

## **Práctica 6.-Transformaciones globales**

### **Índice:**

**1.-Introducción**

**2.-Objetivos** (En este apartado se presentan los objetivos de la práctica)

**3.-Material** (En este apartado se detalla el material de trabajo)

**4.-Desarrollo de la práctica** (En este apartado se realiza y valora lo que se hizo en la práctica)

**5.-Desarrollo del programa** (En este apartado se habla sobre el programa realizado en prácticas y se presenta una propuesta de mejora, además, se discute los posibles errores y mejoras de éste)

## 1.-Introducción

Debido a la poca diferencia de absorción fotoeléctrica al haz de rayos X entre tejidos, la radiografía de mama necesita de un tubo especial de rayos X (con ánodo de Molibdeno en lugar de Wolframio) para poder obtener una imagen que, por su particularidad, se la denomina mamografía.

La imagen obtenida digitalmente es una imagen bidimensional que nos muestra el grado de atenuación de las distintas secciones de la mama, la cual ha de estar comprimida a la hora de irradiar para poder disminuir la cantidad de radiación necesaria al disminuir el espesor de la mama.

Con esta prueba diagnóstica, se pretende encontrar microcalcificaciones mamarias. Las microcalcificaciones son acumulaciones de calcio pequeñas que aparecen en los ductos mamarios. La aparición de éstas está asociada al cáncer, dependiendo de su tamaño, localización, número, forma... habrá un mayor o menor grado de malignidad asociado.

Esas microcalcificaciones atenúan mucho la radiación, y en la imagen, aparecen con una gran intensidad. Un 30/50% de casos de cáncer de mama son reconocidos a través del hallazgo de microcalcificaciones.

Para un correcto control se tienen que analizar las siguientes características:

- **Tamaño:** Las superiores a 2 mm se clasifican de macrocalcificaciones y suelen ser benignas. Por debajo de los 2 mm se denominan microcalcificaciones. Cuanto más pequeñas y agrupadas más sospechosas son de malignidad.
- **Morfología:** Las calcificaciones malignas suelen ser heterogéneas en forma y tamaño, puntiagudas, anguladas, irregulares, en forma de coma, ramificadas y con forma de punto y raya. Las benignas suelen ser homogéneas, redondas y en ocasiones anulares y de centro claro.
- **Número:** Se considera que cuando hay tres o más calcificaciones menores de 1 mm en un área de 1x1 cm<sup>2</sup> de mamografía, existe sospecha de malignidad. Cuanto mayor es el número de calcificaciones en esa área, peor es el pronóstico.
- **Distribución:** Las calcificaciones distribuidas de forma segmentaria, no al azar, son sospechosas e indicativas de biopsia.
- **Variación** en el tiempo de las calcificaciones: Las calcificaciones malignas varían con el tiempo. La estabilidad de las calcificaciones durante año y medio a dos años, se consideran como benignas.

Debido a la escasa diferenciación entre los tejidos que se encuentran en la mama, la búsqueda de microcalcificaciones es complicada. En esta práctica intentaremos, con el uso de Matlab, diseñar un algoritmo capaz de encontrar dichas microcalcificaciones.

## **2.-Objetivos**

El objetivo principal es usar este problema práctico de búsqueda y análisis de microcalcificaciones para poner en práctica los conceptos estudiados en clase.

Dichos objetivos son:

- Iniciación al paquete Matlab e inmersión al *Toolbox Image Processing*.
- Manejo y visualización de una imagen digital:
  - Información DICOM.
  - Consideraciones respecto al tipo de datos, resolución y profundidad de pixel.
- Análisis, manejo y visualización del histograma. Posibilidades y limitaciones.
- Transformaciones globales sobre el histograma:
  - Mejora de contraste por ecualización.
  - Ajuste de ventana.
- Diseño de un programa en Matlab para la reproducción automática de comandos.
- Presentación de resultados científicos.

