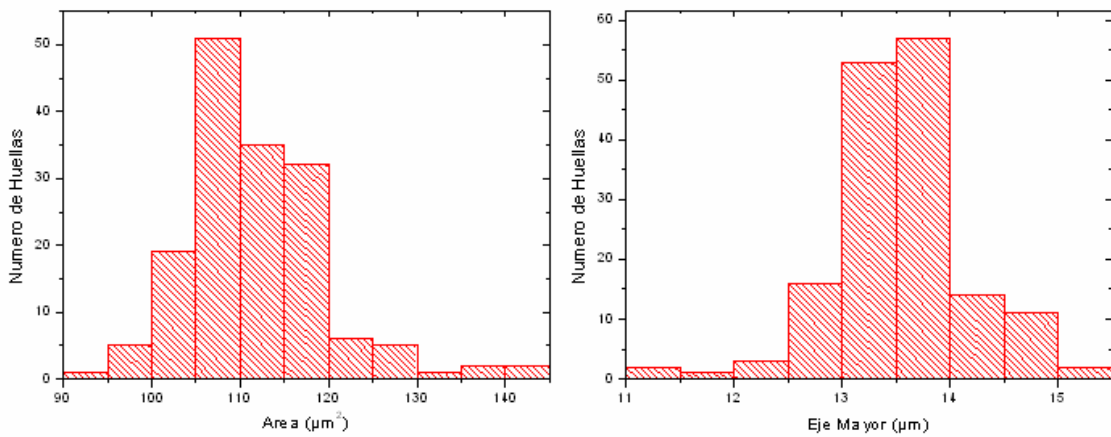


Fig. 6.9.  $E_{ref} = 4.22$  MeV



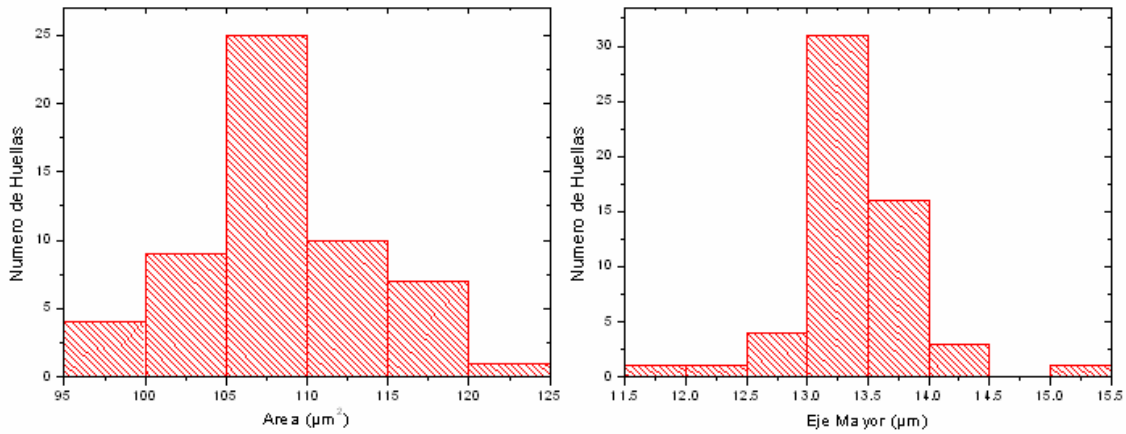


Fig. 6.10. Eref = 4.39 MeV.

A partir de los histogramas anteriores, usando los puntos de corte, indicados con la fecha vertical en cada histograma, realizamos una cuidadosa separación de datos, de las características de las huellas alfa correspondientes a las 2 energías de incidencia para cada energía de referencia. En la figura 6.11 se puede apreciar el resultado obtenido para el parámetro Área. Hay que mencionar que para las energías de referencia de 4.22 MeV y 4.39 MeV fue imposible hacer la separación de los datos, en estos casos se usó la energía de referencia como valor de entrada para las curvas variación de parámetros con la energía.

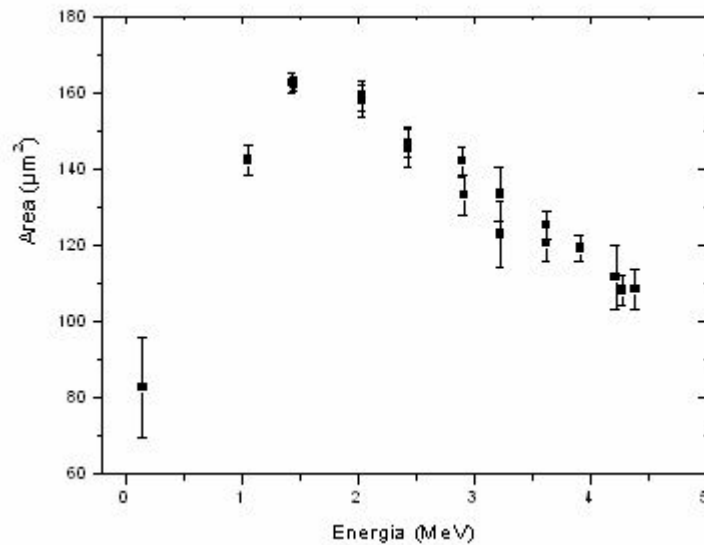


Fig. 6.11. Variación del área de las huellas con respecto a la energía de las partículas incidentes

### 6.1.2. Fuentes de Torio

Procedemos de igual forma que para el caso de las fuentes de Uranio.

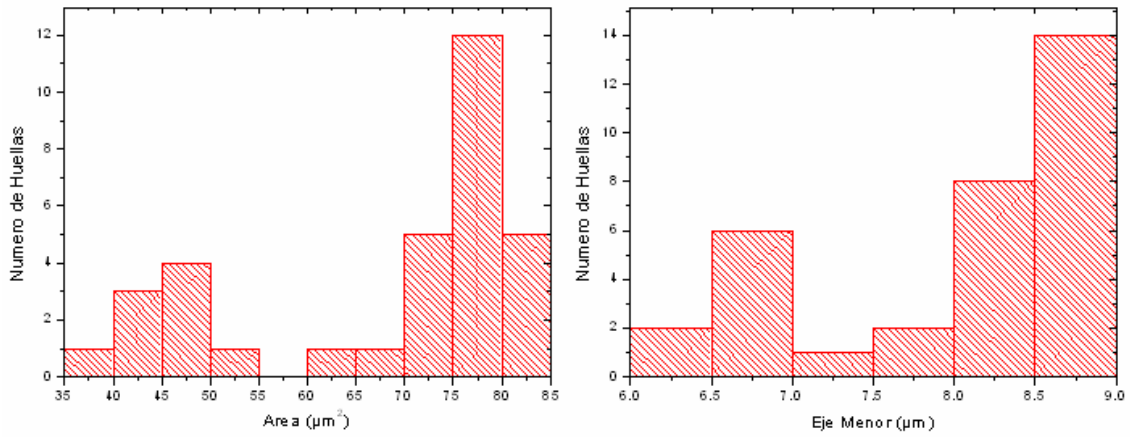


Fig. 6.12. Eref = 4.8 MeV.

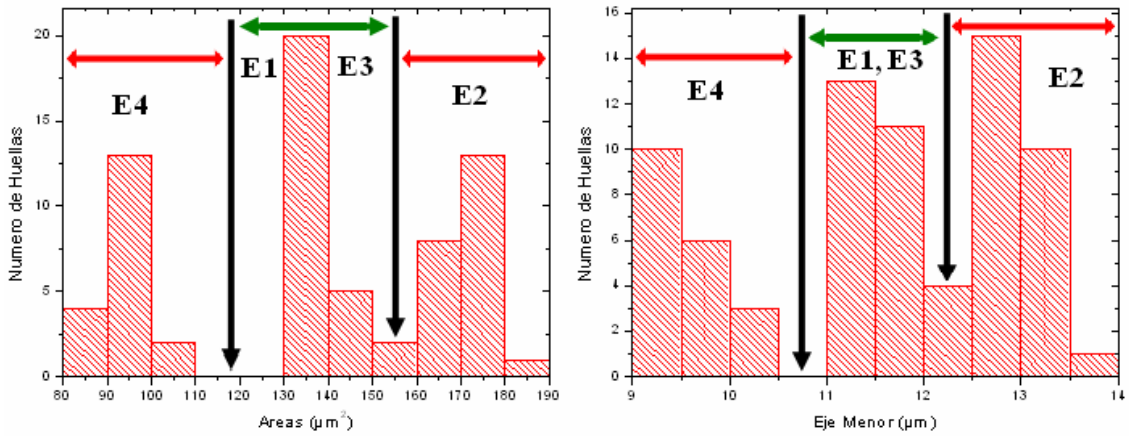


Fig. 6.13. Correlación de los grupos de energía en la configuración Eref = 5.89 MeV con E1 = 0.16 MeV, E2 = 1.9 MeV, E3 = 2.94 MeV, E4 = 5.89 MeV.

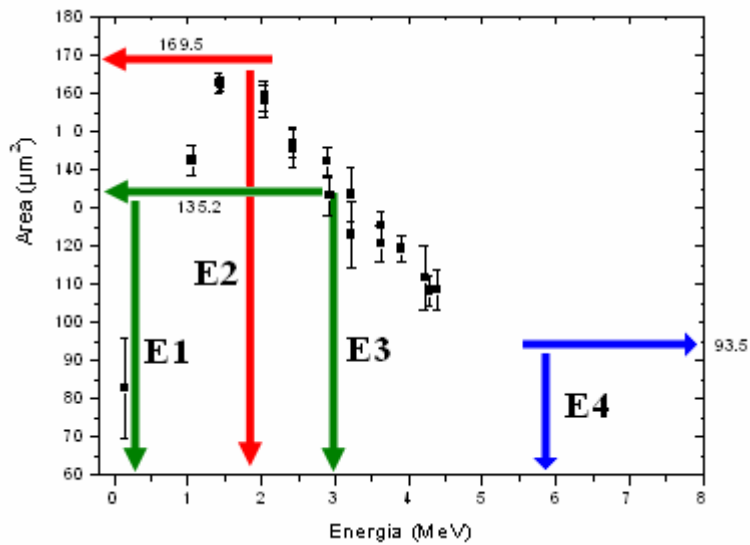


Fig. 6.14. Ubicación adecuada de las energías con Eref = 5.89 MeV con E1 = 0.16 MeV, E2 = 1.9 MeV, E3 = 2.94 MeV, E4 = 5.89 MeV.

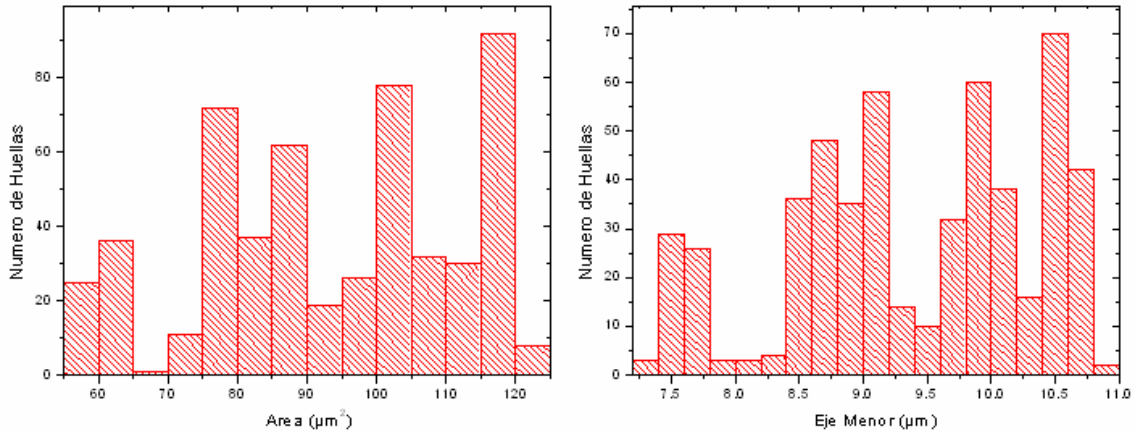


Fig.6.15. Eref = 7 MeV.

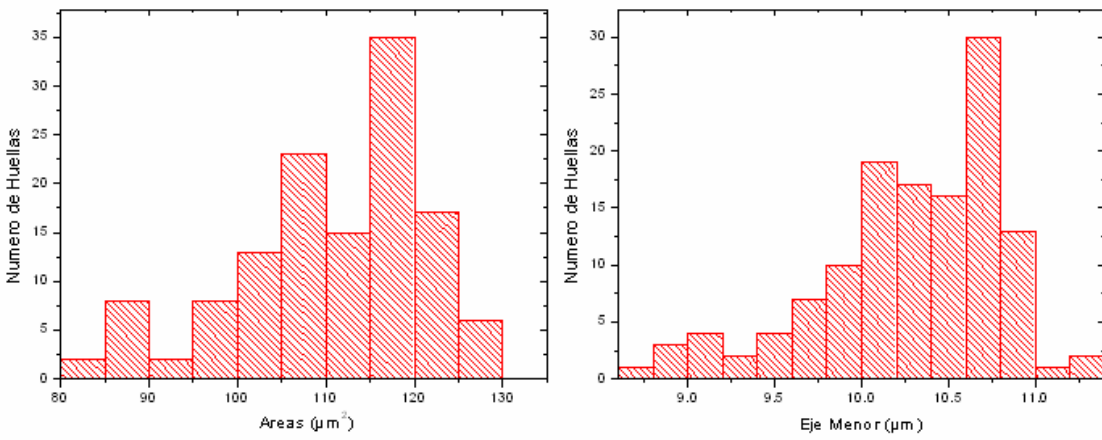


Fig. 6.16. Eref = 8 MeV.

**Tabla 6.1.**

**Resultados de los parámetros en función de la energía de las partículas alfa incidentes**

Energía (MeV)	Área ( $\mu\text{m}^2$ )	$\sigma_{\text{Area}}$ ( $\mu\text{m}^2$ )	Eje Mayor ( $\mu\text{m}$ )	$\sigma_{\text{Eje Mayor}}$ ( $\mu\text{m}$ )	Eje Menor ( $\mu\text{m}$ )	$\sigma_{\text{Eje Menor}}$ ( $\mu\text{m}$ )	Elongaci.	$\sigma_{\text{Elong}}$	Perímetro ( $\mu\text{m}$ )	$\sigma_{\text{Perim}}$ ( $\mu\text{m}$ )	D. Equi ( $\mu\text{m}$ )	$\sigma_{\text{DEqui}}$
0.15	82.56	13.17	11.74	0.97	8.89	0.71	0.758	0.020	33.67	2.80	10.22	0.82
1.45	162.44	2.05	16.61	0.19	12.44	0.11	0.749	0.009	47.68	0.35	14.38	0.09
1.06	142.47	4.09	15.58	0.30	11.65	0.18	0.748	0.010	44.65	0.75	13.47	0.20
2.04	157.98	4.14	16.44	0.26	12.24	0.15	0.745	0.008	47.01	0.68	14.18	0.18
1.43	162.63	2.43	16.60	0.13	12.45	0.13	0.750	0.008	47.67	0.37	14.39	0.11
2.43	146.98	3.87	15.76	0.24	11.86	0.17	0.753	0.009	45.28	0.62	13.68	0.18
2.04	159.29	4.11	16.43	0.21	12.31	0.12	0.749	0.007	47.21	0.52	14.24	0.14
2.91	133.23	5.42	14.94	0.32	11.32	0.25	0.757	0.013	43.05	0.87	13.02	0.26
2.43	145.49	5.13	15.67	0.25	11.79	0.26	0.753	0.007	45.11	0.80	13.61	0.24
3.23	123.02	8.80	14.35	0.50	10.88	0.39	0.758	0.006	41.31	1.46	12.51	0.44
2.9	141.97	4.01	15.60	0.34	11.59	0.24	0.743	0.017	44.65	0.89	13.44	0.14
3.63	125.42	3.74	14.67	0.38	10.87	0.15	0.747	0.014	41.88	0.73	12.64	0.19
3.22	133.50	7.09	15.11	0.47	11.21	0.32	0.743	0.021	43.22	1.11	13.03	0.35
3.91	119.24	3.42	14.21	0.22	10.70	0.19	0.753	0.011	40.71	0.58	12.32	0.15
3.62	120.56	4.56	14.33	0.43	10.70	0.25	0.747	0.027	41.00	0.84	12.39	0.23
4.28	108.25	3.91	13.56	0.32	10.12	0.22	0.747	0.021	38.82	0.77	11.74	0.21
4.22	111.65	8.17	13.56	0.62	10.49	0.62	0.765	0.031	39.32	1.46	11.92	0.43
4.39	108.53	5.28	13.45	0.50	10.28	0.30	0.765	0.034	38.86	1.20	11.77	0.32
5.88	93.53	4.86	12.44	0.27	9.55	0.31	0.768	0.014	36.00	1.02	10.91	0.28
1.9	169.60	6.80	16.86	0.38	12.82	0.37	0.761	0.022	48.68	0.97	14.69	0.30
7.89	88.17	3.98	12.16	0.33	9.16	0.27	0.754	0.023	34.96	0.76	10.59	0.24

Tabla 6.2.

Resultados de los parámetros en función de la energía de las partículas alfa incidentes

Energía (MeV)	Redon	$\sigma_{\text{Redon}}$	NGMaxE	$\sigma_{\text{NGMaxE}}$	NGMinE	$\sigma_{\text{NGMinE}}$	NGME	$\sigma_{\text{NGME}}$	NGPE	$\sigma_{\text{NGPE}}$	SNGPE	$\sigma_{\text{SNGPE}}$
0.15	0.910	0.010	0.800	0.018	0.416	0.051	0.636	0.022	0.635	0.022	0.022	0.072
1.45	0.898	0.006	0.791	0.018	0.354	0.023	0.619	0.016	0.623	0.019	0.019	0.075
1.06	0.898	0.011	0.816	0.013	0.407	0.031	0.628	0.019	0.621	0.022	0.022	0.073
2.04	0.898	0.004	0.808	0.036	0.376	0.060	0.613	0.033	0.610	0.031	0.031	0.074
1.43	0.899	0.006	0.813	0.014	0.387	0.030	0.615	0.015	0.610	0.019	0.019	0.074
2.43	0.902	0.004	0.828	0.018	0.407	0.029	0.625	0.013	0.623	0.013	0.013	0.074
2.04	0.898	0.006	0.826	0.017	0.410	0.018	0.623	0.015	0.621	0.013	0.013	0.073
2.91	0.903	0.006	0.825	0.018	0.392	0.023	0.623	0.015	0.621	0.013	0.013	0.075
2.43	0.898	0.005	0.824	0.016	0.402	0.030	0.612	0.019	0.611	0.016	0.016	0.076
3.23	0.905	0.006	0.838	0.018	0.400	0.021	0.632	0.021	0.628	0.019	0.019	0.074
2.9	0.895	0.011	0.816	0.040	0.369	0.056	0.613	0.037	0.606	0.035	0.035	0.075
3.63	0.898	0.010	0.809	0.027	0.363	0.027	0.616	0.017	0.611	0.017	0.017	0.075
3.22	0.898	0.008	0.800	0.052	0.296	0.148	0.597	0.056	0.590	0.056	0.056	0.091
3.91	0.904	0.005	0.811	0.029	0.357	0.024	0.613	0.023	0.609	0.024	0.024	0.076
3.62	0.901	0.009	0.791	0.038	0.328	0.050	0.598	0.034	0.598	0.034	0.034	0.077
4.28	0.903	0.008	0.796	0.025	0.320	0.056	0.597	0.029	0.599	0.027	0.027	0.079
4.22	0.907	0.007	0.761	0.081	0.269	0.117	0.575	0.094	0.569	0.096	0.096	0.085
4.39	0.907	0.008	0.809	0.024	0.207	0.129	0.573	0.053	0.564	0.055	0.055	0.101
5.88	0.907	0.009	0.759	0.092	0.353	0.067	0.581	0.086	0.589	0.087	0.087	0.075
1.9	0.899	0.008	0.810	0.057	0.379	0.065	0.624	0.044	0.632	0.047	0.047	0.073
7.89	0.906	0.007	0.794	0.056	0.306	0.054	0.607	0.056	0.610	0.057	0.057	0.079

- Redon : Redondez
- NGMaxE : Nivel de gris máximo del entorno.
- NGMinE : Nivel de gris mínimo del entorno.
- NGME : Nivel de gris mediana del entorno.
- NGPE : Nivel de gris promedio del entorno.
- SNGPE : Desviación estándar del nivel de gris promedio del entorno.

**Tabla 6.3.**

**Resultados de los parámetros en función de la energía de las partículas alfa incidentes**

Energía (MeV)	NGMaxH	$\sigma_{\text{NGMaxH}}$	NGMinH	$\sigma_{\text{NGMinH}}$	NGMH	$\sigma_{\text{NHMH}}$	NGPH	$\sigma_{\text{NGPH}}$	SNGPH	$\sigma_{\text{SNGPH}}$
0.15	0.689	0.033	0.077	0.045	0.376	0.030	0.381	0.042	0.100	0.010
1.45	0.614	0.020	0.065	0.024	0.263	0.010	0.272	0.010	0.078	0.004
1.06	0.677	0.036	0.040	0.029	0.320	0.020	0.333	0.020	0.109	0.005
2.04	0.648	0.045	0.037	0.033	0.260	0.019	0.271	0.019	0.084	0.006
1.43	0.633	0.028	0.044	0.032	0.275	0.013	0.291	0.016	0.099	0.006
2.43	0.658	0.026	0.025	0.023	0.241	0.012	0.252	0.013	0.082	0.002
2.04	0.657	0.021	0.054	0.031	0.260	0.010	0.274	0.011	0.088	0.003
2.91	0.665	0.037	0.008	0.016	0.214	0.015	0.224	0.017	0.082	0.003
2.43	0.657	0.026	0.023	0.020	0.229	0.015	0.240	0.016	0.082	0.004
3.23	0.672	0.039	0.007	0.013	0.195	0.024	0.203	0.023	0.084	0.004
2.9	0.669	0.070	0.006	0.021	0.221	0.016	0.229	0.016	0.078	0.005
3.63	0.647	0.029	0.000	0.000	0.139	0.020	0.154	0.017	0.090	0.005
3.22	0.641	0.049	0.000	0.000	0.172	0.042	0.183	0.034	0.085	0.010
3.91	0.630	0.062	0.000	0.000	0.106	0.020	0.125	0.018	0.097	0.005
3.62	0.597	0.042	0.005	0.011	0.151	0.042	0.166	0.039	0.084	0.008
4.28	0.590	0.048	0.002	0.007	0.087	0.047	0.116	0.038	0.094	0.009
4.22	0.590	0.072	0.009	0.027	0.109	0.067	0.134	0.062	0.092	0.013
4.39	0.614	0.069	0.003	0.011	0.101	0.038	0.126	0.032	0.096	0.009
5.88	0.618	0.080	0.005	0.018	0.121	0.081	0.153	0.072	0.116	0.023
1.9	0.678	0.050	0.062	0.033	0.279	0.031	0.291	0.031	0.094	0.012
7.89	0.644	0.066	0.000	0.000	0.115	0.070	0.145	0.064	0.112	0.015

- NGMaxH : Nivel de gris máximo de la Huella.
- NGMinE : Nivel de gris mínimo de la Huella.
- NGME : Nivel de gris mediana de la Huella.
- NGPE : Nivel de gris promedio de la Huella.
- SNGPE : Desviación estándar del nivel de gris promedio de la Huella.

