

## **Capítulo 4: Diseño e implementación de la aplicación.**

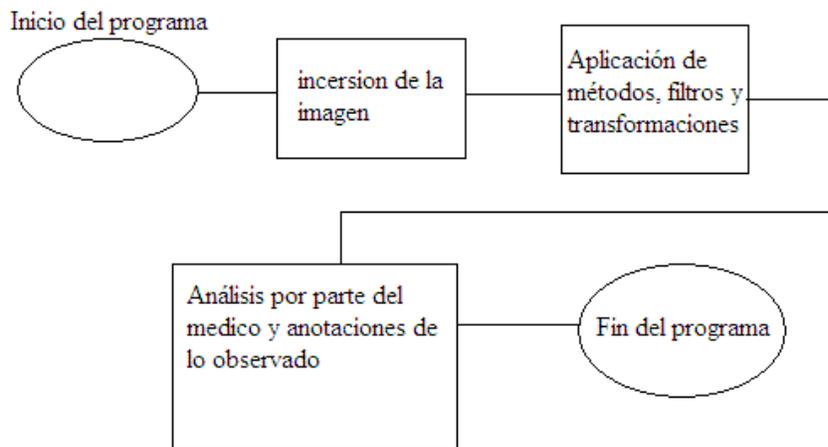
En este capítulo se tomaron los resultados obtenidos del análisis de las imágenes tratadas con el software neatvision; estos datos los ocuparemos para realizar una aplicación la cual tenga los modelos de procesamiento de imágenes seleccionados gracias a la investigación realizada, ya que la complejidad de uso del neatvision, para personas nos especializadas es alto y requiere de tiempo, así como los sistemas existentes para el análisis del cáncer de piel solo se encuentran en zonas urbanizadas y son muy costosos los estudios que se hacen con dichos sistemas; ahora los métodos que se analizaron en neatvision, se implementaran para hacer un software el cual trate a las imágenes para que posteriormente, el medico general pueda hacer uso de este sistema, así como para que por medio del tratado de la imagen obtengamos las características de la enfermedad y sea más sencillo para el medico dar un diagnóstico o recomendación al paciente.