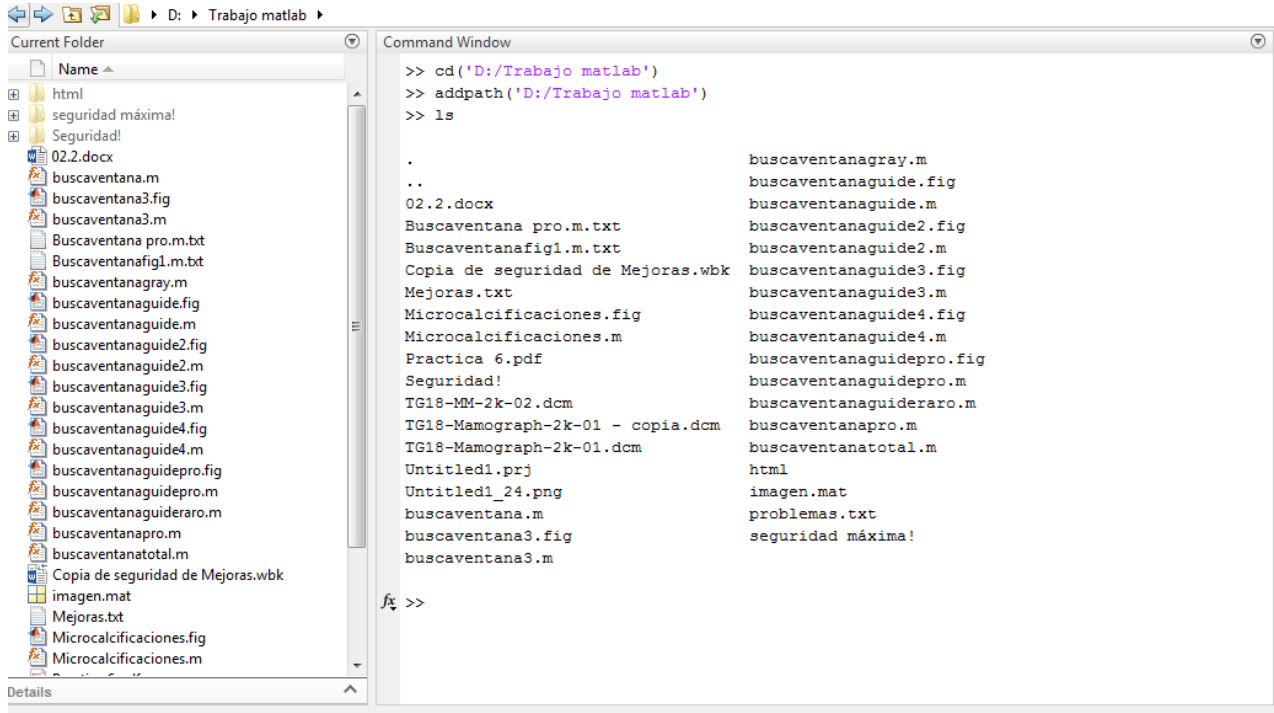
## **3.-Material**

- *Software* para el procesamiento de imágenes MATLAB Version 8.1.0.604 (R2013a); Image Processing Toolbox Version 8.2 (R2013a)
- Imágenes DICOM de mamografía digital, tomadas de la *Assessment of Display Performance for Medical Imaging Systems*, cedidas por la *American Association of Physicists in Medicine (AAPM) - Task Group 18*.
- Intel Celeron CPU P4600 @ 2.00GHz con 3 gigas de RAM. Sistema operativo de 64 bits, Windows 7, ServicePack 1.