**Tabla 6.4.**

Resultados de los parámetros en función de la energía de las partículas alfa incidentes

Energía (MeV)	Contraste Absoluto	$\sigma_{\text{ContAbs}}$	Contraste Relativo	$\sigma_{\text{ContRela}}$	Contraste Propor.	$\sigma_{\text{ContProp}}$	Skewness	$\sigma_{\text{Skew}}$	Kurtosis	$\sigma_{\text{Kurt}}$
0.15	0.261	0.038	0.251	0.030	0.599	0.037	2.78	0.23	0.07	0.20
1.45	0.350	0.013	0.392	0.021	0.437	0.022	3.95	0.39	0.65	0.13
1.06	0.288	0.018	0.302	0.021	0.537	0.025	2.65	0.15	0.34	0.06
2.04	0.339	0.024	0.385	0.023	0.445	0.024	4.06	0.50	0.70	0.10
1.43	0.316	0.021	0.352	0.022	0.480	0.024	2.85	0.16	0.48	0.08
2.43	0.373	0.014	0.426	0.019	0.403	0.019	5.41	0.71	1.05	0.15
2.04	0.347	0.014	0.388	0.016	0.441	0.017	3.72	0.35	0.77	0.09
2.91	0.398	0.023	0.471	0.030	0.360	0.029	6.76	0.95	1.32	0.19
2.43	0.371	0.015	0.437	0.021	0.392	0.021	6.05	1.16	1.21	0.28
3.23	0.425	0.024	0.513	0.040	0.323	0.035	7.69	1.14	1.51	0.26
2.9	0.377	0.027	0.452	0.021	0.378	0.020	6.65	1.58	1.19	0.28
3.63	0.459	0.019	0.602	0.031	0.246	0.021	8.32	0.77	1.92	0.15
3.22	0.407	0.051	0.528	0.062	0.311	0.051	7.08	1.46	1.44	0.38
3.91	0.484	0.028	0.660	0.040	0.205	0.030	8.13	2.07	1.98	0.45
3.62	0.431	0.046	0.568	0.071	0.278	0.064	6.91	1.25	1.62	0.28
4.28	0.483	0.039	0.680	0.083	0.194	0.061	7.30	1.19	1.97	0.29
4.22	0.435	0.102	0.623	0.136	0.241	0.111	7.17	2.01	1.85	0.52
4.39	0.429	0.065	0.635	0.076	0.226	0.058	7.21	1.87	1.86	0.50
5.88	0.436	0.112	0.594	0.149	0.266	0.128	5.51	2.33	1.53	0.74
1.9	0.340	0.039	0.369	0.040	0.462	0.043	3.49	0.59	0.65	0.18
7.89	0.465	0.085	0.624	0.139	0.241	0.111	5.92	2.61	1.56	0.78

Figura 6.17.

Variación de los parámetros de las huellas como función de la energía de las partículas alfa Incidentes

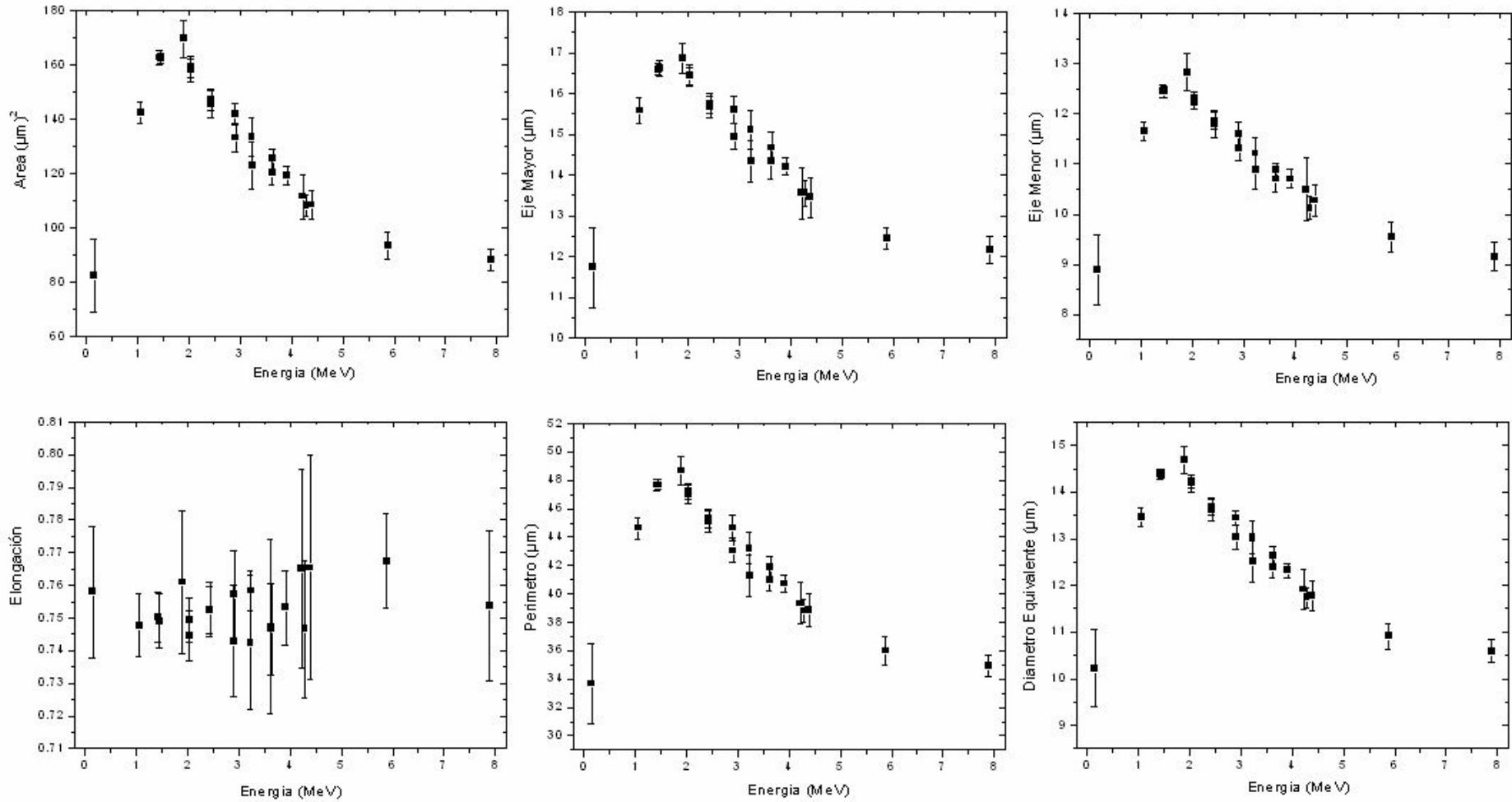




Figura 6.18

Variación de los parámetros de las huellas como función de la energía de las partículas alfa Incidentes

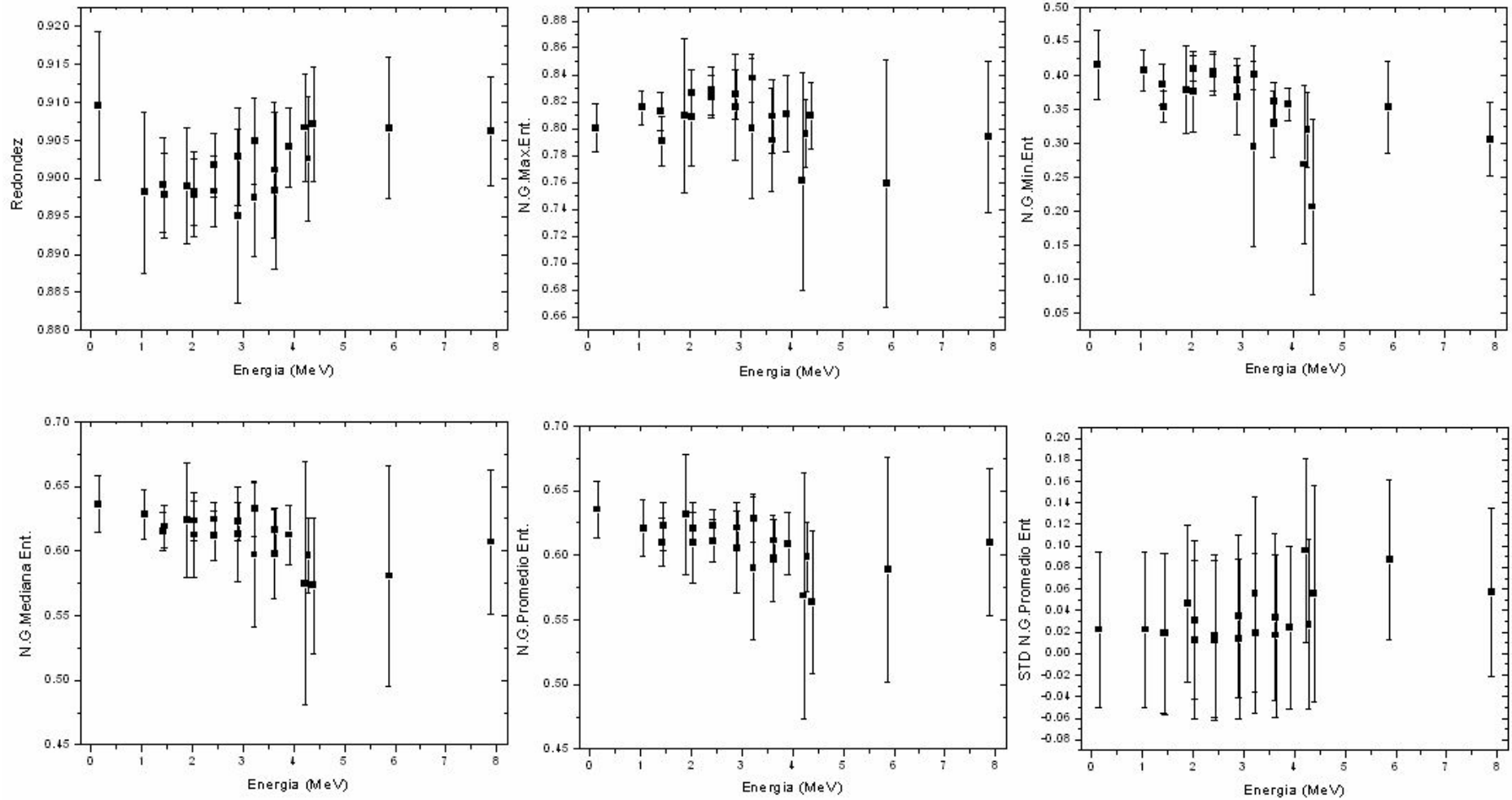


Figura 6.19

Variación de los parámetros de las huellas como función de la energía de las partículas alfa Incidentes

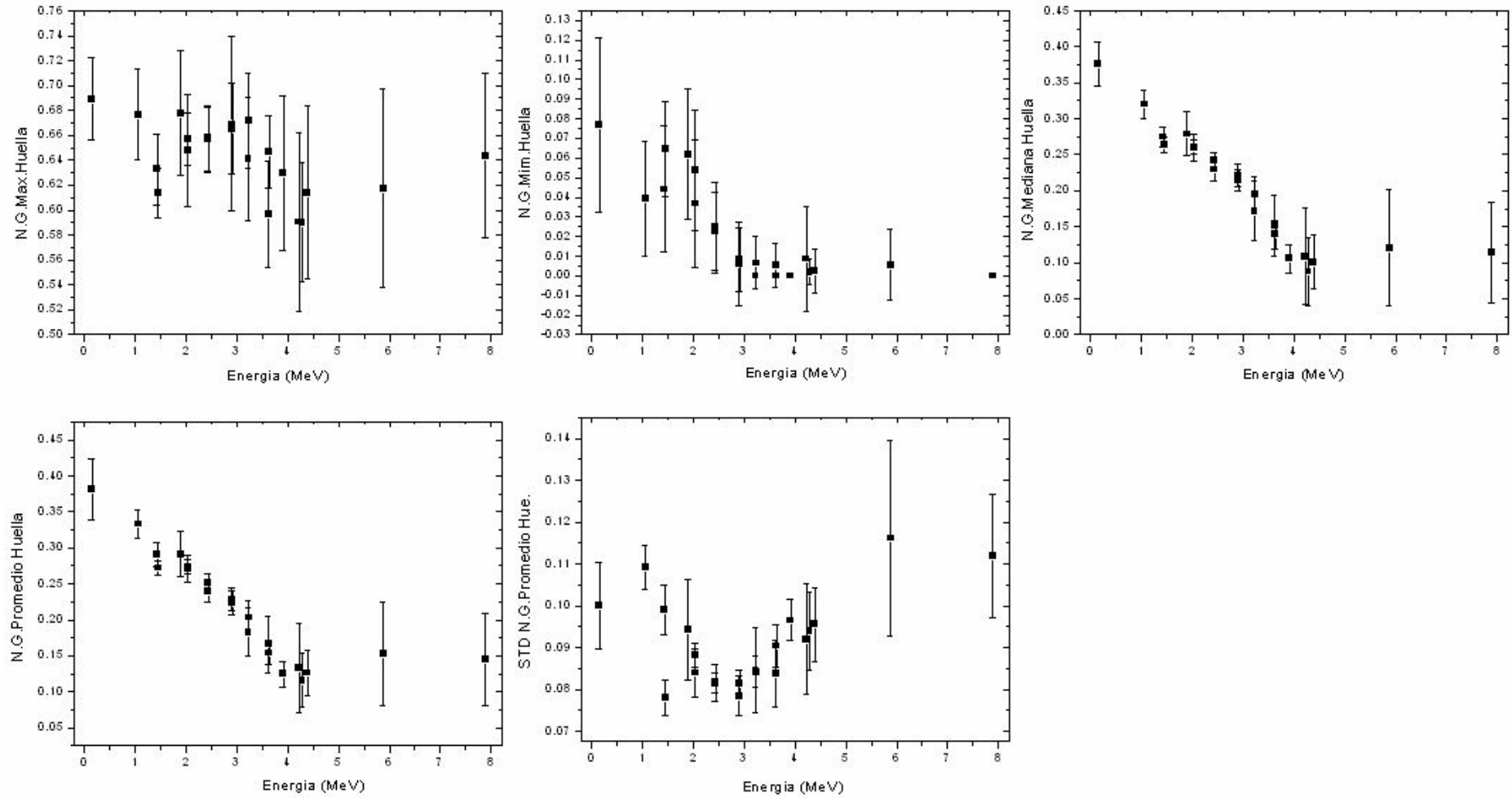
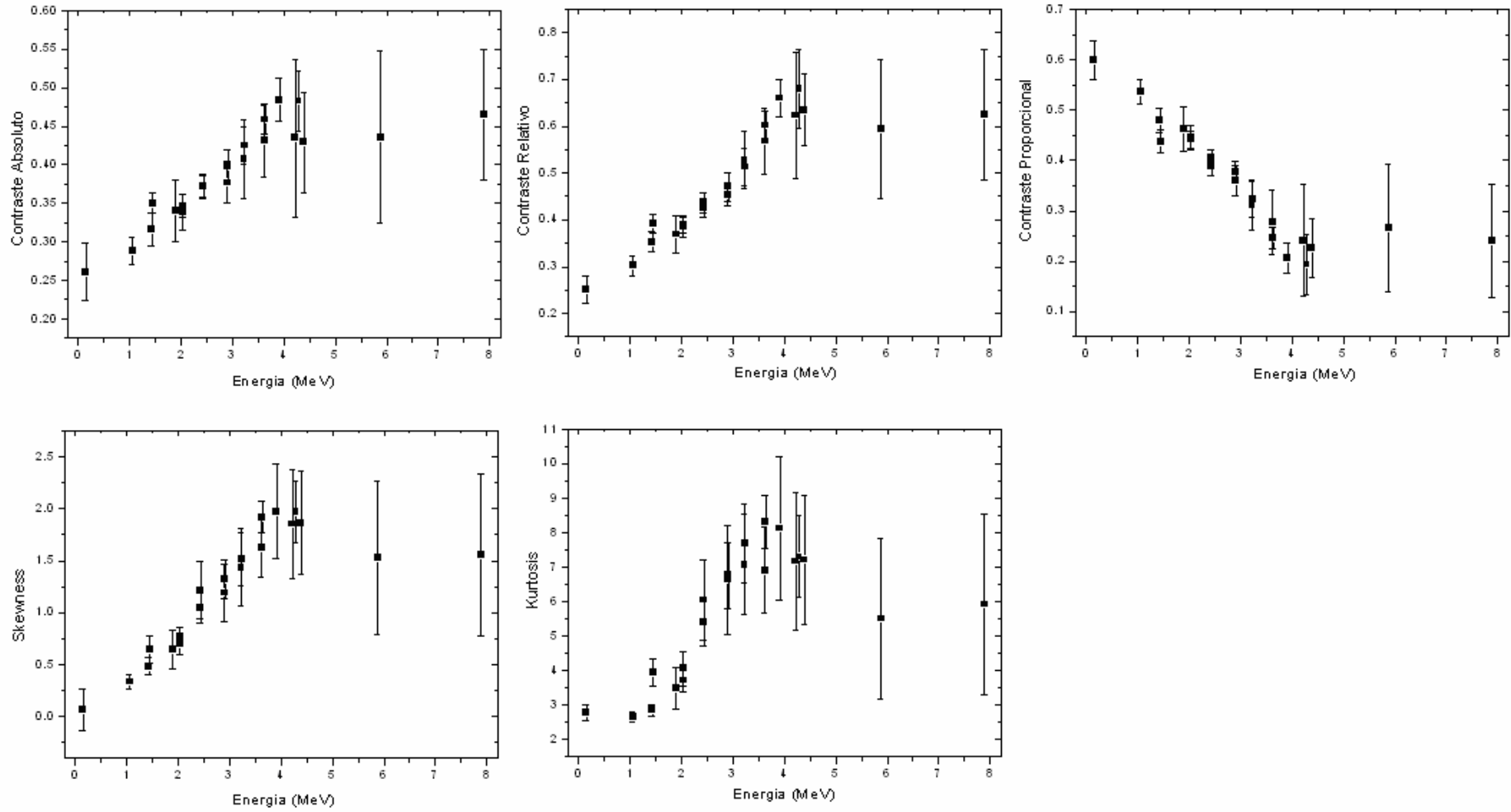


Figura 6.20.

Variación de los parámetros de las huellas como función de la energía de las partículas alfa Incidentes



## 6.2. Clasificación de las huellas nucleares alfa a través de un clasificador basado en RNAs

### 6.2.1. Definir RNA a utilizar

En este trabajo se utilizó una red neuronal de retropropagación de tres capas como la mostrada en la figura 2.19. Donde la capa de entrada recibe las características de las huellas identificadas en el proceso de extracción de características (área, eje mayor, eje menor, elongación, perímetro... etc.), por lo tanto el número de neuronas en la capa de entrada viene ser igual al número de características extraídas a las huellas (21 en total), una capa oculta establecerá las fronteras de clasificación y una capa de salida determinará la clase o categoría a la que pertenece la huella. De la tabla 6.1 se observa que existen 21 posibles categorías, es decir 21 posibles valores de energía en el que se puede clasificar la huella, por lo tanto tomamos 21 neuronas en la capa de salida.

Se tomó la decisión de considerar sólo 3 capas, tomando como referencia el teorema de Kolmogorov [86], que afirma: "Dada cualquier función continua  $f : [0, 1]^n \rightarrow R^m$ ,  $f(x) = y$ , existe una RNA de 3 capas, de propagación hacia delante sin retroalimentación, con  $n$  elementos de proceso en la capa de entrada,  $m$  en la de salida y  $(2n+1)$  en la capa oculta; que implementa dicha función de forma exacta". Esto implica que el número de neuronas de la capa intermedia dependerá de la capa de entrada. Sin embargo, el Teorema de Kolmogorov es un teorema de existencia. No hay ninguna técnica que nos diga cómo obtener la arquitectura de red para un determinado problema, ni mucho menos cuál es la arquitectura de red óptima para el problema. Por lo tanto nuestra red neurona queda definida con 22 neuronas en la capa de entrada, 45 neuronas en la capa oculta y 21 neuronas en la capa de salida, en la figura 6.21 se puede ver una representación de la red neuronal seleccionada.

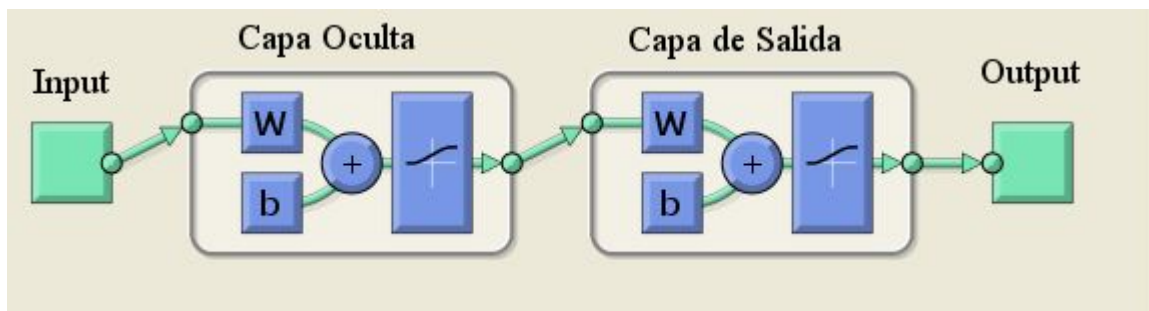


Fig. 6.21. RNA seleccionada con 22 neuronas en la capa de entrada, 45 neuronas en la capa oculta y 21 neuronas en la capa de salida.

### 6.2.2. Definir patrones de entrenamiento

Para llevar a cabo el entrenamiento de la RNA y que ésta presente un buen desempeño, se dividió la matriz de características arrojada por el proceso de detección de huellas en tres subconjuntos: huellas que pertenecen al subconjunto de entrenamiento, los que forman parte del subconjunto de validación y los que constituyen el subconjunto de prueba. En la tabla 6.5 se muestra la división de la data total (tomando 2 sigmas alrededor del valor medio) en tres subconjuntos: Entrenamiento, validación y prueba

Data Total	500
Data Entrenamiento	350
Data Validación	75
Data Prueba	75

Tabla 6.5. Distribución de la data total para los tres subconjuntos: Entrenamiento, validación y prueba

### 6.2.3. Preprocesamiento de la información

El primer método de preprocesamiento de la data y el más común en RNA consiste en un simple reescalamiento lineal de las variables de entrada. Este proceso es de gran utilidad si se tiene variables con valores que difieren significativamente entre ellas ya que los tamaños de las entradas podrían no reflejar la importancia relativa para determinar las salidas requeridas.