### **4.1 Propuesta del sistema.**

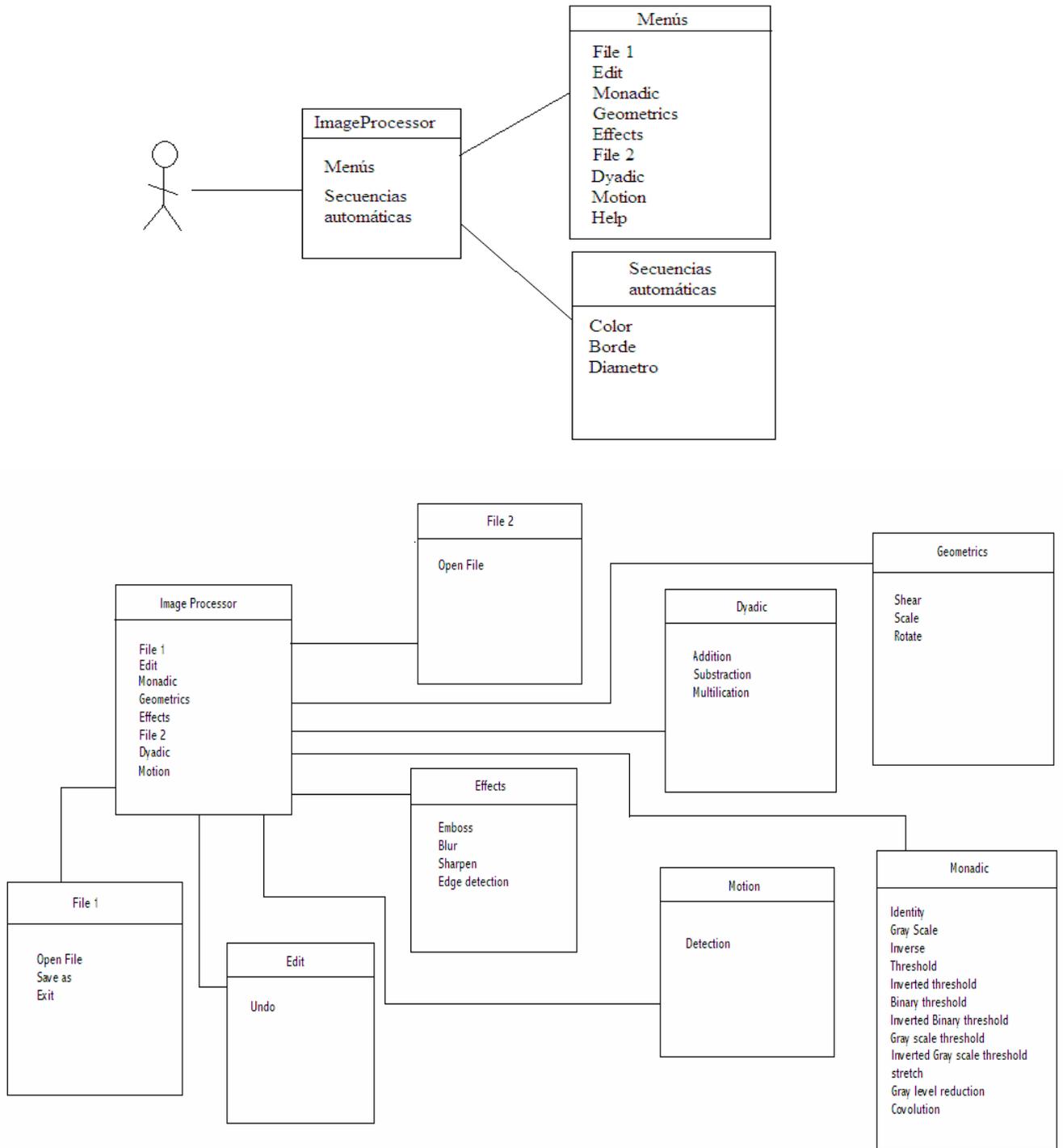
Hacer la implementación de una aplicación la cual usando la secuencia de métodos previamente investigada, pueda tratar a la imagen del melanoma cutáneo y de este modo regresar las características esenciales que el medico requiere para poder dar un diagnostico o una recomendación al paciente sobre que debe de hacer con respecto a la gravedad de la enfermedad.

## 4.2 Diseño del Sistema.

Para hacer el diseño del sistema primero se tuvo que hacer una serie de diagramas de bajo nivel para darnos una idea de cómo poder unir los métodos o algoritmos de procesamiento de imágenes y además para que pueda ser manejado en un futuro para la implementación del reconocedor de patrones y la base de datos.