	1	2	3	4	5	6	7	8	...	18	19	20	21	22
1	63.2426	10.3920	7.6102	0.7323	29.5677	8.9735	0.9090	0.8000		0.2337	0.2220	0.6367	2.8262	0.0234
2	65.2321	10.3311	8.1071	0.7847	29.6326	9.1135	0.9335	0.7961		0.2330	0.2210	0.6380	2.5226	-0.0237
3	69.5250	10.7746	8.2265	0.7635	30.9342	9.4086	0.9130	0.8039		0.2643	0.2511	0.5985	2.8036	-0.1000
4	72.9803	10.9055	8.4698	0.7767	31.6258	9.6396	0.9169	0.8118		0.2465	0.2373	0.6165	2.6200	-0.0523
5	73.1897	11.4149	8.1056	0.7101	32.0133	9.6534	0.8974	0.8078		0.3982	0.4541	0.3754	2.4260	0.1690
6	73.3468	10.9825	8.4207	0.7667	31.8520	9.6637	0.9085	0.7569		0.2732	0.2986	0.5401	2.7707	0.5351
7	74.6033	11.1959	8.5529	0.7639	31.8279	9.7462	0.9254	0.8078		0.2477	0.2290	0.6273	2.6547	-0.0790
8	79.3151	11.2983	8.7755	0.7767	32.9274	10.0492	0.9193	0.8118		0.2565	0.2494	0.6008	2.8337	-0.1908
9	79.9957	11.5034	9.1350	0.7941	33.3112	10.0923	0.9059	0.7882		0.2657	0.2522	0.5972	2.7414	0.0635
10	81.5139	11.7830	8.8138	0.7480	33.4892	10.1876	0.9133	0.8039	...	0.2675	0.2550	0.5936	2.8189	-0.0414
11	84.1316	11.8231	9.0888	0.7687	34.0473	10.3499	0.9120	0.8000		0.2314	0.2462	0.6049	2.3226	0.3121
12	85.9639	11.9900	9.1174	0.7604	34.5646	10.4620	0.9042	0.8431		0.2301	0.2240	0.6340	2.8294	-0.0876
13	86.6969	12.0257	9.1413	0.7601	34.5646	10.5065	0.9119	0.8039		0.2765	0.2785	0.5644	3.3562	0.0675
14	86.9586	12.1299	9.0160	0.7433	34.9243	10.5223	0.8959	0.7961		0.2322	0.2307	0.6252	2.7603	0.0075
15	89.0004	12.2335	9.1840	0.7507	35.0819	10.6451	0.9087	0.7569		0.2770	0.3026	0.5354	3.0981	0.5446
16	89.5239	12.2022	9.3253	0.7642	35.2154	10.6764	0.9072	0.8196		0.3002	0.3108	0.5258	2.7588	0.3531
17	92.6651	12.3490	9.5150	0.7705	35.7735	10.8621	0.9099	0.8000		0.2738	0.2650	0.5811	2.8308	0.0947
18	100.8846	13.0406	9.9693	0.7645	37.2327	11.3336	0.9145	0.7961		0.2194	0.2152	0.6458	2.9032	-0.1032
19	101.0416	13.0738	9.8241	0.7514	37.4348	11.3424	0.9061	0.8000		0.2874	0.2818	0.5603	2.9336	0.0273
20	104.0781	13.5994	9.7123	0.7142	38.4212	11.5116	0.8860	0.7961		0.2613	0.2567	0.5914	2.7027	0.1451
21	104.2875	13.2087	9.9924	0.7565	38.0856	11.5232	0.9035	0.8000		0.2458	0.2472	0.6036	3.1378	-0.0488
22	137.5842	15.1917	11.4303	0.7524	43.6443	13.2355	0.9077	0.8118		0.3161	0.3148	0.5212	2.6777	0.3379
23	139.1024	15.3613	11.4202	0.7434	44.1375	13.3083	0.8973	0.8471		0.2813	0.2823	0.5597	2.6339	0.3929
24	140.1495	15.4359	11.5988	0.7514	44.1375	13.3583	0.9040	0.8078		0.2751	0.2955	0.5438	2.5310	0.2762
25	140.1495	15.4359	11.5988	0.7514	44.1375	13.3583	0.9040	0.8078	...	0.2751	0.2955	0.5438	2.5310	0.2762
26	140.1495	15.4359	11.5988	0.7514	44.1375	13.3583	0.9040	0.8078		0.2751	0.2955	0.5438	2.5310	0.2762
27	141.5630	15.4665	11.6970	0.7563	44.4045	13.4255	0.9022	0.8039		0.2859	0.2957	0.5435	2.4307	0.3807
			⋮								⋮			
494	88.4769	11.8231	9.4571	0.7999	34.7871	10.6138	0.9188	0.8196		0.5034	0.6691	0.1983	5.2257	0.2651
495	93.5028	12.4772	9.4874	0.7604	35.8662	10.9111	0.9134	0.8314		0.5261	0.7677	0.1314	8.7675	0.4159
496	94.6545	12.9648	9.2591	0.7142	36.4280	10.9781	0.8964	0.6431		0.2542	0.3721	0.4576	2.8473	0.2695
497	95.7016	12.5114	9.7241	0.7772	36.2667	11.0386	0.9144	0.8196		0.4019	0.4295	0.3991	2.9828	0.3285
498	94.6022	12.5114	9.5564	0.7638	36.1332	10.9750	0.9105	0.6980	...	0.3201	0.5195	0.3162	4.4139	0.3139
499	94.8640	12.5114	9.6403	0.7705	36.2908	10.9902	0.9051	0.8471		0.5505	0.7583	0.1374	7.6299	0.3379
500	95.8587	12.6409	9.6598	0.7642	36.6505	11.0477	0.8968	0.8118		0.5567	0.7699	0.1300	8.5046	0.5108

Fig. 6.22. Matriz de características, Data Input.

Con esto en mente, los valores de cada columna de la matriz de características, matriz de dimensiones [500,22] como se muestra en la figura 6.22, resultado del proceso de extracción de características, fueron mapeados para que cayeran en el intervalo [-1,1]. El escalonamiento lineal requiere encontrar el valor mínimo (*Min*) y máximo (*Max*) de cada variable.

La formula usada para convertir los datos originales D en datos normalizados N es:

$$N = 1 - 2 \frac{Max - D}{Max - Min} \quad (6.1)$$

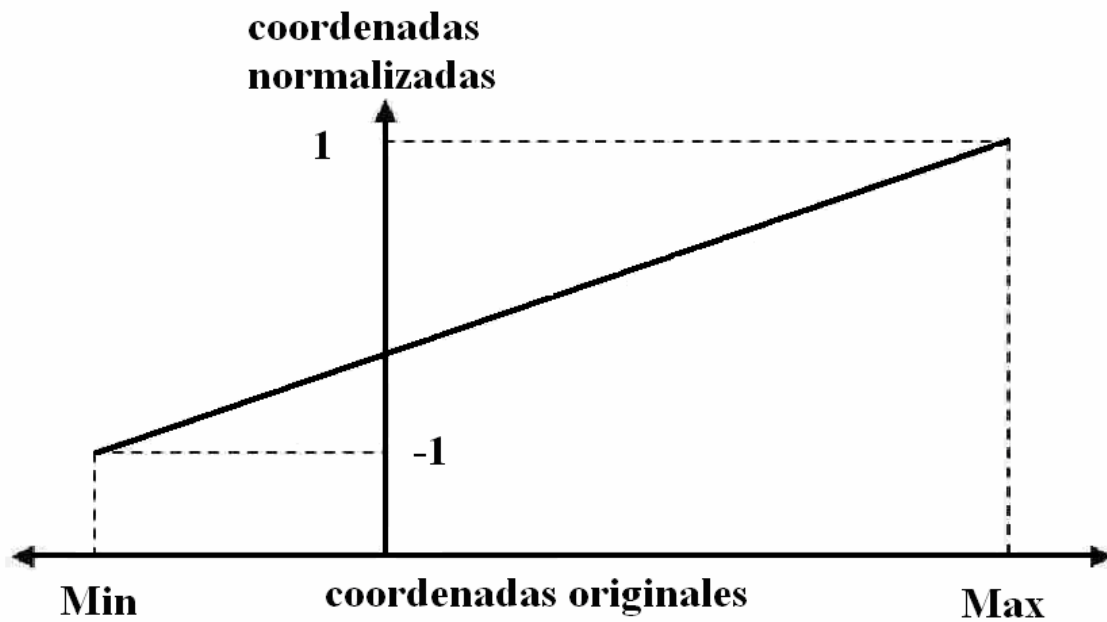


Fig. 6.23. Normalización de datos al rango [-1, 1]

	1	2	3	4	5	6	7	8		17	18	19	20	21	22
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
21	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
489	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
490	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
491	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
492	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
493	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
494	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
495	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
496	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
497	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
498	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
499	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
500	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Fig. 6.24. Matriz de Salida, Data Ouput.

La figura 6.24 muestra la matriz de salida, de dimensiones [500,22], donde la fila 1 de esta matriz, es la salida de la fila 1 de la matriz de características, fila 2 es la salida de la fila 2 de la matriz de características y así sucesivamente. Cada fila de la matriz de salida esta conformado únicamente por ceros y un único uno, la posición de este único uno indica la energía a la que equivale la fila.

Hasta el momento tenemos la siguiente configuración para nuestra red neuronal: 22 neuronas en la capa de entrada, 45 neuronas en la capa oculta y 21 neuronas en la capa de salida, Sin embargo aun falta determinar las funciones transferencia para las configuraciones capa de entrada-capa oculta y capa oculta-capa de salida. Para la elección de las funciones de transferencia se hicieron distintas pruebas con las funciones de transferencias mas comúnmente usadas para este propósito.

Funciones de Transferencia	MSE	T (Seg.)	Inter.	Efecti.
LogSig-PureLin	0.0279	117	10000	0.53
LogSig-TanSig	0.0446	121	10000	0.62
PureLine-Tansig	0.0461	107	10000	0.36
LogSig-LogSig	0.0008	67	5405	0.64
PureLin-PureLin	0.0420	99	10000	0.37
TanSig-TanSig	0.0231	125	10000	0.59
PureLine-Tansig	0.0480	106	10000	0.33
TanSig-LogSig	0.0009	64	5340	0.60

Tabla 6.6. Resultados de modificar las funciones de transferencia entre las distintas capas de la red neuronal.

LogSing : Función de transferencia Logaritmo Sigmoidal.  
 PureLine : Función de transferencia Lineal.  
 TanSig : Función de transferencia Tangente Hiperbólica Sigmoidal.

En la tabla 6.6 se observa los resultados de las distintas configuraciones de funciones de transferencia. Cada entrada de la tabla representa el promedio de 10 interacciones echas para cada configuración. Se utilizó como algoritmo de entrenamiento al algoritmo de *retropropagación de gradiente descendente con tasa de aprendizaje adaptativo* y como función de medida del error el *error cuadrático medio* (MSE). Cabe mencionar que para cada experimento se tomo distintos conjuntos de entrenamiento, validación y prueba.

Finalmente basándonos en las eficiencias de clasificación mostradas en la tabla 6.6 anterior tomamos como función de transferencia para todos los nodos la *Logarítmica sigmoideal* con un 64% de huellas bien clasificadas. Este modelo de clasificador será usado en todos los casos donde se requiera una etapa de clasificación (aprendizaje y simulación), lo que cambiaremos, con el fin de mejorar la eficiencia de clasificación son las características en la entrada y por lo tanto el número de neuronas en la capa de entrada, el número de neuronas de capa intermedia.

En el segundo método de preprocesamiento de la data agrupamos los datos correspondientes a valores de energía con una separación muy pequeña entre ellos (menores de 100 Kev), usando para ello la teoría de propagación de errores. Como se

puede apreciar en la figura 6.25 de la variación del parámetro área con la energía de las partículas alfa incidentes, existen valores de energía que están muy próximos entre si, encerrados en un círculo de color rojo. La figura 6.26 muestra el resultado de la agrupación del parámetro área, en el anexo 9.5 se muestra los resultados para el resto de parámetros.

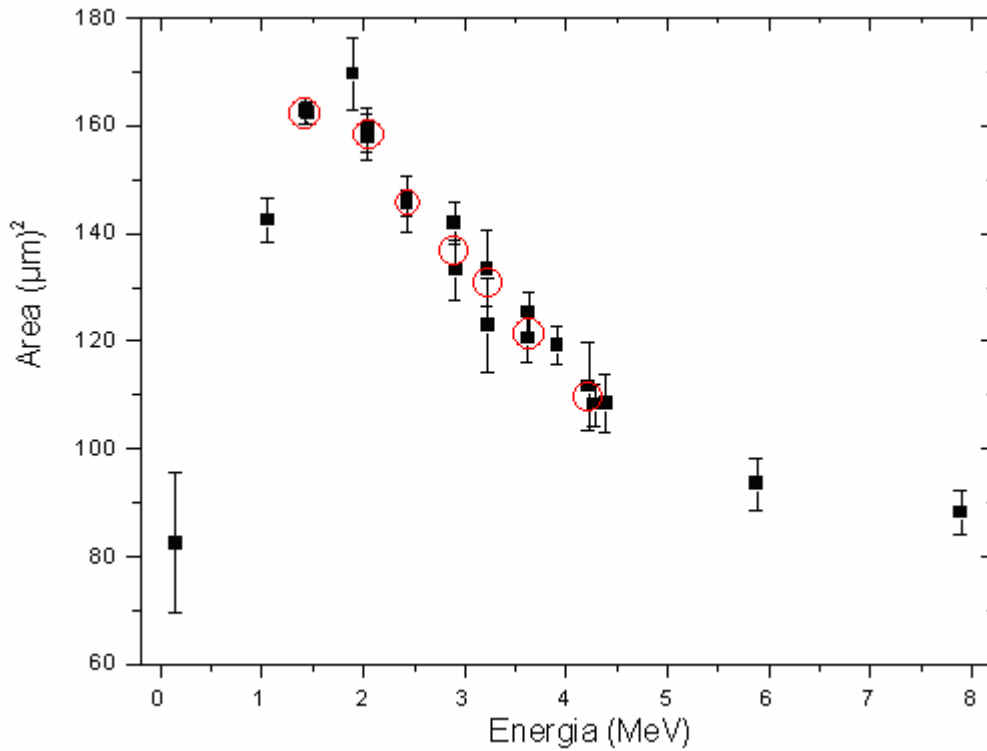


Fig. 6.25. Variación del área con la energía de las partículas alfa incidentes.

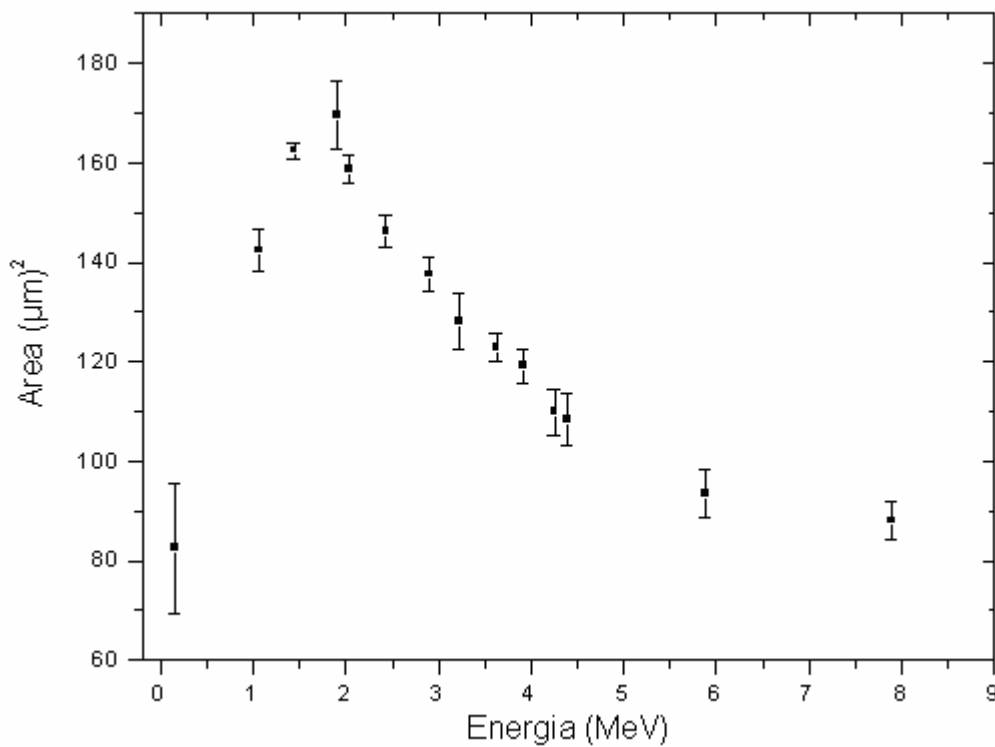


Fig. 6.26. Variación del área con la energía de las partículas alfa incidentes después de haber agrupado los valores de energía cercanos.



Como resultado del proceso anterior el número de valores de energía se reduce de 21 a 14, por lo tanto la el número de neuronas de la capa de salida sera ahora 14.

## 6.2.4. Pruebas del clasificador

### 6.2.4.1. Estudio de la variación del número de neuronas de la capa oculta y los algoritmos de entrenamiento en la clasificación de huellas nucleares alfa

El objetivo de este estudio es evaluar el desempeño de 12 algoritmos de entrenamiento para el clasificador de huellas nucleares alfa basado en RNAs presentado anteriormente en las secciones 6.2.1 a 6.2.3 utilizando como medida de desempeño MSE. Adicionalmente se hizo variar el número de neuronas de la capa intermedia entre 5 y 45 neuronas para cada algoritmo de entrenamiento.

En la tabla 6.7 se presenta los 12 algoritmos de entrenamiento considerados en este estudio, así como los acrónimos respectivamente utilizados para identificarlos

Tabla 6.7. Algoritmos de entrenamiento.

Acrónimo	Descripción
GD	Entrenamiento de retropropagación de gradiente descendente.
GDM	Entrenamiento de retropropagación de gradiente descendente con Momento.
GDA	Entrenamiento de retropropagación de gradiente descendente con tasa de aprendizaje adaptativo.
GDX	Entrenamiento de retropropagación de gradiente descendente con momento y con tasa de aprendizaje adaptativo.
RP	Entrenamiento de retropropagación resiliente.
CGF	Entrenamiento de retropropagación de gradiente conjugado Fletcher-Reeves.
CGP	Entrenamiento de retropropagación de gradiente conjugado Polak-Ribière.
CGB	Entrenamiento de retropropagación de gradiente conjugado Powell-Beale.
SCG	Entrenamiento de retropropagación de gradiente conjugado escalado
BFGS	Entrenamiento de retropropagación Cuasi-Newton.
OSS	Entrenamiento de retropropagación secante de un solo paso.
LM	Entrenamiento de retropropagación Levenberg-Marquardt.

Las figuras 6.27, 6.28, 6.29 y 6.30 así como las tablas 6.8, 6.9 y 6.10 muestran los resultados de este estudio, pudiendo extraer las siguientes observaciones:

- Los algoritmos GD, GDM, GDA, GDX y RP, figuras 6.27 y 6.28, muestran una caída suave el error cuadrático medio (MSE) conforme aumenta el número de neuronas de la capa intermedia hasta un valor de 0.07 en las 2 primeras y 0.003 para las 3 últimas. Para el resto de algoritmos el MSE oscila alrededor de 0.03 no habiendo una disminución del MSE con el aumento de número de neuronas de la capa intermedia.
- Las eficiencias de clasificación de los algoritmos GD y GDM es muy baja con

máximos de 12% y 10 % respectivamente, por otro lado las eficiencias de clasificación de algoritmos GDA, GDX y RP están en promedio alrededor del 75%. Las tres variantes del gradiente conjugado CGF, CGP y GCB tienen eficiencias de clasificación entre 40% y 50%. El resto de algoritmos tienen eficiencias muy variables yendo desde un mínimo de 10% hasta un 70 % como máximo donde las peores eficiencias son para los algoritmos BFGS y OSS.

- En promedio los tiempos de ejecución tomados por los distintos algoritmos de aprendizaje están entre 1 y 2 minutos los casos extremos se observan para los algoritmos de SCG con 10 segundos en promedio y para el algoritmo LM con 4 minutos.

De igual forma que en la tabla 6.6 cada entrada de las tablas 6.8, 6.9 y 6.10 representa el promedio de 10 interacciones hechas para cada configuración.

En conclusión basándonos en el error cuadrático medio, las eficiencias de clasificación alcanzadas y el tiempo de ejecución empleado, los algoritmos que muestran mejor desempeño son GDA, GDX y RP.

Ya que debemos elegir solo uno de los algoritmos anteriormente mencionados (GDA, GDX y RP) procedemos a evaluar el desempeño de los 3 algoritmos de entrenamiento, haciendo variar el número de neuronas de la capa intermedia entre 20 y 45 neuronas, región donde se observa las mejores eficiencias para estos algoritmos, en la tabla 6.11 se muestra los resultados obtenidos. Finalmente basándonos en el error cuadrático medio (0.0016), la eficiencia de clasificación alcanzada (79%) y el tiempo de ejecución empleado (42 s), se decidió tomar como algoritmo de entrenamiento al gradiente descendente con momento y con tasa de aprendizaje adaptativo (GDX) y se fijó el número de neuronas de la capa intermedia en 30.

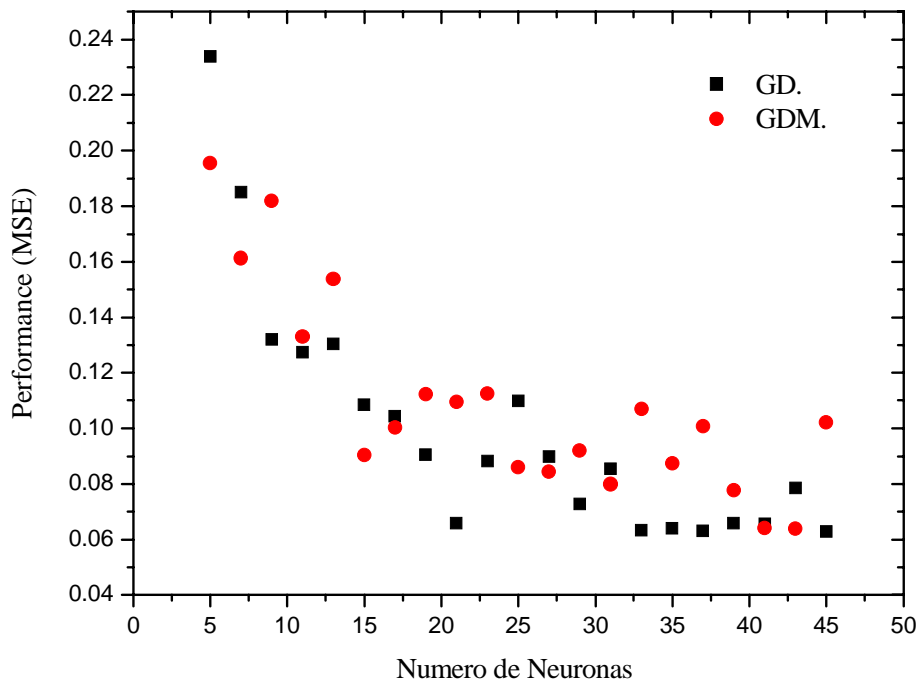


Fig. 6.27. Variación del performance en función del número de neuronas para los algoritmos GD y GDM

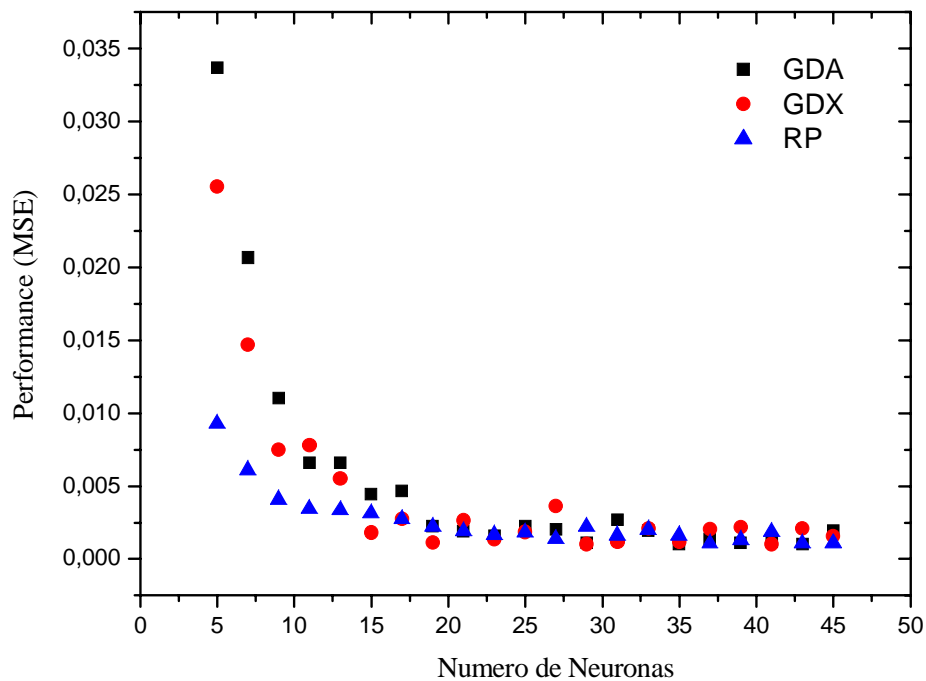


Fig. 6.28. Variación del perfomance en función del número de neuronas para los algoritmos GDA y GDX y RP

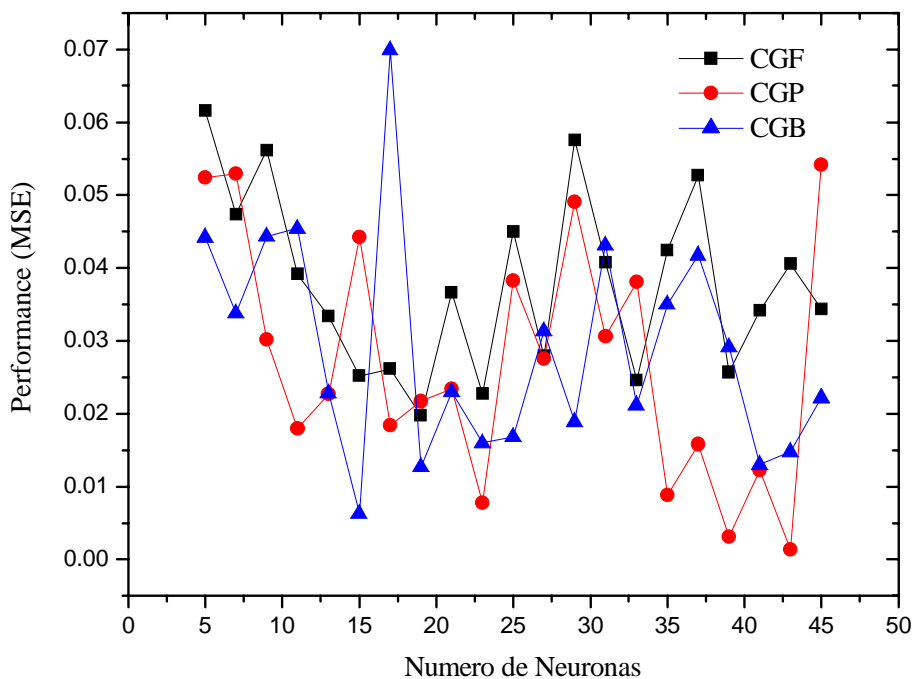


Fig. 6.29. Variación del perfomance en función del número de neuronas para los algoritmos CGF, CGP y GCB

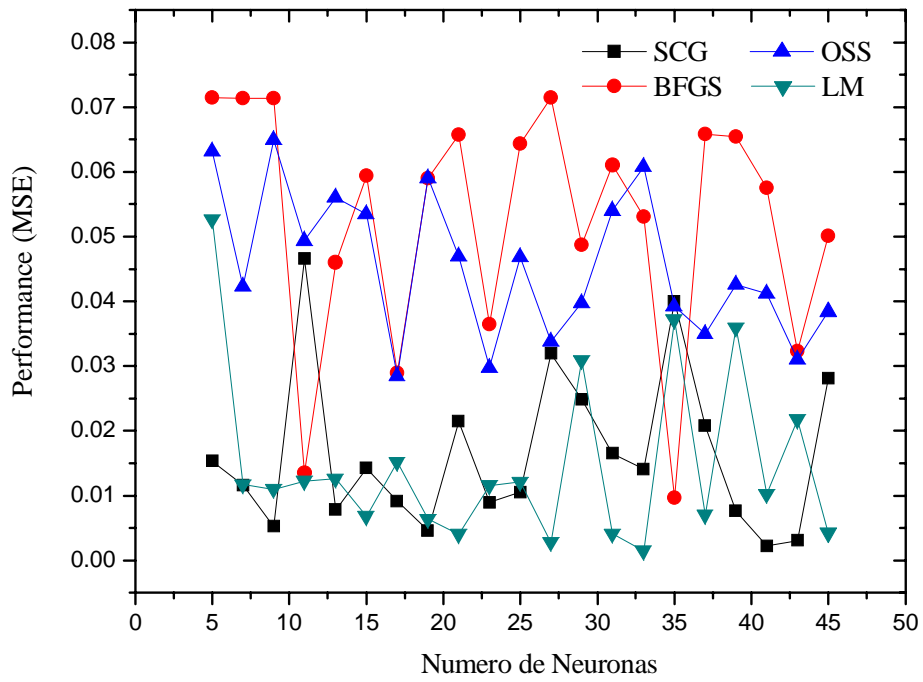


Fig. 6.30. Variación del performance en función del número de neuronas para los algoritmos SCG, OSS, OSS, LM

**Tabla 6.8**  
**Evaluación de los algoritmos GD, GDM, GDA, GDX**

Neuronas	GD				GDM				GDA				GDX			
	MSE	T (Seg.)	Inter.	Efecti.	MSE	T (Seg.)	Inter.	Efecti.	MSE	T (Seg.)	Inter	Efecti.	MSE	T (Seg.)	Inter.	Efecti.
5	0.234	56.2	10000	0.081	0.195	62.2	10000	0.018	0.0337	81.4	10000	0.428	0.0255	80.2	10000	0.581
7	0.185	59.9	10000	0.041	0.161	66.2	10000	0.122	0.0207	84.7	10000	0.626	0.0147	83.2	10000	0.685
9	0.132	59.6	10000	0.036	0.182	69.1	10000	0.023	0.0110	87.1	10000	0.770	0.0075	83.5	10000	0.748
11	0.127	58.9	10000	0.027	0.133	66.3	10000	0.054	0.0066	85.1	10000	0.757	0.0078	83.1	10000	0.653
13	0.130	61.9	10000	0.086	0.154	68.7	10000	0.027	0.0066	86.3	10000	0.680	0.0055	79.1	9435	0.752
15	0.108	68.4	10000	0.059	0.090	76.3	10000	0.104	0.0044	94.8	10000	0.775	0.0018	86.8	9417	0.770
17	0.104	69.8	10000	0.045	0.100	76.2	10000	0.036	0.0047	96.2	10000	0.779	0.0027	45.8	4876	0.775
19	0.090	71.8	10000	0.090	0.112	76.7	10000	0.072	0.0022	98.8	10000	0.752	0.0011	54.7	5696	0.815
21	0.066	75.2	10000	0.077	0.110	83.0	10000	0.059	0.0019	99.8	9916	0.757	0.0026	43.7	4446	0.730
23	0.088	79.2	10000	0.027	0.133	83.7	10000	0.050	0.0016	104.4	10000	0.856	0.0013	58.4	5769	0.788
25	0.110	73.1	10000	0.027	0.086	74.3	10000	0.050	0.0022	89.1	9389	0.757	0.0018	43.6	4400	0.653
27	0.090	80.0	10000	0.068	0.084	85.3	10000	0.081	0.0020	77.7	7721	0.793	0.0036	42.1	4081	0.712
29	0.073	83.2	10000	0.113	0.122	94.3	10000	0.032	0.0011	91.8	8293	0.806	0.0010	26.7	2492	0.793
31	0.085	87.5	10000	0.113	0.080	95.4	10000	0.090	0.0027	89.1	7978	0.775	0.0011	48.8	4428	0.833
33	0.063	89.0	10000	0.122	0.107	95.8	10000	0.140	0.0019	109.6	9605	0.815	0.0021	35.4	3119	0.766
35	0.064	92.0	10000	0.090	0.087	99.2	10000	0.095	0.0010	92.2	7824	0.730	0.0011	38.6	3384	0.775
37	0.063	94.4	10000	0.117	0.101	102.2	10000	0.099	0.0014	89.8	7556	0.743	0.0020	46.7	3989	0.770
39	0.066	96.6	10000	0.095	0.078	101.7	10000	0.131	0.0011	93.9	7669	0.788	0.0022	42.8	3576	0.793
41	0.065	100.0	10000	0.122	0.064	100.3	10000	0.131	0.0012	112.0	8989	0.788	0.0010	27.3	2231	0.797
43	0.078	98.9	10000	0.090	0.064	103.1	10000	0.117	0.0010	106.0	8378	0.802	0.0021	44.2	3534	0.725
45	0.063	101.3	10000	0.104	0.102	109.8	10000	0.059	0.0019	95.9	7486	0.784	0.0016	43.5	3446	0.739

GD : Gradiente descendente.

GDM : Gradiente descendente con Momento.

GDA : Gradiente descendente con tasa de aprendizaje adaptativo.

GDX : Gradiente descendente con Momento y con tasa de aprendizaje adaptativo

**Tabla 6.9.**  
**Evaluación de los algoritmos RP, CGF, CGP, CGB**

Neuronas	RP				CGF				CGP				CGB			
	MSE	T (Seg.)	Inter.	Efecti.	MSE	T (Seg.)	Inter.	Efecti.	MSE	T (Seg.)	Inter	Efecti.	MSE	T (Seg.)	Inter.	Efecti.
5	0.0093	86.3	10000	0.752	0.0616	6.7	456	0.203	0.0524	40.5	2855	0.221	0.0442	41.2	2902	0.374
7	0.0061	93.0	10000	0.743	0.0474	65.3	4616	0.311	0.0529	30.3	2188	0.203	0.0338	91.2	6266	0.387
9	0.0041	74.9	7841	0.766	0.0561	28.6	2055	0.171	0.0301	102.5	6668	0.441	0.0444	60.2	3936	0.356
11	0.0034	40.0	4241	0.694	0.0392	55.6	4236	0.410	0.0180	69.9	4675	0.599	0.0454	10.7	702	0.261
13	0.0034	29.0	2880	0.730	0.0334	103.5	7323	0.428	0.0227	52.0	3404	0.536	0.0228	24.8	1353	0.523
15	0.0031	30.5	3080	0.698	0.0252	75.5	5157	0.559	0.0442	10.2	628	0.333	0.0063	12.5	736	0.707
17	0.0027	17.7	1712	0.703	0.0261	81.5	5512	0.496	0.0184	13.6	807	0.613	0.0699	6.6	392	0.221
19	0.0022	17.3	1663	0.716	0.0198	87.3	5967	0.635	0.0217	11.1	622	0.550	0.0127	9.8	563	0.685
21	0.0019	15.6	1429	0.734	0.0367	87.3	5746	0.532	0.0234	8.3	461	0.491	0.0230	9.0	484	0.590
23	0.0017	14.8	1325	0.721	0.0227	24.0	1413	0.613	0.0078	9.9	530	0.667	0.0160	8.7	430	0.640
25	0.0018	12.9	1186	0.725	0.0450	18.5	1207	0.320	0.0382	6.1	330	0.374	0.0168	6.5	369	0.617
27	0.0014	12.0	1063	0.703	0.0279	28.2	1773	0.455	0.0276	9.0	444	0.496	0.0313	6.8	301	0.464
29	0.0022	13.7	1152	0.698	0.0576	23.7	1268	0.302	0.0491	7.4	363	0.473	0.0189	9.0	396	0.577
31	0.0016	11.6	947	0.770	0.0408	21.0	1186	0.369	0.0306	8.8	399	0.459	0.0431	7.4	327	0.432
33	0.0020	14.2	1137	0.739	0.0246	37.4	2036	0.527	0.0381	4.3	191	0.401	0.0212	6.9	303	0.509
35	0.0016	9.4	732	0.752	0.0424	33.0	1751	0.320	0.0088	6.2	276	0.689	0.0350	5.8	227	0.374
37	0.0011	8.2	632	0.725	0.0527	12.4	622	0.396	0.0158	6.6	283	0.608	0.0418	4.8	186	0.347
39	0.0013	6.7	504	0.725	0.0257	20.0	1123	0.518	0.0031	7.4	311	0.806	0.0292	7.8	308	0.477
41	0.0019	7.3	540	0.743	0.0341	23.0	1099	0.410	0.0123	9.0	379	0.685	0.0130	6.3	260	0.680
43	0.0011	10.6	766	0.685	0.0405	8.1	354	0.324	0.0013	6.8	292	0.766	0.0148	7.9	313	0.707
45	0.0011	7.4	521	0.725	0.0344	0.4	2811	0.428	0.0542	2.4	87	0.203	0.0222	9.3	343	0.559

RP : Resilente.  
CGF : Gradiente conjugado Fletcher-Reeves.  
CGP : Gradiente conjugado Polak-Ribière.  
CGB : Gradiente conjugado Powell-Beale.

**Tabla 6.10.**  
**Evaluación de los algoritmos SCG, OSS, LM**

Neuronas	SCG				BFGS				OSS				LM			
	MSE	T (Seg.)	Inter.	Efecti.	MSE	T (Seg.)	Inter.	Efecti.	MSE	T (Seg.)	Inter.	Efecti.	MSE	T (Seg.)	Inter.	Efecti.
5	0.0153	122.2	10000	0.667	0.0714	1.7	4	0.000	0.0632	87.5	5096	0.113	0.0527	96.2	467	0.405
7	0.0116	129.4	10000	0.595	0.0713	2.9	13	0.000	0.0423	99.7	6668	0.365	0.0117	26.0	88	0.644
9	0.0052	73.8	5539	0.685	0.0713	17.8	26	0.000	0.0650	64.2	3769	0.081	0.0110	37.7	100	0.527
11	0.0466	15.1	1113	0.455	0.0135	943.6	3683	0.662	0.0494	122.9	7066	0.221	0.0122	40.6	86	0.604
13	0.0078	18.6	1347	0.662	0.0460	56.2	231	0.162	0.0561	83.7	4986	0.185	0.0126	103.9	176	0.550
15	0.0142	10.0	690	0.644	0.0594	1544.4	563	0.270	0.0535	71.9	3833	0.225	0.0068	48.9	66	0.653
17	0.0091	11.1	754	0.680	0.0289	2004.7	1798	0.568	0.0284	82.4	4217	0.446	0.0152	93.6	100	0.563
19	0.0046	9.8	641	0.761	0.0590	546.2	432	0.135	0.0590	63.8	3335	0.117	0.0064	64.4	60	0.631
21	0.0214	10.0	614	0.568	0.0657	301.1	188	0.095	0.0469	95.8	4464	0.297	0.0040	189.4	149	0.671
23	0.0089	7.7	451	0.707	0.0364	1690.7	746	0.405	0.0298	149.4	7142	0.496	0.0116	112.1	54	0.703
25	0.0105	5.5	350	0.671	0.0643	370.8	109	0.054	0.0469	90.6	4050	0.284	0.0121	116.6	68	0.613
27	0.0319	7.7	456	0.428	0.0714	10.9	3	0.000	0.0338	151.8	6750	0.428	0.0028	76.8	39	0.716
29	0.0249	5.5	297	0.473	0.0487	2200.8	479	0.378	0.0398	95.5	4145	0.320	0.0309	140.0	49	0.432
31	0.0165	7.7	406	0.635	0.0610	831.6	149	0.059	0.0540	56.2	2304	0.180	0.0041	100.4	39	0.734
33	0.0141	9.8	504	0.608	0.0531	3329.5	478	0.143	0.0608	35.9	1431	0.095	0.0015	153.5	54	0.766
35	0.0400	5.8	289	0.536	0.0097	5225.5	640	0.322	0.0392	79.9	3195	0.333	0.0372	271.5	85	0.401
37	0.0207	7.1	354	0.577	0.0658	322.6	35	0.012	0.0350	105.7	4158	0.360	0.0070	273.8	78	0.608
39	0.0076	10.0	473	0.721	0.0654	1294.8	117	0.081	0.0426	105.5	4189	0.338	0.0360	320.5	47	0.460
41	0.0022	8.4	397	0.743	0.0575	18216.9	574	0.282	0.0412	102.0	3895	0.356	0.0102	278.2	66	0.680
43	0.0030	10.1	471	0.721	0.0323	10103.4	652	0.331	0.0310	110.6	4167	0.491	0.0218	311.9	61	0.586
45	0.0281	8.3	378	0.505	0.0501	5326.2	319	0.154	0.0384	56.8	2131	0.392	0.0042	291.0	57	0.694

SCG : Gradiente conjugado escalado.

BFGS : Cuasi-Newton.

OSS : Secante de un solo paso.

LM : Levenberg-Marquardt.

**Tabla 6.11.**  
**Evaluación de los algoritmos GDA, GDX, RP**

Neuronas	GDA				GDX				RP			
	MSE	T (Seg.)	Inter.	Efecti.	MSE	T (Seg.)	Inter.	Efecti.	MSE	T(Seg.)	Inter.	Efecti.
20	0.00158	92.9	8655	0.749	0.00137	45.1	4439	0.797	0.00167	11.9	1054	0.703
22	0.00151	86.6	8233	0.795	0.00145	39.0	3848	0.789	0.00189	12.3	1083	0.727
24	0.00140	85.7	8126	0.796	0.00272	42.6	4258	0.773	0.00208	11.3	995	0.716
26	0.00185	90.4	8540	0.785	0.00172	40.5	3986	0.757	0.00188	10.7	949	0.721
28	0.00132	94.2	8828	0.772	0.00137	36.0	3511	0.779	0.00182	11.3	1008	0.716
30	0.00190	83.4	7922	0.792	0.00158	41.7	4124	0.796	0.00220	11.1	979	0.754
32	0.00141	81.8	7678	0.799	0.00240	39.5	3892	0.775	0.00180	10.3	916	0.753
34	0.00141	86.6	8191	0.799	0.00119	42.8	4289	0.770	0.00187	10.7	937	0.743
36	0.00236	90.6	8613	0.778	0.00158	43.5	4341	0.785	0.00204	10.8	972	0.719
38	0.00128	85.6	8064	0.779	0.00198	47.0	4649	0.763	0.00149	12.2	1069	0.743
40	0.00163	92.3	8520	0.747	0.00137	44.5	4330	0.797	0.00141	11.0	961	0.701

GDA : Gradiente descendente con tasa de aprendizaje adaptativo.

GDX : Gradiente descendente con Momento y con tasa de aprendizaje adaptativo

RP : Resilente.



### 6.2.4.2. Selección de características

El objetivo de un procedimiento de selección de características es encontrar un subconjunto de características  $S_{m'}$  del conjunto inicial de características  $S_m$  con la finalidad de aumentar la eficiencia del clasificador, dado que se trata de reducir la dimensionalidad de los patrones de entrada  $S_m$ . Para que  $S_{m'}$  sea representativo se debe de construir eliminando las variables redundantes o las que no aportan suficiente información al clasificador. Con este objetivo se aplicará un algoritmo de selección de características, en donde se intenta probar que un subconjunto de las características iniciales puede minimizar el error de un clasificador.

Existen dos formas de resolver el problema, mediante algoritmos de búsqueda óptima que exploran el universo de todas las posibles combinaciones con gran esfuerzo computacional y mediante algoritmos de búsqueda subóptima que exploran algunas soluciones pero con un menor esfuerzo computacional:

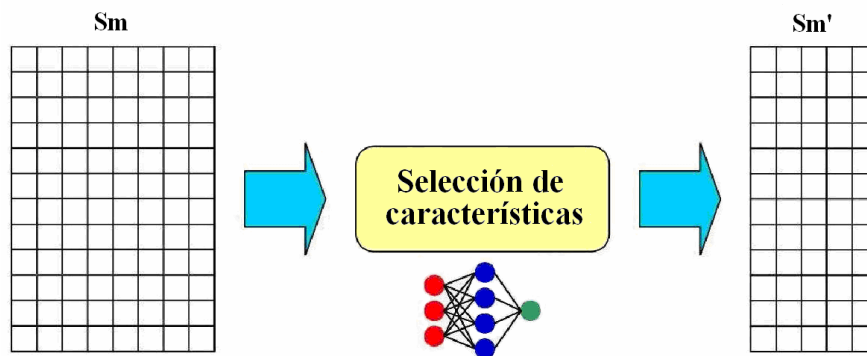


Fig. 6.31. Selección de Características

#### 1. Algoritmos de búsqueda óptima.

Entregan una solución óptima, evaluando todo el espacio de posibles combinaciones, pero a costo de un gran esfuerzo computacional. Si  $n$  es la cantidad de características identificadas y  $m$  es la cantidad de características deseadas, el número total de posibles subconjuntos a evaluar es:

$$\sum_m \binom{n}{m} = \sum_m \frac{n!}{m!(n-m)!} \quad (6.2)$$

Este número crece exponencialmente, haciendo casi imposible una búsqueda exhaustiva. Para calcular el tiempo total este número debe de multiplicar por el tiempo que toma cada modelo en aprender.

Por otro lado para un número fijo de muestras, aparece el efecto conocido como la maldición de la dimensionalidad en donde el error del clasificador se reduce al agregar nuevas características hasta lograr un mínimo, pero luego eventualmente el error se puede incrementar Hoekstra [87].