**Imagen 4.1 Diagrama de flujo del sistema.**

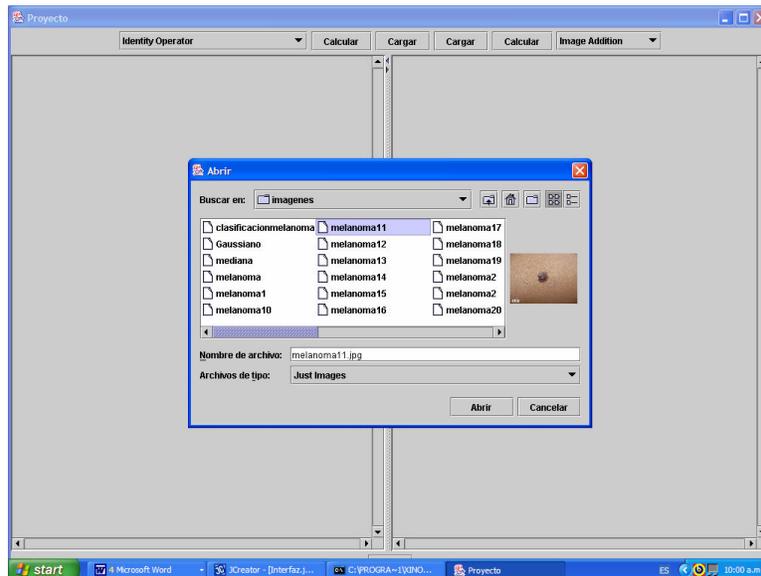


**Imagen 4.2 Diagrama del sistema.**

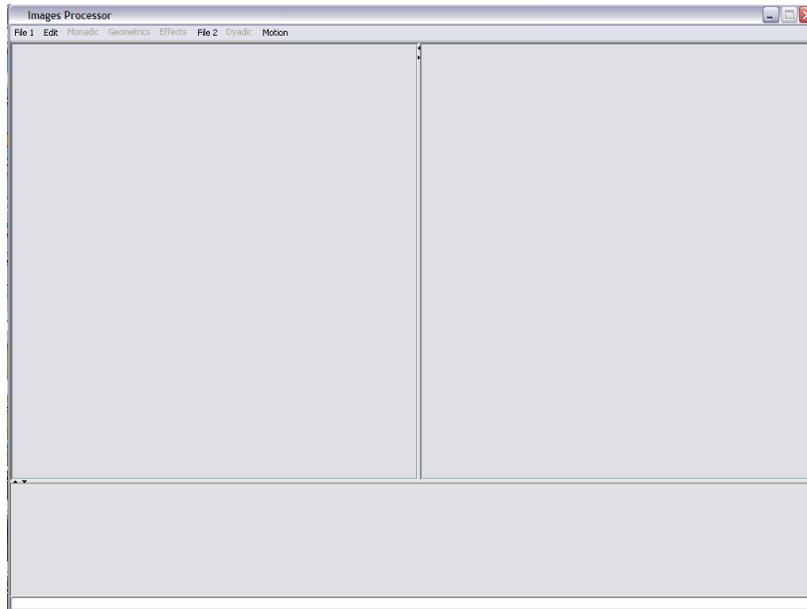
Después de realizar este estudio, y de verificar que neatvision tenia su propio compilador, se tuvo que realizar la programación de cada uno de los métodos y algoritmos de procesamiento de imágenes que previamente fueron seleccionados; para hacerlo se diseño un sistema en pseudo-código para que fuera más sencillo el acoplamiento de las funciones que realizara la aplicación, así como la aplicación automática de la secuencia.

### 4.3 Programación del sistema

En este punto, ya con los diagramas previamente realizados, se comenzó a programar el sistema, lo primero fue hacer una interfaz con la cual se pudiera extraer una imagen desde cualquier carpeta de nuestra computadora, se previo que el sistema pudiera leer cualquier tipo de imagen y que el tamaño no importara, ya que podremos tener diferentes tamaños de resolución de la imagen.