#### 2. Algoritmos de búsqueda subóptima.

Entrega una solución subóptima, que no necesariamente es la mejor pero a un menor costo computacional. Existe una serie de algoritmos que proporcionan un buen resultado, entre ellos tenemos:

- a. **Búsqueda secuencial hacia adelante.**  
 Parte de un conjunto vacío de características y en cada iteración añade una nueva característica (la mejor). El algoritmo hace una búsqueda entre las  $n$  características originales. La característica adicionada en cada iteración es aquella que cuando se considera reduce el error del clasificador. El algoritmo ejecuta:  $n + (n - 1) + (n - 2) + \dots + (n - m - 1) = n(n + 1)/2 - m(m + 1)/2$  evaluaciones en total.
- b. **Búsqueda secuencial hacia atrás.**  
 Parte de un conjunto formado por todas las características disponibles y en cada iteración elimina una característica (la peor). El algoritmo parte con una lista que contiene todas las características. En cada iteración elimina aquella que cuando no se considera reduce el error del clasificador. El algoritmo ejecuta el mismo número de evaluaciones que en la búsqueda secuencial hacia adelante

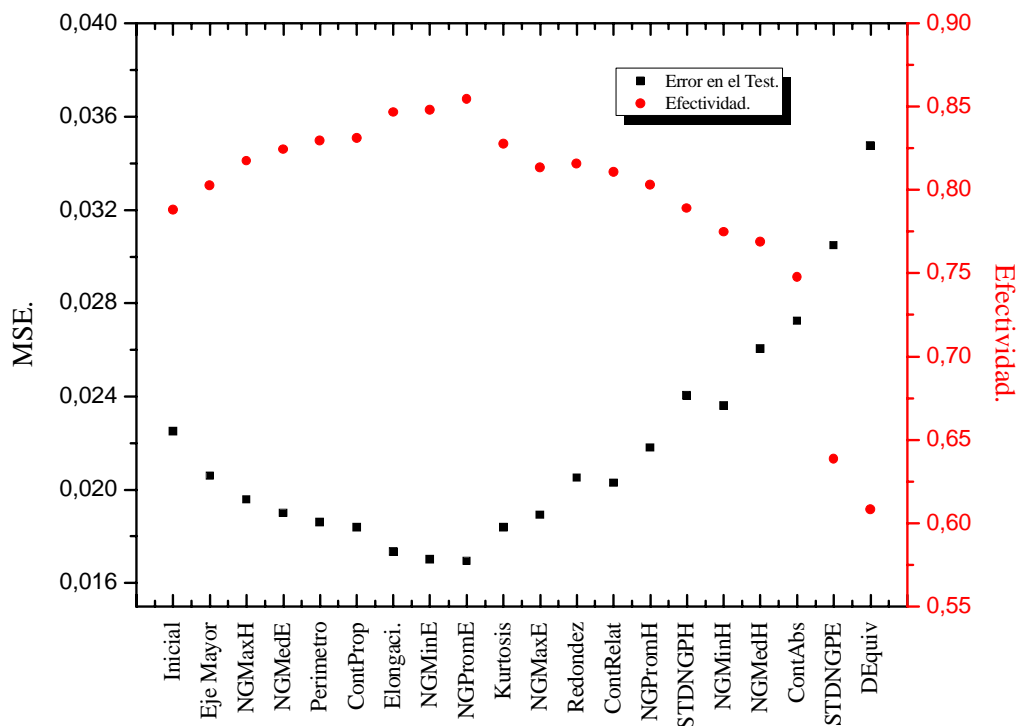


Fig. 6.32. Resultados del algoritmo de selección de características

El algoritmo usado en este trabajo es una derivación del algoritmo de búsqueda de secuencia hacia atrás, parte de la lista completa de 22 características y en cada iteración eliminamos aquella que cuando no se considera reduce el error o aumenta la eficiencia del clasificador. En la figura 6.32 se muestra el resultado de aplicar este algoritmo.

Inicialmente la eficiencia de clasificador es del 78.8%, usando la configuración de red neuronal indicado en la sección 6.2.4.1, es decir: 22 neuronas en la capa de entrada, 30 neuronas en la capa oculta, 14 neuronas en la capa de salida, la función de transferencia

usada en todos los nodos es la *logarítmica sigmoïdal* y el algoritmo de entrenamiento es el gradiente descendente con momento y con tasa de aprendizaje adaptativo.

La figura 6.32 muestra que conforme se va eliminando características como: eje mayor, nivel de gris máximo de la huella, nivel de gris mediano del entorno, perímetro, contraste proporcional, elongación, nivel de gris mínimo del entorno, nivel de gris promedio del entorno, la eficiencia de clasificación aumenta hasta alcanzar un máximo del 85.5%. Posteriormente conforme se va eliminando más características la eficiencia de clasificación comienza a disminuir, finalmente se muestra en la misma figura que después de haber eliminado 19 características la eficiencia del clasificador disminuye hasta el 60% con las últimas 3 características: Áreas, Eje menor, Skewness. Hay que mencionar que cada cuatro características eliminadas se hizo una revisión del número neuronas de la capa intermedia, modificando el número de neuronas en 26, 22, 15, 10 durante todo el algoritmo de selección de características.

Finalmente tenemos la siguiente configuración de RNA: 14 neuronas en la capa de entrada, 22 neuronas en la capa oculta, 14 neuronas en la capa de salida, la función de transferencia usada en todos los nodos es la *logarítmica sigmoïdal* y el algoritmo de entrenamiento es el gradiente descendente con momento y con tasa de aprendizaje adaptativo, es esta RNA la que se implementa en el modulo 2 del software.

### 6.2.4.3. Validación cruzada de K-conjuntos

Para estimar un indicador del desempeño de un clasificador o en su defecto el error de un clasificador, usaremos el estimador por validación cruzada de K conjuntos [88], usado frecuentemente para probar esquemas de clasificación.

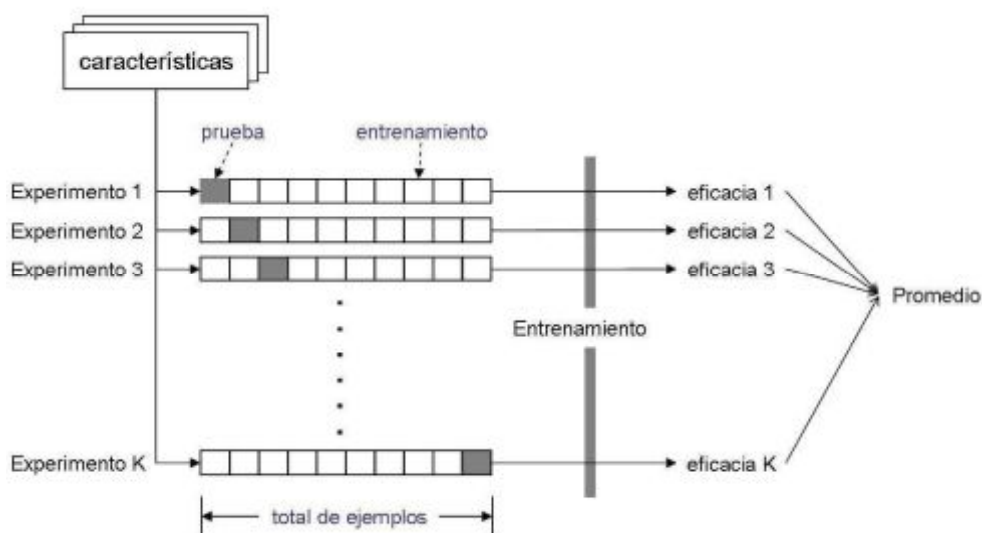


Figura 6.33. Estimador por validación cruzada con K conjuntos

El diseño consiste en armar  $K$  experimentos (ver la figura 6.33). En cada experimento los  $T$  datos de huellas, se dividen en  $K$  conjuntos disjuntos  $T_1, T_2, \dots, T_K$ , de tamaño similar:

$$|T_i| = |T|/K, \text{ donde } i = 1, 2, \dots, K \quad (6.3)$$

Para todo  $k = 1, 2, \dots, K$ , se construye un clasificador usando  $T - T_k$ , como conjunto de aprendizaje. Sea  $D_k$  el clasificador construido, donde ninguna de las muestras de  $T_k$  se ha usado para construir  $D_k$ .

Al finalizar este paso se obtiene  $K$  clasificadores,  $D_k$ , con sus correspondientes estimaciones del desempeño del proceso  $E_k$ .

La estimación final del desempeño del clasificador esta dado por:

$$E_T = \frac{1}{K} \sum_{i=1}^{i=K} E_i \quad (6.4)$$

El inconveniente de este diseño es el gran esfuerzo computacional que requiere, dado que es necesario volver a correr el experimento  $K$  veces, con los  $T$  datos, pero permite obtener una buena estimación del desempeño del clasificador.

En este trabajo para validar la red neuronal diseñada y dar un indicador del desempeño de un clasificador, se dividió la data total, consistente de 500 datos en 10 grupos de 50 y se llevo a cabo 5 experimentos. La tabla 6.12 muestra las distintas eficiencias obtenidas al aplicar la metodología de validación cruzada obteniéndose una eficiencia promedio del clasificador del 84 % con picos de hasta 96% en el mejor de los casos.

K-Conjuntos	Experimentos				
	Eficiencia 1	Eficiencia 2	Eficiencia 3	Eficiencia 4	Eficiencia 5
1	0.84	0.88	0.78	0.82	0.96
2	0.80	0.82	0.80	0.70	0.86
3	0.72	0.84	0.86	0.86	0.86
4	0.84	0.77	0.90	0.86	0.86
5	0.96	0.86	0.94	0.80	0.78
6	0.88	0.80	0.80	0.76	0.84
7	0.86	0.82	0.84	0.84	0.90
8	0.90	0.96	0.82	0.86	0.68
9	0.94	0.84	0.76	0.92	0.84
10	0.82	0.80	0.84	0.90	0.86
Promedio	$0.85 \pm 0.07$	$0.84 \pm 0.05$	$0.83 \pm 0.05$	$0.83 \pm 0.06$	$0.84 \pm 0.07$

Tabla 6.12. Eficiencias obtenidas al aplicar la metodología de validación cruzada.

## 7. Conclusiones

En este trabajo se desarrolló un software para el conteo, caracterización y clasificación de imágenes de huellas de partículas alfa de incidencia normal sobre el polímero detector CR-39. La metodología utilizada se basa en una combinación de detección y conteo de huellas a través del procesamiento digital de imágenes y clasificación de las mismas a partir de redes neuronales artificiales haciendo uso intensivo de las cajas de herramientas *Image Processing Toolbox* y *Neural Network Toolbox* de Matlab.

Se dotó a software de 2 módulos de operación: El primer módulo está orientado al conteo y determinación automática de los parámetros de las huellas, parámetros tales como: área, eje mayor, eje menor, excentricidad, opacidad, radio equivalente, etc., en total 22 parámetros pueden ser determinados con este módulo. El segundo módulo ofrece la capacidad de clasificar las huellas de acuerdo a la energía de las partículas alfa incidentes que las causan.

Se analizó vía procesamiento automático de imágenes los parámetros de imágenes de huellas nucleares alfa características con energías entre 0.15 y 7.8 MeV, con el fin de implementar el modelo propuesto que lleve a cabo de manera satisfactoria el reconocimiento de huellas alfa.

Para validar la red neuronal diseñada y dar un indicador del desempeño del clasificador, se utilizó el estimador por validación cruzada de K conjuntos; obteniéndose una eficiencia promedio del clasificador del 84 % con picos de hasta 96% en el mejor de los casos de huellas bien clasificadas.

Finalmente se desarrolló una interfaz gráfica de usuario amigable escrita en Java usando las librerías de componentes gráficos Swing y *SwingX* y que integra los algoritmos desarrollados en *Matlab*, gracias a la tecnología *Matlab Builder JA*, en una sola aplicación, proporcionando una manera cómoda e intuitiva de manejar el software. Además de permitir la elección del módulo con la cual desea trabajar así como el ingreso de los distintos parámetros, necesarios para las etapas de preprocesamiento y segmentación de las imágenes, el software permite el procesamiento de una o un grupo de imágenes, realizar un procesamiento supervisado o no supervisado de estas. Adicionalmente el software permite guardar, exportar los resultados a una hoja de cálculo *Excel* para su posterior tratamiento.

## 8. Bibliografía.

- [1] D.A. Young, *Etching of radiation damage in lithium fluoride*, Nature 182 (1958) 375-377.
- [2] R. L. Fleischer, *Tracks to Innovation: Nuclear Tracks in Science and Technology* (New York: Springer) (1998).
- [3] R. L. Fleischer, P. B. Price, and R. M. Walker, *Nuclear Tracks in Solids, Principles and Applications* (Berkeley, CA: University of California Press) (1975).
- [4] R. L. Fleischer and P. B. Price, *Charged Particle Tracks in Glass*, J. Appl. Phys. 34 (1963) 2903-2903.
- [5] R. L. Fleischer, P. B. Price, and R. M. Walker, *Ion Explosion Spike Mechanism for Formation of Charged Particle Tracks in Solids*, J. Appl. Phys. 36 (1965) 3645-3652.
- [6] S.A. Durrani, R.K. Bull, *Solid State Nuclear Track Detection. Principles Methods and Applications*, Pergamon Press, (1987).
- [7] R.L. Fleischer, P.B. Price, R.M. Walker, *Solid-state track detectors: Application to nuclear science and geophysics*, Ann. Rev. Nucl. Sci. 15 (1965) 1-28.
- [8] S.A. Durrani, R. Ilic, *Radon Measurements by Etched Track Detectors: Applications in Radiation Protection, Earth Sciences and the Environment*, World Scientific, Singapore, (1997).
- [9] R. Brandt, *Over Review of the Applications of SSNTD`s in Fission Physics*, Nucl. Instrum.Meth., 173 (1980) 147-153.
- [10] A. Chambaudet, et al., *Applications of Nuclear Tracks in Geosciences*, Radiation Measurements, 25 (1995) 487-494.
- [11] S. Manzoor, S. Balestra, M. Cozzi, M. Errico, G. Giacomelli, M.Giorgini, A.Kumar, A. Margiotta, E. Medinaceli, L. Patrizii, V. Popa, I. E. Qureshi and V. Togo, *Nuclear Track Detectors for Particle Searches*, Nuclear Physics B (Proc. Suppl.) 172 (2007) 296-298.
- [12] F. D'Errico, M. Weiss, M. Luszik-Bradra, M. Matzke, L. Bernardi and A. Cecchi, *A CR-39 Track Analyser for Neutron Spectrometry*. Radiat. Meas. 28 (1997) 823-830.
- [13] W. Henrich, et all., *High Energy Heavy ion Interactions Studied with SSNTD`s*, Radiation Measurements, 25 (1995) 203-218.
- [14] P. Meyer, J.E. Groetz, M. Fromm, A. Lacourt, A. Chambaudet, *Neutron dosimetry at low and high fluences with CR-39*, Radiat. Meas. 28 (1997) 423-428.
- [15] J. C. Hadler N., S. R., *Indoor Radon daughter contamination monitoring: the absolute efficiency of CR-39 taking into account the plateout effect and environmental conditions*, Ratiation Protection dosimetry, 51(1994) 283-296.
- [16] V. A. Nikolaev, R. Ilic. *Etched track radiometers in radon measurements: a review*, Radiation Measurements, 30 (1999) 1-13.
- [17] P.B. Price, R.L. Fleisher, D.D. Peterson, C. O`Ceallaigh, D. O`Sullivan and A. Thompson, *Identification of Isotopes of Energetic Particles with Dielectric Track Detectors*, Phys. Rev., 164 (1967) 618-1620.

- [18] G.H. Paretzke, E. Benton, P.R. Henke, *On particle track evolution in dielectric track detectors and charge identification through track radius measurement*, Nucl. Instr. Methods, 108 (1973) 73-80.
- [19] Yadav, J.S. *Charged particle identification using CR-39 detectors*. Radiat. Meas. 24 (1995) 115-128.
- [20] J. C. Hadler N., P.J. Iunes, A.M. Osorio A., S.R. Paulo, *Relationship between track size and energy for Alpha particles in CR-39*, Nucl. Tracks. Radiat. Meas., 19 (1991) 313-317.
- [21] J. C. Hadler N., P.J. Iunes, A. P. Policastro, S.R. Paulo, *Performance of alpha-Spectrometer CR-39 versus etching time and its possible application in Radon/Thoron Discrimination*, Journal of Radioanalytical and Nuclear chemistry, 182 (1994) 139-147.
- [22] J. C. Hadler N., P.J. Iunes, S. R. Paulo, *An Automated System for Track Measurement for the employment of CR-39 as alpha spectrometer in radon and its daughters monitoring*, of Radioanalytical and Nuclear chemistry, 182 (1994) 133-138.
- [23] Bondarenko, O. A., Salmon, P. L., Henshaw, D. L. and Fewes, A. P. *Performance of Alpha Particle Spectroscopy using a Tasrak Detector*. Radiat. Meas. 26 (1996) 59–64.
- [24] A. H. Khayrat, S. A. Durrani, *Variation of alpha-particle track diameter in CR-39 as a function of residual energy and etching conditions*, Radiation Measurements 30 (1999) 15-18.
- [25] M. IZERROUKEN, J. SKVAR, R. ILI, *Low energy Alpha particle spectroscopy using CR-39*, Radiation Measurements 31(1999) 141-144.
- [26] C. Amero, J.I. Golzarri, M. Izerrouken, G. Espinosa,  $^{148}\text{Gd}$ ,  $^{238}\text{U}$ ,  $^{239}\text{Pu}$  and  $^{244}\text{Cm}$  *Alpha particle energy analysis using tracks in solids*, Radiation Measurements, 34 (2001) 341–343.
- [27] G. Espinosa, C. Amero and R. B. Gammage, *Measurements of alpha particle energy using nuclear tracks in solids methodology*, Radiation Protection Dosimetry, 101 (2002) 561–564.
- [28] E.M. Awad, A.A. Soliman, Y.S. Rammah, *Alpha particle spectroscopy for CR-39 detector utilizing matrix of energy equations*, Physics Letters A, 369 (2007) 359–366.
- [29] Yu, K.N., Lee, H.H.W., Wong, A.W.T., Law, Y.L., Cheung, S.F.L., Nikezic, D., Ng, F.M.F. *Optical appearance of alpha-particle tracks in CR-39 SSNTD*. Nucl. Instrum. Methods B 263 (2007) 271–278.
- [30] Y.L. Law, D. Nikezic, K.N. Yu, *Optical appearance of alpha-particle tracks in CR-39 SSNTDs*, Radiation Measurements, 43 (2008) S128–S131.
- [31] B. G. Cartwright, and E. K. Shirk, *A Nuclear-Track-Recording Polymer of Unique Sensitivity and Resolution*, Nucl. Instr. and Meth., 153 (1978) 457-460.
- [32] R. M., Cassou and E. V. Benton, *Properties and Applications of CR-39 Polymeric Nuclear Track Detector*, Nucl. Track Detection, 2 (1978) 173-179.

- [33] P.F. Green, A.G. Ramli, S.A.R. Al-Najjar, F. Abu-Jarad, and S.A. Durrani, *A Study of Bulk-Etch Rates and Track-Etch Rates in CR-39*, Nucl. Instr. And Meth., 203 (1982) 551-559.
- [34] Espinosa, G., Gammage, R. B., Meyer, K. E. and Dudney, S. C. *Nuclear Track Analysis by Digital Image*. Radiat. Prot. Dosim. 66 (1996) 363–366.
- [35] Gammage, R. B. and Espinosa, G. *Digital Imaging System for Track Measurements*. Radiat. Meas. 28 (1997) 835–838.
- [36] A.A.R. Da Silva, E.M.Yoshimura, *Track analysis system for application in alpha particle detection with plastic detectors*, Radiation Measurements, 39 (2005) 621-625.
- [37] D.L. Patiris, K. Blekas , K.G. Ioannides, *TRIAC: A code for track measurements using image analysis tools*, Nuclear Instruments and Methods in Physics Research B 244 (2006) 392–396.
- [38] D.L. Patiris, K. Blekas, K.G. Ioannides, *TRIAC II. A MatLab code for track measurements from SSNT detectors*, Computer Physics Communications 177 (2007) 329–338.
- [39] F. Hernández, S. Gonzalez-Manrique, L. Karlsson, J. Hernández-Armas, A. Aparicio, *Counting radon tracks in Makrofol detectors with the “image reduction and analysis facility” (IRAF) software package*, Radiation Measurements, 42 (2007) 496 – 504.
- [40] Fuminobu SATO, Takahiro KUCHIMARU, Yushi KATO, and Toshiyuki IIDA, *Digital Image Analysis of Etch Pit Formation in CR-39 Track Detector*, Japanese Journal of Applied Physics, 47 (2008) 269–272.
- [41] D. Nikezic, K.N. Yu, *Computer program TRACK\_TEST for calculating parameters and plotting profiles for etch pits in nuclear track materials*, Computer Physics Communications, 174 (2006) 160–165.
- [42] D. Nikezic, K.N. Yu, *Computer program TRACK\_VISION for simulating optical appearance of etched tracks in CR-39 nuclear track detectors*, Comput. Phys. Commun, 178 (2008) 591-595.
- [43] Glenn F. Knoll, *Radiation Detection and Measurement*, Second Edition, John Wiley & Sons (1989).
- [44] Bloch F., *Zur Bremsung rasch bewegter Teilchen beim Durchgang durch Materie*, Ann. Phys. (Leipzig), 5 (1933) 285-321
- [45] H.A. Bethe, J. Askin, *Passage of radiations through matter*, Experimental Nuclear Physics, vol. I, Wiley, New York, (1953) 176.
- [46] J. Lindhard, M. Scharff and H. E. Schiott, *Range concepts and heavy ion ranges*, Mat. Fys. Medd. Dan. 33 (1963) 1-42.
- [47] S. P. Ahlen, *Theoretical and experimental aspects of the energy loss of relativistic heavily ionizing particles*; Rev. Mod. Phys. 52 (1980) 121-173.
- [48] P.R. Henke, E. Benton, *On the geometry of tracks in dielectric nuclear track detectors*, Nucl. Instr. Methods, 97 (1971) 483-489.
- [49] G. Somogyi, A.S. Szalay, *Track diameter kinetics in dielectric track detectors*, Nucl. Instr. Methods, 109 (1973) 211-232.