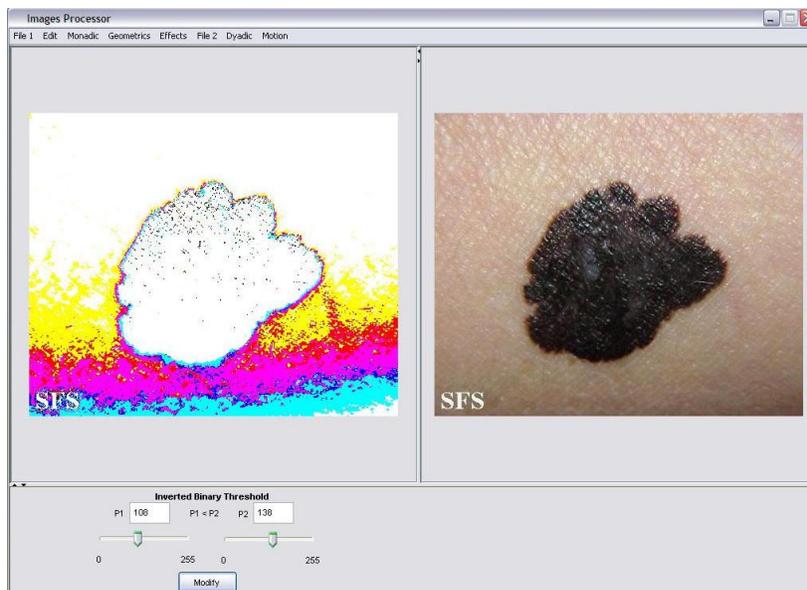


**Imagen 4.3 Muestra de la 1era interfaz del sistema**



**Imagen 4.4 Muestra de la interfaz mejorada.**

Una vez realizada la interfaz, entonces se continuó agregando los diferentes operadores de procesamiento de imágenes, en este caso se implementaron los umbrales, estos fueron colocados en un menú, para que de manera manual (de uno en uno), estos se puedan ir aplicando a la imagen.

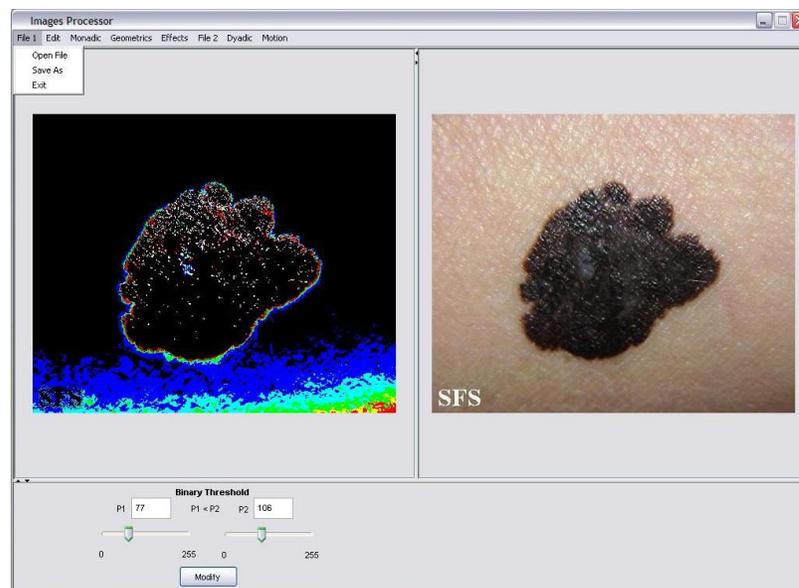


**Imagen 4.5 Ejemplo de la función Invertid binary threshold.**

Esto se hizo con el fin de poder probar todos los operadores y así una vez probados, en un futuro, poder introducirlos en otro menú el cual tenga la opción de aplicación automática, esto quiere decir que las funciones que estén dentro de este menú, es la combinación de varios operadores de procesamiento de imágenes, los cuales obtuvimos del análisis previo con el software de neatvision.

#### 4.4 Pruebas y evaluación del sistema.

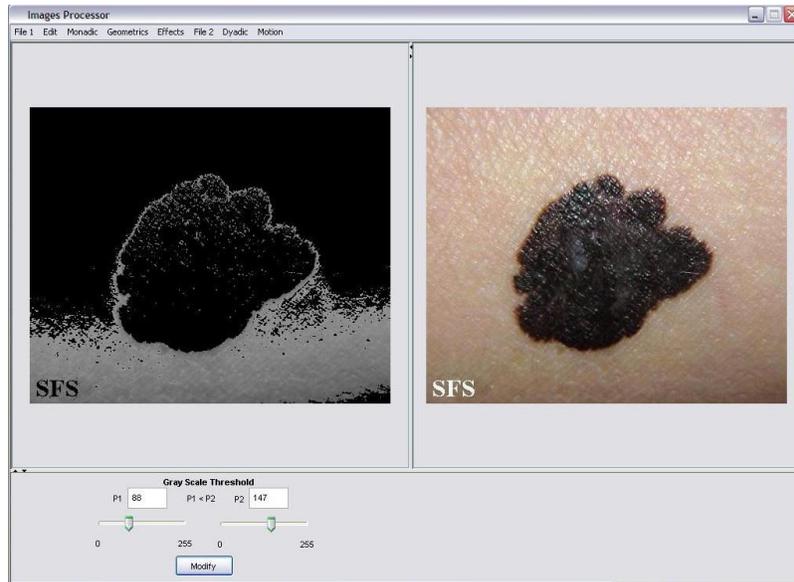
En esta parte del capítulo, se harán las pruebas del sistema, así como los comentarios sobre los métodos o transformaciones que se analizaron para aplicarlas a las imágenes de manera automática; también se darán las recomendaciones correspondientes para el uso del sistema.



**Imagen 4.6 Pruebas del umbral “Binary threshold Operator”.**

Como primer ejemplo tenemos lo que es el operador de binary threshold, esta imagen se colocó para hacer mención de que funciona el método, y que se puede

comenzar a usar; así como la siguiente imagen a la cual se esta aplicando el operador de Gray scale thrshold, con el cual se obtiene una imagen la cual se transformaron sus colores a grices y se le dio un umbral para la obtención de la característica deseada.



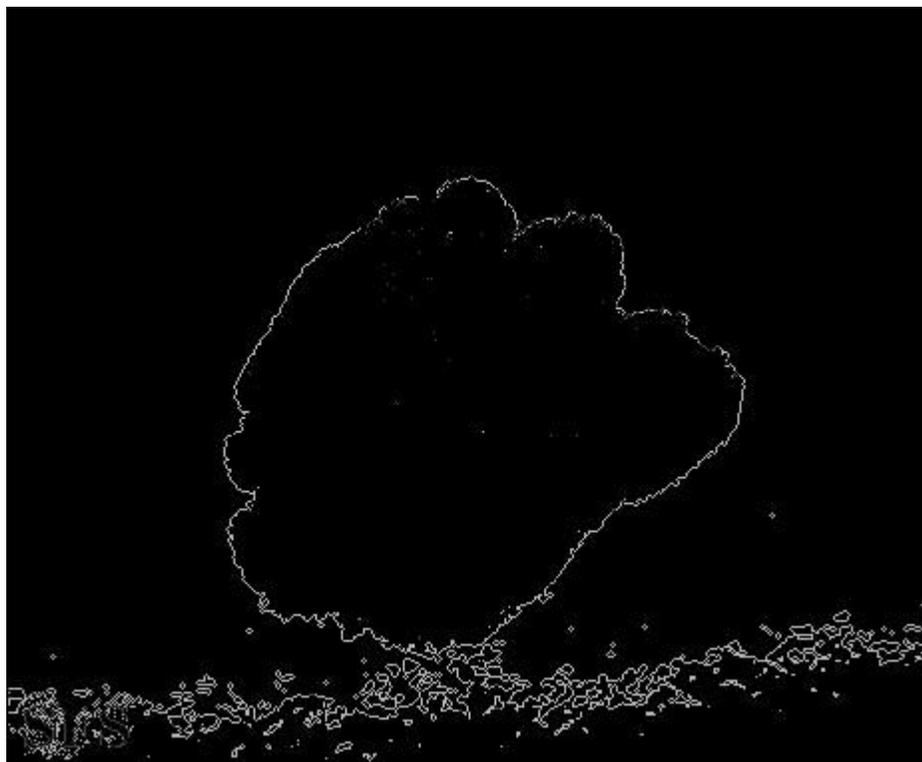
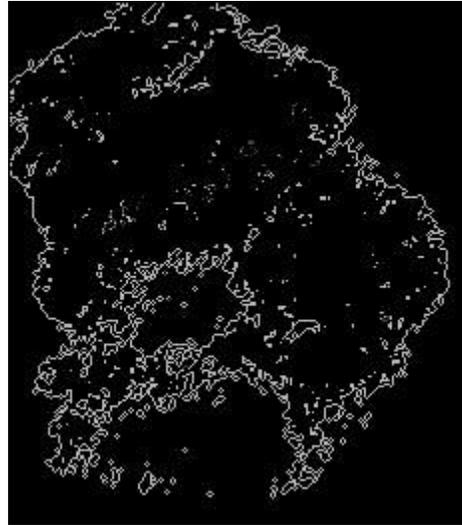
**Imagen 4.7 Pruebas del Umbral “Gray sacale threshold operator”.**