- [50] G. Somogyi, *Development of etched nuclear tracks*, Nucl. Instr. Methods, 173 (1980) 21-42.
- [51] D. Nikezic, D. Kostic, *Simulation of the track growth and determination the track parameters*, Radiat. Meas. 28 (1997) 185-190.
- [52] D. Nikezic, *Three-dimensional analytical determination of the track parameters*, Radiat. Meas. 32 (2000) 277-282.
- [53] D. Nikezic, K.N. Yu, *Three-dimensional analytical determination of the track parameters. Over-etched tracks*, Radiat. Meas. 37 (2003) 39-45.
- [54] D. Nikezic, D. Kostic, K.N. Yu, *Comparison among different models of track growth and experimental data*, Radiat. Meas. 41 (2006) 253-256.
- [55] D. Nikezic, K.N. Yu, *Formation and growth of tracks in nuclear track materials*, Mater. Sci. Eng. R46 (2004) 51-123.
- [56] Rafael C. González y Richard E. Woods, *Tratamiento Digital de Imágenes*, Addison-Wesley, (1996).
- [57] T. Masters, *Advanced algorithms for neural networks a C++ sourcebook*, John Wiley & Sons, INC. (1993).
- [58] Bishop, M. Christopher, *Neural networks for pattern recognition*, Oxford. Clarendon Press, (1990).
- [59] S. Haykin, *Neural networks. A comprehensive Foundation*, Macmillam College Publishing Campany, Inc. (1994).
- [60] Howard Demuth, Mark Beale, Martin Hagan, *Neural Network Toolbox. User's Guide*, (2010). MathWorks, Inc.
- [61] Pao, Yoh-Han, *Adaptive Pattern Recognition and neural networks*, (1989). Addison-Wesley Publishing Company, Inc.
- [62] Hagan, T. Martin, Demuth, B. Howard, M. Beale, *Neural network design*. (1996). PWS Publishing Company, a division of International Thomson Publishing Inc.
- [63] J. Hilera, V. Martinez, *Redes Neuronales Artificiales: Fundamentos, modelos y aplicaciones*, Addison-Wesley, (2000), Iberoamericana. Buenos Aires. Argentina.
- [64] Agustin Zuñiga, *Caracterização do CR-39 como Espectrometro Alfa*, Tesis para optar el grado de Doctor en física de la Universidad Estadual de Campinas, (1998).
- [65] Yagoda, H. *Radioactive Measurements with Nuclear Emulsions*, Jhon Wiley, N.Y, (1949).
- [66] Lederer C. M and V. S. Shirley, *"Table of Isotopes"*, Seven Ed., Jhon Wiley., N.Y, (1978).
- [67] J.F. Ziegler, J.P. Biersack, U. Littmark, *The Stopping and Ranges of Ions in Solids*, Pergamon Press, Oxford, (1985).
- [68] <http://www.srim.org/>
- [69] N. Otsu, *A Threshold Selection Method from Gray-Level Histogram*, IEEE Trans. Systems Man, and Cybernetics, 9 (1979) 62-66.
- [70] Rafael C. Gonzalez, Richard E.Woods & Steven L. Eddins, *Digital Image Processing Using Matlab*, Gatesmark Publishing, (2009).

- [71] L. da Fontoura and R. Marcondes. *Shape Analysis and Classification. Theory and Practice*. CRC Press, (2001).
- [72] <http://www.mathworks.com/>
- [73] *Image Processing Toolbox User's Guide*, (2010). MathWorks, Inc.
- [74] *Matlab Compiler 4 User's Guide*, (2010). MathWorks, Inc.
- [75] *Matlab Builder JA 2 User's Guide*, (2010). MathWorks, Inc.
- [76] <http://www.java.com/es/>
- [77] <http://es.sun.com/>
- [78] <http://www.oracle.com/technetwork/java/javase/downloads/index.html>
- [79] <http://netbeans.org/>
- [80] <https://jai.dev.java.net/>
- [81] [http://java.sun.com/products/java-media/jai/downloads/download-1\\_1\\_2.html](http://java.sun.com/products/java-media/jai/downloads/download-1_1_2.html)
- [82] <http://www.jfree.org/jfreechart>
- [83] <http://download.oracle.com/javase/tutorial/uiswing/index.html>
- [84] <http://www.swinglabs.org/>
- [85] <http://jexcelapi.sourceforge.net/>
- [86] K. Kurkova. Kolmogorov's Theorem. *The Handbook of Brain Theory and Neural Networks*,. MIT Press, (1995) 501-502
- [87] A. Hoekstra. *Generalisation in feed forward neural classifiers*, Master's thesis, Technische Universiteit Delft, Octobre (1998).
- [88] M. Stone, *Cross-validation choice and assessment of statistical predictions*, (1993).

## 9. Anexos

### 9.1. Anexo 1.

#### Medidas de separación fuente-detector.

**Tabla 9.1**

Medición de la separación entre placas en el caso de fuentes de Uranio.

Medición del Parámetro, d(mm)

incidencia: normal

E(MeV)	cod	n	d1	d2	d3	d4	dt(mm)	sig(m)	d(mm)	sig(mm)
4.39	X10	2	1.68	1.66	1.66	1.66	1.67	0.01	1.07	0.01
4.22	2X12	4	3.30	3.24	3.28	3.22	3.26	0.04	2.66	0.04
3.95	2X13	7	5.70	5.76	5.74	5.72	5.73	0.03	5.13	0.03
3.57	2X14	11	9.14	9.00	9.02	9.08	9.06	0.06	8.46	0.06
3.27	2X15	14	11.54	11.46	11.48	11.50	1.50	0.03	10.90	0.03
2.84	2X16	18	14.76	14.72	14.66	14.80	14.74	0.06	14.14	0.06
2.50	2X17	21	17.18	17.32	17.16	17.18	17.21	0.07	16.61	0.07
2.23	2X18	23	18.88	18.86	18.94	18.92	18.90	0.04	18.30	0.04
1.95	2X19	25	20.50	20.46	20.68	20.54	20.55	0.10	19.95	0.10
0.80	2X20	32	26.08	26.36	26.22	26.24	26.23	0.11	25.63	0.11

**E** : Energía de referencia para construir el montaje.

**cód** : Código del CR-39.

**n** : Número de arandelas de separación ( $e=0.822\pm 0.0001$  mm)

**dt** : Distancia total de separación entre placas.

**d** : Distancia de separación entre placas descontando el espesor del CR-39 y la fuente.

**Tabla 9.2**

Medición de la separación entre placas en el caso de Fuentes de Torio

Medición del Parámetro, d(mm)

Incidencia: normal

T=12C

E(MeV)	cod	N	d1	d2	d3	d4	dt(mm)	sig	d(mm)	sig
7.99	T19	11	12.16	12.22	12.24	12.20	12.21	0.03	11.61	0.03
6.94	T20	24	26.38	26.34	26.30	26.26	26.32	0.05	25.72	0.05
5.88	T21	35	39.66	39.58	39.64	39.68	39.64	0.04	39.04	0.04
4.82	T22	46	50.96	50.80	51.18	51.10	51.01	0.17	50.41	0.17

**E** : Energía de referencia para construir el montaje.

**cód** : Código del CR-39.

**n** : Número de arandelas de separación ( $e=1.11\pm 0.006$  mm)

**dt** : Distancia total de separación entre placas.

**d** : Distancia de separación entre placas descontando el espesor del CR-39 y la fuente.

## 9.2. Anexo 2.

### 9.2.1. Código Matlab® de la función principal utilizada para el conteo y clasificación de imágenes de huellas nucleares alfa.

Archivo: ProcesImageFinal.m

```
function[ExtraCaracteristicas,NumHuellas,PerimetroHuellas,Centroides,DataResultClasifica]=ProcesImagenFinal(RutaImagen,CalivPixelX,CalivPixelY,RadioFitroPromedio,RadioFitroGauss,SigmaGauss,STR,SobelFactor,UmbraGlobal,Umbra16Win,ClearBorder,AreaMinima,AreaMaxima,SeparHuellas,RestOpacidad,RestRedondez,RestExentric,IsClasificacion)
```

```
% Parámetros de ingreso:
% RutaImagen = Ruta Absoluta de la imagen a Tratar.
% CalivPixelX = Calibración en X.
% CalivPixelY = Calibración en Y
% RadioFitroMediana = Radio del filtro de de la mediana usado.
% RadioFitroGauss = Radio del filtro de Gauss.
% SigmaGauss = Sigma del filtro de Gauss.
% STR = Tamaño del elemento estructurante.
% SobelFactor = Factor de Sobel.
% UmbraGlobal = Variable lógica que determina si se aplica umbralización
% global o no.
% Umbra16Win = Variable lógica que determina si se aplica umbralización
% ponderada con 16 ventanas.
% ClearBorder = Variable lógica que determina si se elimina las huellas
% encontradas en la frontera de la imagen.
% AreaMinima = Área mínima considerada como huella.
% AreaMaxima = Área máxima considerada como huella.
% SeparHuellas = Variable lógica que determina si se aplica el algoritmo
% de separación de huellas desarrollado.
% RestOpacidad = Parámetro de restricción para la opacidad.
% RestRedondez = Parámetro de restricción para la redondez
% RestExentric = Parámetro de restricción para la elongación.
% IsClasificacion = Variable lógica que determina si se aplica el
% algoritmo clasificación de huellas desarrollado.
```

```
% Parámetros de salida:
% ExtraCaracteristicas = Estructura que almacena las características de
% las huellas encontradas en la imagen.
% NumerObjetos = Número de huellas alfa encontradas.
% PerimetroHuellas = Perímetro de las huellas alfa encontradas.
% Centroides = Centroides de las huellas alfa encontradas.
% DataResultClasifica = Matriz que contiene los resultados de la
% clasificación de las huellas alfa encontradas en la imagen
```

```
PixelArea = CalivPixelX.*CalivPixelY;
DataResultClasifica = [];
DataForClasificar = [];
```

```
Centroides = [];
ExtraCaracteristicas.Areas=[];
ExtraCaracteristicas.EjeMayor=[];
ExtraCaracteristicas.EjeMenor=[];
ExtraCaracteristicas.Exentricidad=[];
ExtraCaracteristicas.Perimetro=[];
ExtraCaracteristicas.DiametroEquivalent=[];
```

```

ExtraCaracteristicas.Redondez=[];
ExtraCaracteristicas.NivGrizMaxBackgr=[];
ExtraCaracteristicas.NivGrizMinBackgr=[];
ExtraCaracteristicas.MedianaGrizBackgr=[];
ExtraCaracteristicas.PromedioGrizBackgr=[];
ExtraCaracteristicas.StdGrizBackgr=[];
ExtraCaracteristicas.NivGrizMaxForeg=[];
ExtraCaracteristicas.NivGrizMinForeg=[];
ExtraCaracteristicas.MedianaGrizForeg=[];
ExtraCaracteristicas.PromedioGrizForeg=[];
ExtraCaracteristicas.StdGrizForeg=[];
ExtraCaracteristicas.ContrasAbsolut=[];
ExtraCaracteristicas.ContrasRelat=[];
ExtraCaracteristicas.ContrasPropor=[];
ExtraCaracteristicas.Opacidad=[];

% Pasos del procesamiento de la imagen
% IOriginal = Imagen de entrada
% Inicio del proceso

% Lectura de la imagen
% Trasformando a una imagen en escala de grises si la imagen de entrada
% una imagen a color.

IOriginal=imread(RutaImagen);
Dimensiones=size(IOriginal);
K=length(Dimensiones);
if K==3
    IOriginal=rgb2gray(IOriginal);
end

[M,N]=size(IOriginal);

% Preprocesamiento de la imagen
% Aplicando Filtros a la imagen
% Usando el Filtro de la mediana

A = medfilt2(IOriginal,[RadioFitroMediana RadioFitroMediana]);

% Usando el Filtro de la Gauss

h = fspecial('gaussian', RadioFitroGauss,SigmaGauss);
A=imfilter(A,h);

% Mejorando la calidad de la imagen
% Mejorando el contraste

AContras=imadjust(A,stretchlim(A),[0 1]);
UnbralContras=graythresh(A);

% Segmentando La Imagen

se = strel('disk',STR);
A=imsubtract(imadd(AContras,imtophat(AContras,se)),imbothat(AContras,se))
;

% Segmentando la imagen, usando umbral global

Unbral=graythresh(A);
if(UmbralGlobal)

```

```

    A=im2bw(A,Unbral);
    BWcomplemen = imcomplement(A);
end

% Segmentando la imagen, usando umbralización local en 16 ventanas

if (Umbral16Win)
    Partes=cell(16,1);
    DivCel=DivImag(A);
    i=1;
    for K=1:4
        Div=DivCel{K};
        DivTemp=DivImag(Div);
        for m=1:4
            Partes{i}=DivTemp{m};
            i=i+1;
        end
    end
    Q=cell(16,1);
    UnbralVentana = zeros(16,1);
    for i=1:16
        Q1=Partes{i};
        UnbralVentana(i)=graythresh(Partes{i});
        Umbr=(Unbral+UnbralVentana(i))/2;
        Q2=im2bw(Q1,Umbr);
        Q{i}=Q2;
    end
    F=cell(4,1);
    k=1;
    for i=1:4:16
        F{k}=JuntImagen(Q{i},Q{i+1},Q{i+2},Q{i+3});
        k=k+1;
    end
    A=JuntImagen(F{1},F{2},F{3},F{4});
    BWcomplemen = imcomplement(A);
end

% Determinando los bordes de los objetos

BW1 = edge(AContras,'sobel',(UnbralContras*SobelFactor));
BW1Fill =imdilate(BW1,strel('disk',1));
BWoffbord1 = imclearborder(BW1Fill,4);

BW1Fill = imfill(BWoffbord1,'holes');

% Sumando los bordes encontrados con la imagen obtenida por
% segmentación usando ya sea umbralización global o ponderada.

BW1=im2uint8(BW1);
BWcomplemen=im2uint8(BWcomplemen);
BW1Fill=im2uint8(BW1Fill);
Suma=imadd(BW1Fill,BWcomplemen);

% Eliminando los elementos del borde de la imagen

BWoffbord = Suma;
ClearBorder = 1;
if (ClearBorder)
    BWoffbord = imclearborder(Suma, 4);
end

```

```

% Dilatando la imagen Con un elemento estructurante en forma de disco

ElemenEstruct1= strel('disk',1);
BWdilt=imdilate(BWoffbord,ElemenEstruct1);

% Rellenando la imagen

BWfill= imfill(BWdilt,'holes');

% Aplicando la operación morfológica Opening

BWopen = imopen(BWfill,se);

% Aplicando la operación morfológica erosión

BWerosi = imerode(BWopen ,ElemenEstruct1);
IBWFINAL=BWerosi;

[L,NumerObjetos] = bwlabel(IBWFINAL,4);
if (NumerObjetos ~=0)

% Eliminando objetos segmentados de muy pequeño tamaño que puede ser
% basura
% Etiquetando Regiones Conectadas

    S=regionprops(L,'Area');
    AREAS = [S.Area];
    Media=mean(AREAS);
    P=abs(Media/2.5);
    IBWFINAL = ismember(L, find(AREAS >= P));

    [L,NumerObjetos] = bwlabel(IBWFINAL,4);

% Eliminando objetos segmentados tomando un nivel de Gris Umbral
if NumerObjetos ~=0
    ResultaEstadisticos=regionprops(L,'all');
    [Opacid]= Opacidad(IOriginal,NumerObjetos,ResultaEstadisticos);

    % OpacMedia=mean(Opacid);
    % OpacStandDesv=std(Opacid);
    % Opa=abs(OpacMedia+OpacStandDesv);
% Parámetro de Restricción en la opacidad de Referencia:
% RestOpacidad=0.45

    IBWFINAL = ismember(L, find(Opacid <= RestOpacidad));
    [L,NumerObjetos] = bwlabel(IBWFINAL,4);
end
end

% Restricción en el area

if (NumerObjetos ~=0)
    S=regionprops(L,'Area');
    AREAS = [S.Area].*PixelArea;
    IBWFINAL = ismember(L, find(AREAS>AreaMinima));
    [L,NumerObjetos] = bwlabel(IBWFINAL,4);
    if (NumerObjetos ~=0)
        S=regionprops(L,'Area');
    end
end

```

```

        AREAS = [S.Area].*PixelArea;
        IBWFINAL = ismember(L, find(AREAS<AreaMaxima));
    end
    [L, NumerObjetos] = bwlabel(IBWFINAL,4);
end

if ((NumerObjetos ~=0) && SeparHuellas)
    stats=regionprops(L, 'all');

    % Algoritmo para la separación de huellas alfa unidas

    area=[stats.Area];
    P1=abs(mean(area));
    for i=1:NumerObjetos
        if area(i)>(P1*1.5)
            Temp=stats(i).BoundingBox;
            EsquiSuperX=Temp(1)-10.5;
            EsquiSuperY=Temp(2)-10.5;
            if EsquiSuperX<=0
                EsquiSuperX=1;
            end
            if EsquiSuperY<=0
                EsquiSuperY=1;
            end
            DistX=Temp(3)+15; % Ancho
            DistY=Temp(4)+15; % Alto
            if DistX>N
                DistX=N;
            end
            if DistY>M
                DistY=M;
            end
            PartImagFordividirI=zeros(DistY,DistX);
            for m=1:DistY
                for n=1:DistX
                    PartImagFordividirI(m,n)=IBWFINAL(EsquiSuperY+m,EsquiSuperX+n);
                end
            end
            L=WaterFinal(PartImagFordividirI);
            for m=1:DistY
                for n=1:DistX
                    IBWFINAL(EsquiSuperY+m,EsquiSuperX+n)=L(m,n);
                end
            end
        end
    end
end

% Eliminando objetos de menos de 20 píxeles que puedan haber quedado
% después de watershed

IBWFINAL = bwareaopen(IBWFINAL,20,4);
[L,NumerObjetos] = bwlabel(IBWFINAL,4);
Stats=regionprops(L, 'all');
if NumerObjetos ~=0
    Struc = diameter(L);
    for i=1:NumerObjetos

```



```

        MajorAxis(i) = sqrt(((Struc(i).MajorAxis(1,1)-
Struc(i).MajorAxis(2,1))*CalivPixelX)^2+((Struc(i).MajorAxis(1,2)-
Struc(i).MajorAxis(2,2))*CalivPixelY)^2);
        MinorAxis(i) = sqrt(((Struc(i).MinorAxis(1,1)-
Struc(i).MinorAxis(2,1))*CalivPixelX)^2+((Struc(i).MinorAxis(1,2)-
Struc(i).MinorAxis(2,2))*CalivPixelY)^2);
        Exentric(i) = MinorAxis(i)/MajorAxis(i);
    end
    Areas = [Stats.Area].*PixelArea;
    PerimetroHuellas = bwboundaries(IBWFINAL);
    Perimet = Perimetro(PerimetroHuellas,CalivPixelX,CalivPixelY);
    Redondez=RedondezII(NumerObjetos,Areas,Perimet);

% Parámetro de Restricción en la Redondez de Referencia:
% RestRedondez = 0.9

    IBWFINAL = ismember(L, find(Redondez > RestRedondez));
    [L, NumerObjetos] = bwlabel(IBWFINAL,4);
    if NumerObjetos ~=0

% Parámetro de Restricción en la elongación de Referencia:
% RestExentric = 0.7

        Struc = diameter(L);
        for i=1:NumerObjetos
            MajorAxis(i) = sqrt(((Struc(i).MajorAxis(1,1)-
Struc(i).MajorAxis(2,1))*CalivPixelX)^2+((Struc(i).MajorAxis(1,2)-
Struc(i).MajorAxis(2,2))*CalivPixelY)^2);
            MinorAxis(i) = sqrt(((Struc(i).MinorAxis(1,1)-
Struc(i).MinorAxis(2,1))*CalivPixelX)^2+((Struc(i).MinorAxis(1,2)-
Struc(i).MinorAxis(2,2))*CalivPixelY)^2);
            Exentric(i) = MinorAxis(i)/MajorAxis(i);
        end
        Exentricidad = Exentric;
        IBWFINAL = ismember(L, find(Exentricidad >RestExentric));
    end
    [L, NumerObjetos] = bwlabel(IBWFINAL,4);
end

% Creando la estructura de salida con las características de las huellas
% alfa encontradas

IEtiqueta = L;
NumHuellas = NumerObjetos;
Stats = regionprops(L, 'all');
if (NumHuellas ~= 0)

    [NivGrizMaxBackgr, NivGrizMinBackgr, MedianaGrizBackgr,
PromedioGrizBackgr, StdGrizBackgr
]=CaracterBackgr(IOriginal, IBWFINAL, NumHuellas, Stats);
    [NivGrizMaxForeg, NivGrizMinForeg, MedianaGrizForeg,
PromedioGrizForeg, StdGrizForeg
]=CaracterForeg(IOriginal, NumHuellas, Stats);
    [ContrasAbsolut, ContrasRelat, ContrasPropor] =
ContrasRelativ(PromedioGrizForeg, PromedioGrizBackgr, NumHuellas);
    [KurtosisHuella,
SkewnessHuella]=KurtandSkewForeg(IOriginal, NumHuellas, Stats);
    [Opacid]= Opacidad(IOriginal, NumHuellas, Stats);
    PerimetroHuellas = bwboundaries(IBWFINAL);
    Centroides = cat(1, Stats.Centroid);
end

```

```

Struc = diameter(IEtiqueta);
for i=1:NumHuellas
    MajorAxis(i) = sqrt(((Struc(i).MajorAxis(1,1)-
Struc(i).MajorAxis(2,1))*CalivPixelX)^2+((Struc(i).MajorAxis(1,2)-
Struc(i).MajorAxis(2,2))*CalivPixelY)^2);
    MinorAxis(i) = sqrt(((Struc(i).MinorAxis(1,1)-
Struc(i).MinorAxis(2,1))*CalivPixelX)^2+((Struc(i).MinorAxis(1,2)-
Struc(i).MinorAxis(2,2))*CalivPixelY)^2);
    Exentric(i) = MinorAxis(i)/MajorAxis(i);
end
Perimeter = Perimetro(PerimetroHuellas,CalivPixelX,CalivPixelY);
Areas = [Stats.Area].*PixelArea;
Redondez=RedondezII(NumerObjetos,Areas,Perimeter);

for i=1:NumHuellas
    ExtraCaracteristicas(i,1).Areas=(Stats(i).Area)*PixelArea;
    ExtraCaracteristicas(i,1).EjeMayor=MajorAxis(i);
    ExtraCaracteristicas(i,1).EjeMenor=MinorAxis(i);
    ExtraCaracteristicas(i,1).Exentricidad=Exentric(i);
    ExtraCaracteristicas(i,1).Perimetro=Perimeter(i);

ExtraCaracteristicas(i,1).DiametroEquivalent=sqrt(4*(Stats(i).Area)*Pixel
Area/pi);
    ExtraCaracteristicas(i,1).Redondez=Redondez(i);
    ExtraCaracteristicas(i,1).NivGrizMaxBackgr=NivGrizMaxBackgr(i);
    ExtraCaracteristicas(i,1).NivGrizMinBackgr=NivGrizMinBackgr(i);

ExtraCaracteristicas(i,1).MedianaGrizBackgr=MedianaGrizBackgr(i);

ExtraCaracteristicas(i,1).PromedioGrizBackgr=PromedioGrizBackgr(i);
    ExtraCaracteristicas(i,1).StdGrizBackgr=StdGrizBackgr(i);
    ExtraCaracteristicas(i,1).NivGrizMaxForeg=NivGrizMaxForeg(i);
    ExtraCaracteristicas(i,1).NivGrizMinForeg=NivGrizMinForeg(i);
    ExtraCaracteristicas(i,1).MedianaGrizForeg= MedianaGrizForeg(i);

ExtraCaracteristicas(i,1).PromedioGrizForeg=PromedioGrizForeg(i);
    ExtraCaracteristicas(i,1).StdGrizForeg=StdGrizForeg(i);
    ExtraCaracteristicas(i,1).ContrasAbsolut=ContrasAbsolut(i);
    ExtraCaracteristicas(i,1).ContrasRelat=ContrasRelat(i);
    ExtraCaracteristicas(i,1).ContrasPropor=ContrasPropor(i);
    ExtraCaracteristicas(i,1).Opacidad=Opacid(i);
    ExtraCaracteristicas(i,1).Kurtosis = KurtosisHuella(i);
    ExtraCaracteristicas(i,1).Skewness = SkewnessHuella(i);
end

if (IsClasificacion)

% Características seleccionadas durante el proceso de selección de
% características

for i=1:NumHuellas
    Area(i) = (Stats(i).Area)*PixelArea;
    EjeMenor(i) = MinorAxis(i);
    DiametroEquiva(i) = sqrt(4*(Stats(i).Area)*PixelArea/pi);
    Redon(i) = Redondez(i);
    NivGrizMaxBack(i) = NivGrizMaxBackgr(i);
    StdGrizBack(i) = StdGrizBackgr(i);
    NivGrizMinFor(i) = NivGrizMinForeg(i);
    MedianaGrizFor(i) = MedianaGrizForeg(i);
    PromedioGrizFor(i) = PromedioGrizForeg(i);
    StdGrizFore(i) = StdGrizForeg(i);

```

```

ContrasAbsolu(i) = ContrasAbsolut(i);
ContrasRelat(i) = ContrasRelat(i);
Kurtosis(i) = KurtosisHuella(i);
Skewness(i) = SkewnessHuella(i);
end

```

```

DataForClasificar =
[Area',EjeMenor',DiametroEquiva',Redon',NivGrizMaxBack',StdGrizBack',NivG
rizMinFor',MedianaGrizFor',PromedioGrizFor',StdGrizFore',ContrasAbsolu',C
ontrasRelat',Kurtosis',Skewness'];

```

```

% Parámetros de la red diseñada
% Matriz de pesos entre la capa de entrada y la capa oculta.