La pruebas que se realizaron, nos produjeron resultados ligeramente diferentes a los de neatvision, ya que como el software previamente mencionado tiene su propio compilador, es de difícil manejo para alguien que no es experto y por la diferente forma en cómo ellos manejan los operadores para tratar la imagen, sin mencionar que este programa puede unir los elementos sin necesidad de hacer la programación o la implementación del código; de ahí en fuera, al parecer se esta generando de manera adecuada todas las pruebas y las evaluaciones del sistema; las imágenes siguientes demuestran como se aplicaron los diferentes operaciones a la imagen, pero de manera automática para obtener las características.



**Imagen 4.8 Se aplico el operador de Color**

Para este operador de Color, se le aplicaron los diferentes métodos y filtros que previamente se analizaron para que de manera automática se obtenga la característica deseada la cual es mostrar las tonalidades de sombras o colores que tiene la lesión.

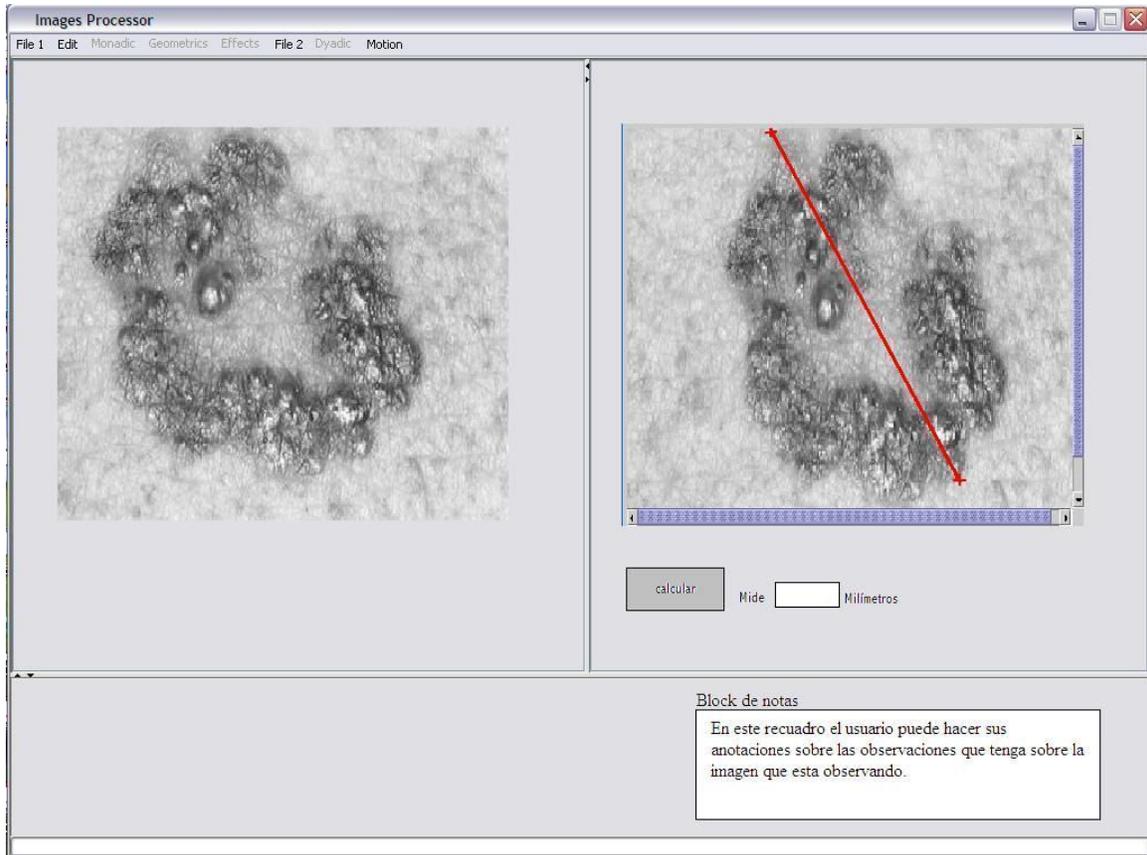


**Imagen 4.9** En este ejemplo se muestra el operador de Bordes.

En las imágenes anteriores se muestra como es que funciona el operador

de Borde, el cual ayuda al medico a tratar la imagen aplicando filtros y

transformaciones necesarias para obtener la característica del borde, y así pueda el medico analizar la imagen y tener una mejor visión de esta característica.



**Imagen 4.10 Ejemplo del operador de tamaño.**

Con este operador Tamaño, se muestra una imagen la cual puedes medir localizando dos puntos en la fotografía, los cuales deben formar una línea recta la cual marque la distancia mas larga dentro del perímetro de la lesión; esto se hace con el fin de que podamos obtener el tamaño de la enfermedad de manera automática; esta función también sirve para calcular el ancho de la lesión con la observación de que la fotografía sea tomada lateralmente desde la parte que el medico observe como la más elevada en la lesión.