```

```

IW1 = [-1.7118 -2.7745 -2.5356 -1.4884 0.47738 1.6924 1.088 2.8052 1.5282
1.7112 -1.9702 -2.1241 -1.119 -2.1416;-5.6227 -2.9359 -5.1598 -1.9995
3.3563 -2.1478 -1.1079 -0.5171 -2.5854 -2.1365 -7.2915 -2.0034 -0.66694
3.8991; 5.4132 6.6911 8.0139 2.105 1.9986 0.64987 -3.642 2.4143 1.8268 -
4.9486 -1.0021 -1.1945 3.4718 -0.8973; 2.0266 -7.454 1.3983 -3.9571 -
3.727 0.90767 1.1262 -0.57883 -3.5763 1.5037 -0.67604 5.1035 0.84672
8.1401; -4.9521 -6.6071 -4.3675 -1.4431 3.2732 -1.0174 -2.3489 -0.9126 -
3.3748 -5.6004 5.7381 3.6232 4.4221 -6.0727; -1.0729 -1.1463 0.44026
0.43318 -2.2914 -7.2721 1.4606 -0.35949 -3.4029 -9.1843 6.0433 3.7474
5.19 8.2773; 3.5892 1.0047 3.1838 -0.23814 -0.70384 0.68331 2.3532 -
1.2039 -1.5897 -5.2191 1.8094 1.6576 -0.66088 3.9373; -1.4498 1.0692
0.6584 2.1981 1.1464 1.2261 1.4899 -0.36722 1.7651 3.1035 0.07542 -1.0101
-4.0412 2.4163; -0.66064 0.70372 1.3265 0.99124 3.2711 -0.35754 0.35276
6.354 5.557 -0.61434 -3.6405 -5.3223 1.9479 -0.44805; -6.222 -3.5328 -
5.254 -0.90716 6.6831 -4.1497 -1.3832 4.7729 2.9549 -3.3491 -3.9825 -
0.94116 3.3022 3.4519; -9.7213 -6.4708 -11.2855 0.40561 -1.1288 2.3519 -
0.92438 -1.8488 0.32628 1.9579 -0.27541 -1.0277 -0.3393 5.1009; -3.1576
-0.54525 -3.6448 3.7029 -0.67909 -5.2569 0.81181 0.6968 1.2096 -2.1519 -
3.7194 -3.8925 2.8535 2.5895; 8.6326 7.1054 6.3095 2.2079 2.0536 1.1443
0.20245 -0.037824 -2.6593 4.0423 5.2641 2.0037 2.1357 -0.097536; 2.0094
2.4241 2.2646 -0.86484 13.7258 0.12856 -0.65304 3.5117 -0.28101 7.6716
0.55051 1.4205 2.3747 2.2273; -7.0803 -3.5639 -6.2683 -0.3033 -0.079749
-2.2972 0.51447 3.4812 2.193 1.2621 1.093 -1.3654 2.0307 2.2762; 5.1121
7.5843 5.8872 5.2498 -4.8857 -5.3783 2.6981 -2.6156 -1.7424 4.1937 -1.156
-0.53333 -2.2198 3.5569; -6.1663 -3.8152 -4.0101 -3.0651 0.39599 -5.6859
0.074771 -3.4632 -5.9198 -6.5424 -0.22293 1.7816 4.5347 5.6571; 0.48632
0.98995 -0.46805 1.2702 0.35996 1.6386 -1.3757 -0.94359 0.12842 3.3791 -
1.086 1.858 -1.5647 -0.99383; 0.0012976 -3.3715 0.01008 -2.4662 -0.88009
-1.9812 -0.64657 4.4196 1.2858 -0.85308 -0.58112 -1.8495 1.112 1.8802; -
4.7515 -4.2692 -6.9638 -0.90432 1.2899 13.4423 -3.3777 -1.628 -4.3997
8.2487 -10.1584 -0.77697 0.62874 3.5362; 7.8476 6.09 7.0754 1.244 -
2.8057 1.2301 -0.49436 1.053 0.15292 6.0354 1.1107 1.063 2.5757 4.5361; -
1.8434 0.82592 -3.1679 0.49732 -1.6936 2.3813 1.539 4.5436 2.9577 -1.9404
7.5181 0.78486 4.7656 -8.9516];

```

```

% Matriz de pesos entre la capa oculta con la capa de salida

```

```

IW2 = [0.67675 0.39583 -0.69802 -2.5758 2.5484 -3.9604 1.0869 -0.86744
2.3986 0.61037 0.26155 -0.6199 -2.2871 -2.799 2.5221 -7.3698 -1.1597 -
2.7373 0.13077 -2.7343 -2.102 0.020353; -8.0208 -0.5254 1.073 -0.66961 -
0.21185 -5.9256 -4.5782 -0.71272 0.27011 4.2598 -3.5112 -1.5897 -3.6281 -
0.94982 -2.9132 1.9389 -2.2032 -1.7076 -2.032 -1.6887 -3.4388 1.3087;-
1.5181 -3.9306 -0.62496 0.48325 -1.9049 -9.0928 -2.9501 0.32745 -1.742 -
8.2227 -2.3119 -0.78099 9.2322 -5.2169 -1.7265 -0.7957 -0.82958 -1.8608 -
0.27891 -0.79897 -9.938 7.0414; -2.6662 -6.866 0.87009 -1.8763 -1.8468
2.7824 2.6928 -3.8957 0.62541 -2.8937 -1.7644 -5.5801 -0.6743 2.0793
1.5517 -1.7274 4.4143 -0.37937 -0.87517 -1.706 8.5036 -5.1946; -0.63179

```

```

7.0686 -1.3634 -1.5879 -3.2219 9.9138 -0.96013 -1.172 -0.69904 0.65288 -
1.0601 2.2051 1.8644 5.4251 1.2466 -2.0635 -13.4016 -4.8523 -0.75179 -
2.3969 -6.4904 -7.1493; -1.4485 1.8755 -2.3986 0.62666 -11.5745 0.54332
-2.5067 -1.7856 2.0312 3.1863 -4.0165 4.647 5.4661 1.202 -0.45257 -2.1009
8.9338 1.1944 -3.6516 -0.66175 -1.1429 -5.6048;-2.6382 -1.3893 -3.2214
0.25888 6.9377 0.22219 -1.5162 -1.4784 7.2273 0.30508 -3.8514 -0.93901
4.0085 -3.6645 -3.014 -2.9844 1.1524 -0.94215 -1.8613 -2.9439 -1.8965 -
0.40326; 0.83737 1.7437 0.76203 -0.057493 1.1911 1.0653 -2.1246 -5.6101
-1.2411 -8.898 -5.484 2.1968 -2.573 5.2281 -2.3333 -0.92998 4.1838 1.1889
1.3564 1.9298 0.33594 4.7563; -0.67112 1.3497 3.5317 -5.0037 6.1164
0.2196 4.6704 -2.7286 -8.7991 1.9667 -12.532 -2.8256 -8.2473 0.9061 -
3.9734 -0.12821 0.022834 -3.3594 2.5166 -3.3895 -0.96978 -3.6576;
0.94035 -0.52811 0.86467 0.56007 -0.70383 -1.3138 -0.8045 0.62498 -3.7164
0.53981 1.0066 -1.235 1.9425 -0.45888 -1.1844 0.32416 -0.74491 -2.7197
0.44373 -0.72151 -0.84433 0.73577; -3.9981 -1.5566 -0.60824 9.8392 -
1.7287 -2.2328 -6.7937 -1.0739 4.8159 -3.4469 14.3472 -2.0326 -7.8287 -
2.5907 -7.3741 7.0624 3.4927 -3.2856 -4.4775 -12.7706 -2.684 -0.23542;
1.5894 -1.5806 11.6081 -8.1133 0.93555 1.5467 -3.4787 -5.2188 -4.9018
3.6147 7.1385 -3.0716 2.3589 3.9963 2.5522 -3.1614 -1.0756 -4.8205 -
6.9184 9.7784 0.80936 -1.57; -0.7297 0.26856 -10.9482 -1.3652 -5.6493 -
0.24468 -2.374 -0.85215 -4.6076 2.3155 2.0956 2.5782 -1.3911 0.10195
5.3883 10.572 -3.308 0.73999 3.2076 -2.0638 1.281 0.17041; -3.2001 -
0.73879 -5.8114 -0.22817 0.42767 3.7998 2.8063 -2.3863 -1.0517 2.0477 -
0.15015 4.8118 -1.4035 0.044737 -0.32554 -7.8792 -2.4724 0.072582 -2.662
6.2901 0.0076098 1.6041];

```

```
% Umbrales de activación.
```

```

b1=[-4.8015;4.605;2.7813;-4.4797;-1.8371;4.9804;-2.7844;1.8809;-0.26533;
-1.0355;0.6203; .11537;-2.1792;-3.6943;3.3481;6.2496;5.2913;1.9681;3.277;
4.8812;-5.7582;3.1074];

```

```

b2=[1.7875;3.8519;9.7808;3.2029;-6.6866;-9.1459;-1.5502;-5.0534;-
2.6295;0.467;2.6121;-7.5105;-7.5611;-5.7004];

```

```

SrtMinMax.name='mapminmax';
SrtMinMax.xrows = 14;
SrtMinMax.xmax =

```

```

[182.1891;13.6592;15.2306;0.9335;0.8784;0.1679;0.1529;0.4157;0.4169;0.182
4;0.5719;0.8717;13.3047;2.9151];

```

```

SrtMinMax.xmin =
[63.2426;7.6102;8.9735;0.8376;0.5176;0.0612;0;0.0039;0.0398;0.0675;0.1767
; 0.2152;2.1609;-0.1908];

```

```

SrtMinMax.xrange =
[118.9465;6.049;6.2571;0.0959;0.3608;0.1067;0.1529;0.4118;0.3771;0.1149;0
.3952;0.6565;11.1438;3.1059];

```

```

SrtMinMax.yrows = 14;
SrtMinMax.ymax = 1;
SrtMinMax.ymin = -1;
SrtMinMax.yrange = 2;

```

```
DataForClasificarForm = mapminmax('apply',DataForClasificar',SrtMinMax);
```

```

P1= IW1*DataForClasificarForm;
[M,N] = size(P1);
S1 = zeros(M,N);
for j=1:N

```

```

        for i=1:M
            S1(i,j) = P1(i,j) + b1(i);
        end
    end
    A = logsig(S1);

    P2= IW2*A;
    [M,N] = size(P2);
    S2 = zeros(M,N);
    for j=1:N
        for i=1:M
            S2(i,j) = P2(i,j) + b2(i);
        end
    end
    B = logsig(S2);

    Umbr = 0.25;
    [M,N]=size(B);
    for i=1:N
        for j=1:M
            Temp(j) =B(j,i);
            B(j,i) = 0;
        end
        [M1,I]=max(Temp);
        if(M1 > Umbr)
            B(I,i)=1;
        end
    end
    DataResultClasifica = B;

end
else
    PerimetroHuellas={};
end
end

```

## 9.2.2. Código Java que permite hacer uso de las funciones .m desarrolladas en Matlab.

Archivo: ProcesImageWrapper.java

```

package alfatrackanalysisfinal;

import ProcesImage.*; // Libreria que contine ls funciones matlab
import com.mathworks.toolbox.javabuilder.*;
import java.util.Vector;

public class ProcesImageWrapper {

    private ProcesImageclass Process;

    private int NumHuellas = 0;
    private Double [][] DataForTable; // Resultados del proceso

    private double [][] DataCentroid; // Centroide de las huellas
    // encontradas.
    private Vector VectorPerimet; // Perimetro de las huellas
    // encontradas
}

```

```

// Constructor:

public ProcesImageWrapper (){

    try
    {
        Process = new ProcesImageclass();
    }
    catch(Exception ex)
    {
        System.out.println("Exception: " + ex.toString());
    }
}

public void ProcesImage (String PathImagen, Double CalivPixelX, Double
CalivPixelY, int RadioFitroPromedio, int RadioFitroGauss, Double
SigmaGauss, int STR, Double SobelFactor, boolean UmbralGlobal, boolean
Umbral16Win, boolean ClearBorder, int AreaMinima, int AreaMaxima, boolean
SeparHuellas, Double RestOpacidad, Double RestRedondez, Double
RestExentric, boolean IsClasificacion){

    // Declarando variables

    MWNumericArray MatCalivPixelX = null;
    MWNumericArray MatCalivPixelY = null;
    MWNumericArray MatRadioFitroPromedio = null;
    MWNumericArray MatRadioFitroGauss = null;
    MWNumericArray MatSigmaGauss = null;
    MWNumericArray MatSTR = null;
    MWNumericArray MatSobelFactor = null;
    MWLogicalArray MatUmbralGlobal = null;
    MWLogicalArray MatUmbral16Win = null;
    MWLogicalArray MatClearBorder = null;
    MWNumericArray MatAreaMinima = null;
    MWNumericArray MatAreaMaxima = null;
    MWLogicalArray MatSeparHuellas = null;

    MWNumericArray MatVarRestOpacidad = null;
    MWNumericArray MatVarRestExentric = null;
    MWNumericArray MatRestRedondez = null;

    MWCharArray MatVarRutaImagene=null;

// Declaracion de Nuevas variables para la clasificacion

    MWLogicalArray MatIsClasificacion = null;

// Variables de Ingreso para la funcion matlab ProcesImagen encapsulada
// en la clase ProcesImageclass

    MatCalivPixelX = new
MWNumericArray(CalivPixelX,MWClassID.DOUBLE);

    MatCalivPixelY = new
MWNumericArray(CalivPixelY,MWClassID.DOUBLE);

    MatRadioFitroPromedio = new
MWNumericArray(RadioFitroPromedio,MWClassID.DOUBLE);

```

```

        MatRadioFitroGauss = new
MWNumericArray(RadioFitroGauss,MWClassID.DOUBLE);

        MatSigmaGauss = new MWNumericArray(SigmaGauss,MWClassID.DOUBLE);
        MatSTR = new MWNumericArray(4, MWClassID.DOUBLE);
        MatSobelFactor = new
MWNumericArray(SobelFactor,MWClassID.DOUBLE);

        MatUmbralGlobal = new MWLogicalArray(UmbralGlobal);
        MatUmbral16Win = new MWLogicalArray(Umbral16Win);
        MatClearBorder = new MWLogicalArray(ClearBorder);
        MatAreaMinima = new MWNumericArray(AreaMinima,MWClassID.DOUBLE);
        MatAreaMaxima = new MWNumericArray(AreaMaxima,MWClassID.DOUBLE);
        MatSeparHuellas = new MWLogicalArray(SeparHuellas);

MatVarRestOpacidad = new MWNumericArray(RestOpacidad, MWClassID.DOUBLE);
MatVarRestExentric = new MWNumericArray(RestExentric, MWClassID.DOUBLE);
        MatRestRedondez = new MWNumericArray(RestRedondez, MWClassID.DOUBLE);

        Object[] Result = null;    /* Resultados */

        MWNumericArray MatNumerHuellas = null;
        MWStructArray MatExtraCaracteristicas = null;
        MWCellArray MatPerimetroHuellas = null;
        MWArray MatCentroides = null;
        MWArray FieldValCaract = null;

        MWArray MatDataResultClasifica = null;

        MatVarRutaImagene = new MWCharArray(PathImagen);

        MatIsClasificacion = new MWLogicalArray(IsClasificacion);

        DataForTable = null; // Resultados del proceso
        DataCentroid = null; // Centroide de las huellas encontradas
        VectorPerimet = null; // perimetro de las huellas encontradas

        try
        {

            Result =
Process.ProcesImagenFinal(5,MatVarRutaImagene,MatCalivPixelX,MatCalivPixelY,MatRadioFitroPromedio,MatRadioFitroGauss,MatSigmaGauss,MatSTR,MatSobelFactor,MatUmbralGlobal,MatUmbral16Win, MatClearBorder,
MatAreaMinima,MatAreaMaxima,MatSeparHuellas,MatVarRestOpacidad,MatRestRedondez,MatVarRestExentric,MatIsClasificacion);

            MatExtraCaracteristicas = (MWStructArray) Result[0];
            MatNumerHuellas = (MWNumericArray) Result[1];
            MatPerimetroHuellas =(MWCellArray) Result[2];
            MatCentroides = (MWArray) Result[3];
            MatDataResultClasifica = (MWArray) Result[4];

            NumHuellas = Integer.parseInt(MatNumerHuellas.toString());

            if (getNumHuellas())>0){
                int[] dimC = MatPerimetroHuellas.getDimensions();
                int[] dimB = MatCentroides.getDimensions();
                DataCentroid = new double[dimB[0]][dimB[1]];
            }
        }
    }

```

```

        DataCentroid = (double[][]);
MatCentroides.toArray();

        VectorPerimet = new Vector();

        for (int k=1; k< (dimC[0]+1); k++) {
            MWArray DataPeriCell =
MatPerimetroHuellas.getCell(k);
            Object[] DataPerimCell =
DataPeriCell.toArray();
            getVectorPerimet().addElement(DataPerimCell);
        }

        int[] dimE =
MatDataResultClasifica.getDimensions();
        double [][] DataClasificPre = (double[][]);
MatDataResultClasifica.toArray();
        if (IsClasificacion){

            Double [][] Clasifi = new Double[dimE[1]][1];
            Double TempDataClasific = new Double (0);

            for (int i=0;i<dimE[1];i++){

                for(int j=0;j<dimE[0];j++){
                    int Data
=Double.valueOf(DataClasificPre[j][i]).intValue();
                    if (Data == 1) {
                        switch (j) {

                            case 0: TempDataClasific = new Double(0.15); break;
                            case 1: TempDataClasific = new Double(1.06); break;
                            case 2: TempDataClasific = new Double(1.44); break;
                            case 3: TempDataClasific = new Double(1.90); break;
                            case 4: TempDataClasific = new Double(2.04); break;
                            case 5: TempDataClasific = new Double(2.43); break;
                            case 6: TempDataClasific = new Double(2.91); break;
                            case 7: TempDataClasific = new Double(3.23); break;
                            case 8: TempDataClasific = new Double(3.63); break;
                            case 9: TempDataClasific = new Double(3.91); break;
                            case 10: TempDataClasific = new Double(4.25); break;
                            case 11: TempDataClasific = new Double(4.39); break;
                            case 12: TempDataClasific = new Double(5.88); break;
                            case 13: TempDataClasific = new Double(7.89); break;

                        }
                    }
                }
                Clasifi[i][0]= TempDataClasific;
            }

            setDataForTable(new
Double[MatExtraCaracteristicas.numberofElements()][MatExtraCaracteristica
s.numberofFields() + 1]);

            String[] STRfilnames = MatExtraCaracteristicas.fieldNames();

            for (int element = 1; element <=
MatExtraCaracteristicas.numberofElements(); element++) {

```



```

        for (int field = 0; field <
MatExtraCaracteristicas.numberOfFields(); field++) {
            FieldValCaract =
MatExtraCaracteristicas.getField(STRFilnames[field], element);
            DataForTable[element-
1][field]=Double.valueOf(FieldValCaract.toString());
        }
        DataForTable[element-
1][MatExtraCaracteristicas.numberOfFields()] = Clasifi[element-1][0];
    } else{

        setDataForTable(new
Double[MatExtraCaracteristicas.numberOfElements()][MatExtraCaracteristica
s.numberOfFields()]);
        String[] STRFilnames =
MatExtraCaracteristicas.fieldNames();

        for (int element = 1; element <=
MatExtraCaracteristicas.numberOfElements(); element++) {

            for (int field = 0; field <
MatExtraCaracteristicas.numberOfFields(); field++) {
                FieldValCaract =
MatExtraCaracteristicas.getField(STRFilnames[field], element);
                DataForTable[element-
1][field]=Double.valueOf(FieldValCaract.toString());
            }
        }
    }
}
catch(Exception e)
{
    System.out.println("Exception ocurrida en wrapper: " +
e.toString());
}
finally
{
    MWArray.disposeArray(Result);

    MatCalivPixelX.dispose();
    MatCalivPixelY.dispose();
    MatRadioFitroPromedio.dispose();
    MatRadioFitroGauss.dispose();
    MatSigmaGauss.dispose();
    MatSTR.dispose();
    MatSobelFactor.dispose();
    MatUmbralGlobal.dispose();
    MatUmbral16Win.dispose();
    MatClearBorder.dispose();
    MatAreaMinima.dispose();
    MatAreaMaxima.dispose();
    MatSeparHuellas.dispose();

    MWArray.disposeArray(MatVarRestOpacidad);
    MatVarRestExentric.dispose();
    MatRestRedondez.dispose();

    MatVarRutaImagene.dispose();
}