#### 4.5 Conclusiones del capítulo.

Para finalizar este capítulo, podemos observar en la siguiente tabla todo el análisis, la secuencia de métodos, filtros o transformaciones que se usan para la obtención de las características que el médico necesita para diagnosticar o hacer recomendaciones al paciente sobre la lesión que se encontró.

<b>Características</b>	<b>Secuencia de métodos, filtros y transformaciones</b>	<b>Comentarios</b>
Color	Inverted Binary Threshold P1=93 y P2=106 Edge Detection Inverse Gray Scale Addition	La imagen debe ser lo más clara posible, que la lesión se tome lo más cerca que se pueda, que la luz sea de poca intensidad para que no afecte a los operadores.
Borde	Threshold (entre 100 y 120 según lo requiera la imagen). Inverse Edge Detection Gray Scale Inverse Stretch P1=60 y P2=115	La imagen de la lesión debe de ser tomada muy cerca de la misma; no debe de tener ningún tipo de estorbos como: vellosidades que se encuentren encima de la lesión.
Diámetro	La opción de tamaño, toma dos puntos seleccionados por el médico en la imagen, utiliza esos valores para hacer el cálculo de la distancia geométrica y después una hace una conversión para obtener el resultado en milímetros.	Necesita que la fotografía sea tomada desde arriba y que tenga buena resolución para poder observar bien cuales son los puntos extremos de la lesión.
Grosor	Utilizar la opción de tamaño.	La imagen debe ser tomada desde un costado, el médico debe decidir desde que costado se toma la fotografía, por que debe de observarla parte mas ancha de la lesión

Estos operadores automáticos se activan mediante los botones que se encuentran junto al block de notas; gracias a esto podemos decir que el software que se está desarrollando es estable y confiable, ya que puede utilizarse en cualquier computadora, que tenga tecnología java; también se logra una programación básica ya que este proyecto tiene alcances demasiado altos, los cuales serán aterrizados con el paso del tiempo en los trabajos futuros que se realicen sobre este software; al igual que la implementación de todos los métodos, filtros y transformaciones son de gran utilidad para mejorar la aplicación en el futuro y que además de hacer las cosas de manera automática, pueda tener otras aplicaciones.