```

```

        MatNumerHuellas.dispose();
        MatExtraCaracteristicas.dispose();
        MatPerimetroHuellas.dispose();

        MatDataResultClasifica.dispose();
        MatIsClasificacion.dispose();

    }

}

public void dispose()
{
    if(Process != null)
        Process.dispose();
}
/**
 * @return the NumHuellas
 */
public int getNumHuellas() {
    return NumHuellas;
}

/**
 * @return the Results
 */
public Double[][] getResults() {
    return DataForTable;
}

/**
 * @return the DataCentroid
 */
public double[][] getDataCentroid() {
    return DataCentroid;
}

/**
 * @return the VectorPerimet
 */
public Vector getVectorPerimet() {
    return VectorPerimet;
}

/**
 * @param DataForTable the DataForTable to set
 */
public void setDataForTable(Double[][] DataForTable) {
    this.DataForTable = DataForTable;
}

}

```

### 9.3. Anexo 3.

#### Contenido del cd adjunto

1. Instalador del JDK: jdk-6u21-windows-i586.exe.

2. Instalador del Matlab Component Runtime (MCR): MCRInstaller.exe.
3. Ejecutable del software desarrollado. AlfaTrackAnalysisFinal.jar

## **9.4. Anexo 4**

### **Manual de instalación**

El Software esta disponible en un fichero .jar (AlfaTrackAnalysisFinal.jar). Para ejecutar el fichero, el usuario deberá realizar una serie de pasos previos:

1. Instalar el JDK.
2. Definir la variable de usuario en 'Panel de Control > Sistema > Configuración Avanzada > Opciones Avanzadas > Variables de Entorno':

. Java\_HOME con el valor de path donde este instalado el jdk, por ejemplo:

C:\Archivos de programa\Java\jdk1.6.0\_21

3. Instalar el Matlab Component Runtime (MCR).

## **9.5. Anexo 5**

**Variación de los parámetros en función de la energía de las partículas alfa incidentes después de haber agrupado energías cuya diferencia es menor a 100 keV.**

**Tabla 9.3**

**Variación de los parámetros en función de la energía de las partículas alfa incidentes después de haber agrupado las energías**

Energía (MeV)	Área ( $\mu\text{m}^2$ )	$\sigma_{\text{Area}}$ ( $\mu\text{m}^2$ )	Eje Mayor ( $\mu\text{m}$ )	$\sigma_{\text{Eje Mayor}}$ ( $\mu\text{m}$ )	Eje Menor ( $\mu\text{m}$ )	$\sigma_{\text{Eje Menor}}$ ( $\mu\text{m}$ )	Elongaci.	$\sigma_{\text{Elong}}$	Perímetro ( $\mu\text{m}$ )	$\sigma_{\text{Perim}}$ ( $\mu\text{m}$ )	D. Equi ( $\mu\text{m}$ )	$\sigma_{\text{DEqui}}$
0.15	82.56	13.17	11.74	0.97	8.89	0.71	0.758	0.020	33.67	2.80	10.22	0.82
1.06	142.47	4.09	15.58	0.30	11.65	0.18	0.748	0.010	44.65	0.75	13.47	0.20
1.44	162.54	1.59	16.60	0.11	12.45	0.09	0.750	0.006	47.68	0.26	14.39	0.07
1.9	169.60	6.80	16.86	0.38	12.82	0.37	0.761	0.022	48.68	0.97	14.69	0.30
2.04	158.63	2.92	16.43	0.17	12.27	0.10	0.747	0.005	47.11	0.43	14.21	0.12
2.43	146.24	3.21	15.72	0.17	11.83	0.16	0.753	0.006	45.20	0.51	13.64	0.15
2.91	137.60	3.37	15.27	0.23	11.45	0.18	0.750	0.011	43.85	0.62	13.23	0.15
3.23	128.26	5.65	14.73	0.34	11.05	0.25	0.750	0.011	42.27	0.92	12.77	0.28
3.63	122.99	2.95	14.50	0.29	10.79	0.14	0.747	0.015	41.44	0.56	12.51	0.15
3.91	119.24	3.42	14.21	0.22	10.70	0.19	0.753	0.011	40.71	0.58	12.32	0.15
4.25	109.95	4.53	13.56	0.35	10.30	0.33	0.756	0.019	39.07	0.82	11.83	0.24
4.39	108.53	5.28	13.45	0.50	10.28	0.30	0.765	0.034	38.86	1.20	11.77	0.32
5.88	93.53	4.86	12.44	0.27	9.55	0.31	0.768	0.014	36.00	1.02	10.91	0.28
7.89	88.17	3.98	12.16	0.33	9.16	0.27	0.754	0.023	34.96	0.76	10.59	0.24

**Tabla 9.4**

**Variación de los parámetros en función de la energía de las partículas alfa incidentes después de haber agrupado las energías**

Energía (MeV)	Redon	$\sigma_{\text{Redon}}$	NGMaxE	$\sigma_{\text{NGMaxE}}$	NGMinE	$\sigma_{\text{NGMinE}}$	NGME	$\sigma_{\text{NGME}}$	NGPE	$\sigma_{\text{NGPE}}$	SNGPE	$\sigma_{\text{SNGPE}}$
0.15	0.910	0.010	0.800	0.018	0.416	0.051	0.636	0.022	0.635	0.022	0.022	0.072
1.06	0.898	0.011	0.816	0.013	0.407	0.031	0.628	0.019	0.621	0.022	0.022	0.073
1.44	0.898	0.004	0.802	0.011	0.370	0.019	0.617	0.011	0.616	0.013	0.019	0.053
1.9	0.899	0.008	0.810	0.057	0.379	0.065	0.624	0.044	0.632	0.047	0.047	0.073
2.04	0.898	0.004	0.817	0.020	0.393	0.031	0.618	0.018	0.615	0.017	0.022	0.052
2.43	0.900	0.003	0.826	0.012	0.404	0.021	0.618	0.012	0.617	0.010	0.014	0.053
2.91	0.899	0.007	0.820	0.022	0.381	0.030	0.618	0.020	0.613	0.019	0.024	0.053
3.23	0.901	0.005	0.819	0.027	0.348	0.075	0.615	0.030	0.609	0.029	0.037	0.059
3.63	0.900	0.007	0.800	0.023	0.345	0.028	0.607	0.019	0.604	0.019	0.025	0.054
3.91	0.904	0.005	0.811	0.029	0.357	0.024	0.613	0.023	0.609	0.024	0.024	0.076
4.25	0.905	0.005	0.779	0.042	0.295	0.065	0.586	0.049	0.584	0.050	0.061	0.058
4.39	0.907	0.008	0.809	0.024	0.207	0.129	0.573	0.053	0.564	0.055	0.055	0.101
5.88	0.907	0.009	0.759	0.092	0.353	0.067	0.581	0.086	0.589	0.087	0.087	0.075
7.89	0.906	0.007	0.794	0.056	0.306	0.054	0.607	0.056	0.610	0.057	0.057	0.079

- Redon : Redondez
- NGMaxE : Nivel de gris máximo del entorno.
- NGMinE : Nivel de gris mínimo del entorno.
- NGME : Nivel de gris mediana del entorno.
- NGPE : Nivel de gris promedio del entorno.
- SNGPE : Desviación estándar del nivel de gris promedio del entorno.

**Tabla 9.5**

**Variación de los parámetros en función de la energía de las partículas alfa incidentes después de haber agrupado las energías**

Energía (MeV)	NGMaxH	$\sigma_{\text{NGMaxH}}$	NGMinH	$\sigma_{\text{NGMinH}}$	NGMH	$\sigma_{\text{NHMH}}$	NGPH	$\sigma_{\text{NGPH}}$	SNGPH	$\sigma_{\text{SNGPH}}$
0.15	0.689	0.033	0.077	0.045	0.376	0.030	0.381	0.042	0.100	0.010
1.06	0.677	0.036	0.040	0.029	0.320	0.020	0.333	0.020	0.109	0.005
1.44	0.623	0.017	0.054	0.020	0.269	0.008	0.281	0.010	0.089	0.004
1.9	0.678	0.050	0.062	0.033	0.279	0.031	0.291	0.031	0.094	0.012
2.04	0.652	0.025	0.045	0.022	0.260	0.010	0.273	0.011	0.086	0.003
2.43	0.657	0.018	0.024	0.015	0.235	0.010	0.246	0.010	0.082	0.003
2.91	0.667	0.040	0.007	0.013	0.218	0.011	0.226	0.011	0.080	0.003
3.23	0.657	0.031	0.003	0.007	0.183	0.024	0.193	0.020	0.084	0.005
3.63	0.622	0.026	0.003	0.006	0.145	0.023	0.160	0.021	0.087	0.005
3.91	0.630	0.062	0.000	0.000	0.106	0.020	0.125	0.018	0.097	0.005
4.25	0.590	0.043	0.005	0.014	0.098	0.041	0.125	0.036	0.093	0.008
4.39	0.614	0.069	0.003	0.011	0.101	0.038	0.126	0.032	0.096	0.009
5.88	0.618	0.080	0.005	0.018	0.121	0.081	0.153	0.072	0.116	0.023
7.89	0.644	0.066	0.000	0.000	0.115	0.070	0.145	0.064	0.112	0.015

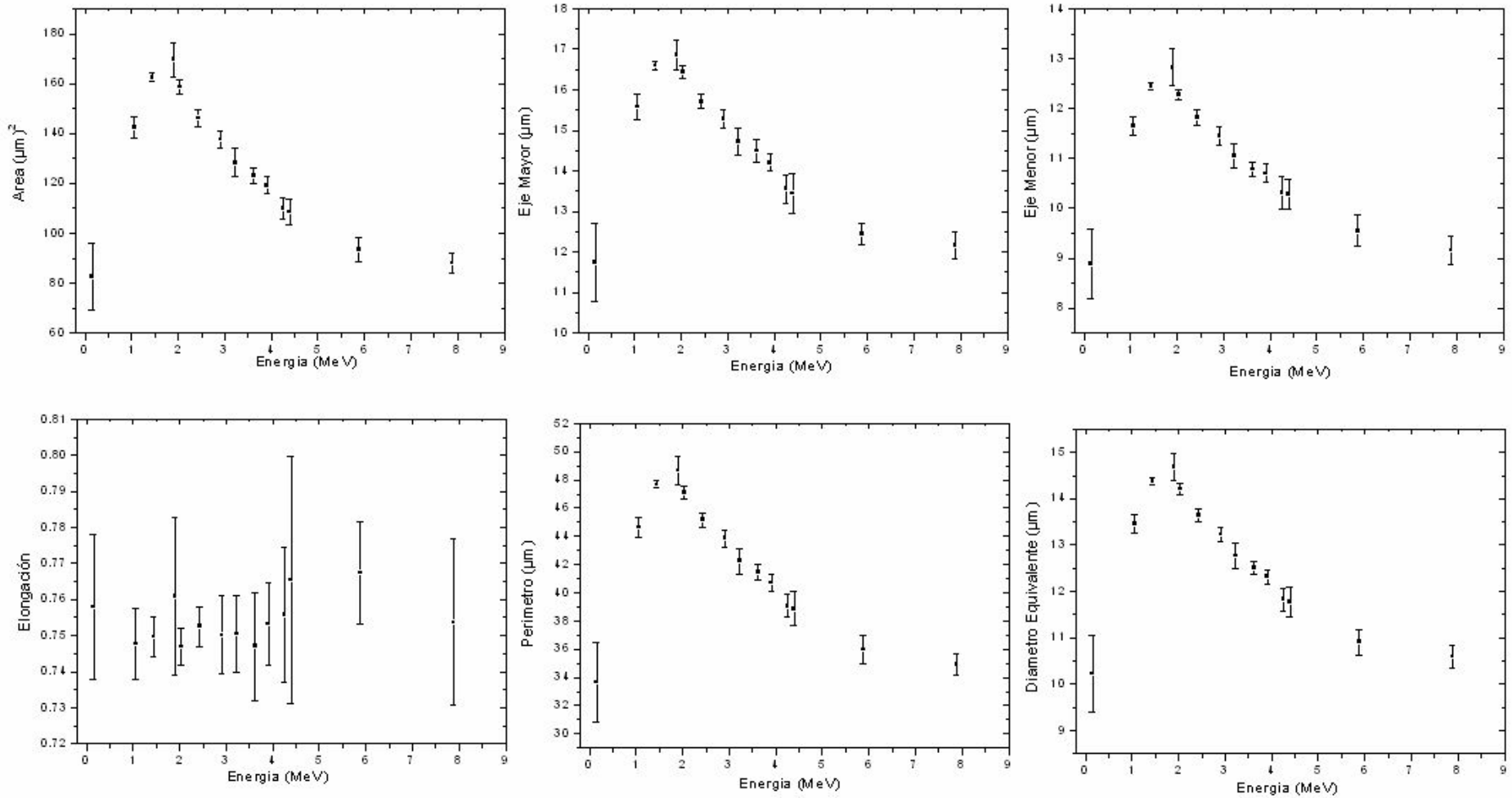
- NGMaxH : Nivel de gris máximo de la Huella.
- NGMinE : Nivel de gris mínimo de la Huella.
- NGME : Nivel de gris mediana de la Huella.
- NGPE : Nivel de gris promedio de la Huella.
- SNGPE : Desviación estándar del nivel de gris promedio de la Huella.

**Tabla 9.6**

**Variación de los parámetros en función de la energía de las partículas alfa incidentes después de haber agrupado las energías**

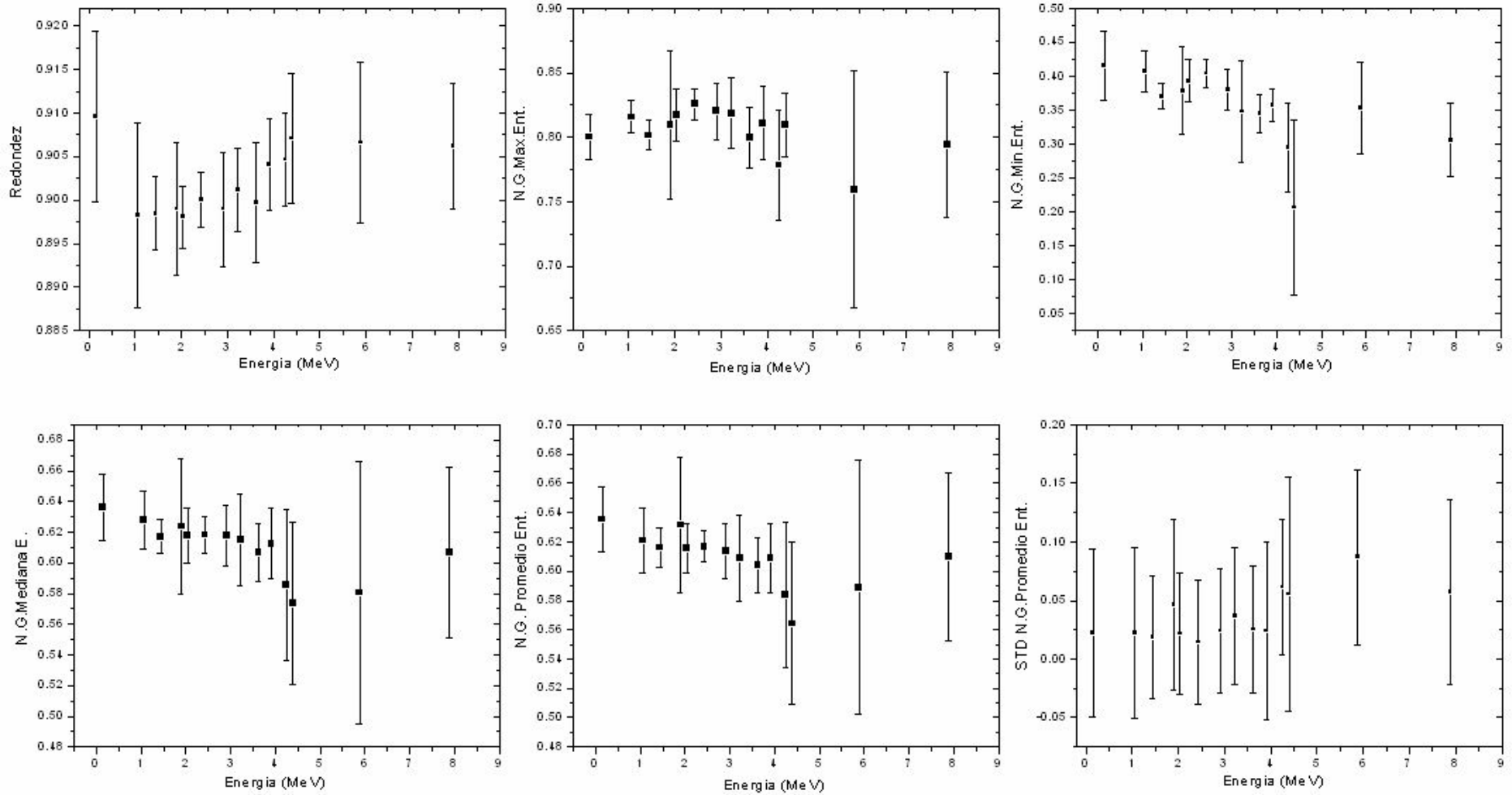
Energía (MeV)	Contraste Absoluto	$\sigma_{\text{ContAbs}}$	Contraste Relativo	$\sigma_{\text{ContRela}}$	Contraste Propor.	$\sigma_{\text{ContProp}}$	Skewness	$\sigma_{\text{Skew}}$	Kurtosis	$\sigma_{\text{Kurt}}$
0.15	0.261	0.038	0.251	0.030	0.599	0.037	2.78	0.23	0.07	0.20
1.06	0.288	0.018	0.302	0.021	0.537	0.025	2.65	0.15	0.34	0.06
1.44	0.333	0.012	0.372	0.015	0.458	0.016	3.40	0.21	0.57	0.08
1.9	0.340	0.039	0.369	0.040	0.462	0.043	3.49	0.59	0.65	0.18
2.04	0.343	0.014	0.386	0.014	0.443	0.015	3.89	0.30	0.73	0.07
2.43	0.372	0.010	0.431	0.014	0.398	0.014	5.73	0.68	1.13	0.16
2.91	0.387	0.018	0.461	0.018	0.369	0.018	6.71	0.92	1.26	0.17
3.23	0.416	0.028	0.520	0.037	0.317	0.031	7.39	0.93	1.47	0.23
3.63	0.445	0.025	0.585	0.039	0.262	0.034	7.62	0.73	1.77	0.16
3.91	0.484	0.028	0.660	0.040	0.205	0.030	8.13	2.07	1.98	0.45
4.25	0.459	0.055	0.652	0.080	0.217	0.063	7.23	1.17	1.91	0.30
4.39	0.429	0.065	0.635	0.076	0.226	0.058	7.21	1.87	1.86	0.50
5.88	0.436	0.112	0.594	0.149	0.266	0.128	5.51	2.33	1.53	0.74
7.89	0.465	0.085	0.624	0.139	0.241	0.111	5.92	2.61	1.56	0.78

**Figura 9.1**  
**Variación de los parámetros en función de la energía de las partículas alfa incidentes después de haber agrupado las energías**

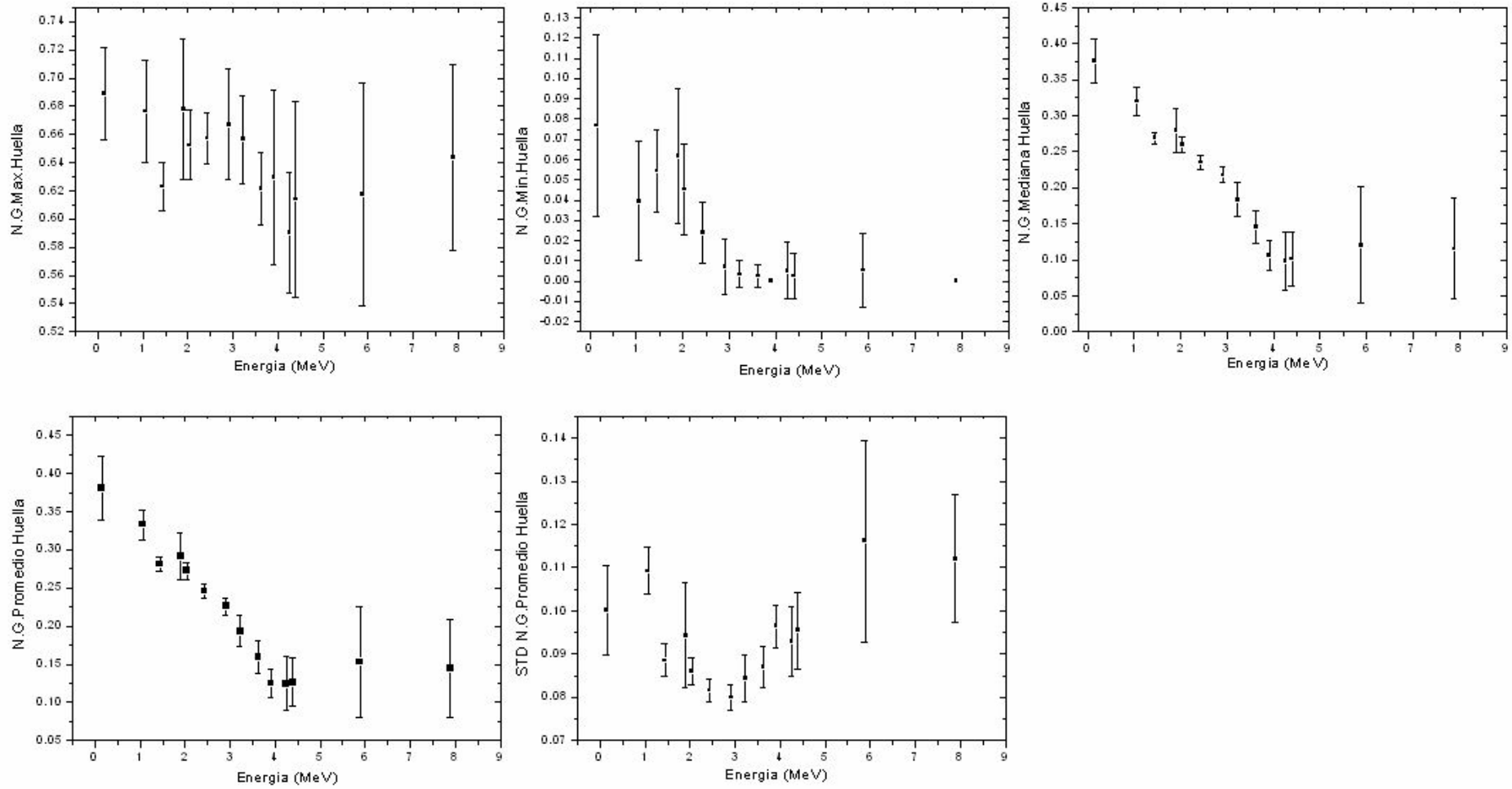




**Figura 9.2**  
**Variación de los parámetros en función de la energía de las partículas alfa incidentes después de haber agrupado las energías**



**Figura 9.3**  
**Variación de los parámetros en función de la energía de las partículas alfa incidentes después de haber agrupado las energías**



**Figura 9.4**  
**Variación de los parámetros en función de la energía de las partículas alfa incidentes después de haber agrupado las energías**