## 4.-Desarrollo de la práctica

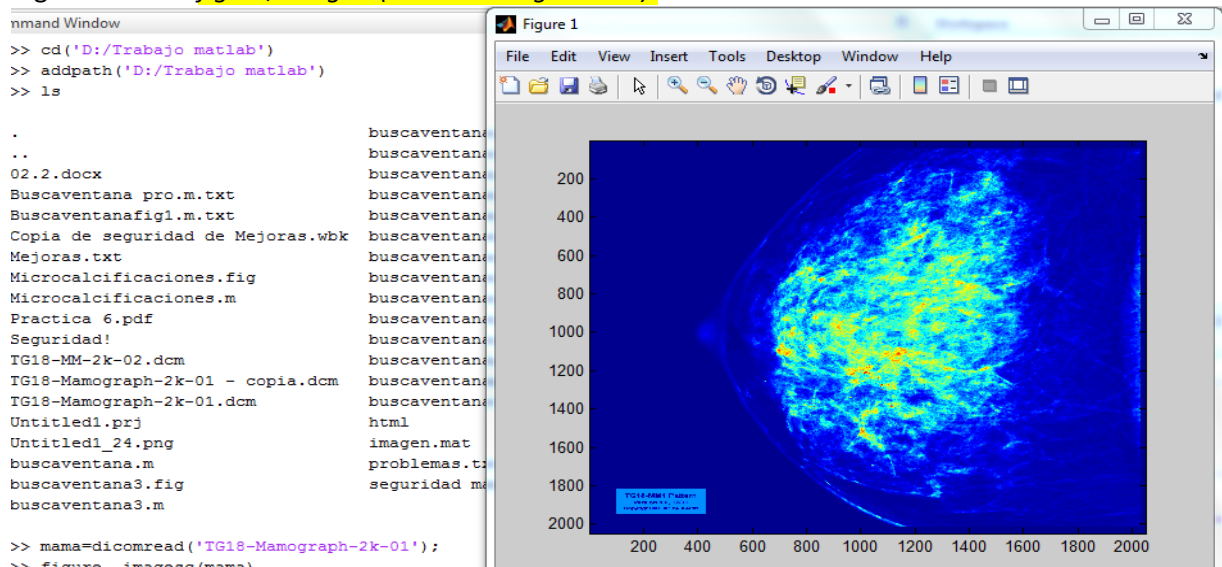
Antes de trabajar con la búsqueda de microcalcificaciones en sí, vamos a realizar algunas tareas para aprender a manejar Matlab.

Lo primero de todo es iniciar el programa, y, una vez iniciado, añadir el directorio y comprobar lo que hay en él. Usamos los comandos `cd`, `addpath`, y `ls`.



Vemos como al añadir el comando `cd`, Aparece en el *Current Folder* los archivos que tenemos situados en esa carpeta, así mismo, usando el comando `ls`, comprobamos con la lista que se observa todos los archivos que allí se encuentran.

A partir de aquí empezaremos a tratar con imágenes digitales. Las imágenes médicas, habitualmente, se guardan en un archivo DICOM. Vamos a pasar a leer una de las dos imágenes de las que disponemos con el comando `dicomread` (La guardaremos en una matriz que llamaremos `mama`), además de mostrar los metadatos que en el archivo DICOM hay con `dicominfo`, y de mostrar la imagen leída con `figure, imagesc(nombre imagen leída)`.



Con **dicominfo** nos vemos los metadatos (La lista no está completa en la imagen):

```
>> dicominfo('TG18-Mamograph-2k-01.dcm')
ans =

    Filename: [1x42 char]
    FileModDate: '04-may-2013 14:22:18'
    FileSize: 8390008
    Format: 'DICOM'
    FormatVersion: 3
    Width: 2048
    Height: 2048
    BitDepth: 16
    ColorType: 'grayscale'
    FileMetaInformationGroupLength: 208
    FileMetaInformationVersion: [2x1 uint8]
    MediaStorageSOPClassUID: '1.2.840.10008.5.1.4.1.1.7'
    MediaStorageSOPInstanceUID: [1x64 char]
    TransferSyntaxUID: '1.2.840.10008.1.2'
    ImplementationClassUID: '1.3.6.1.4.1.9590.100.1.0.100.4.0'
    ImplementationVersionName: 'MATLAB IPT 4.0'
    ImageType: 'ORIGINAL'
    InstanceCreationDate: '20021117'
    InstanceCreationTime: '192613'
    InstanceCreatorUID: '2.16.124.113543.6004.101.103'
    SOPClassUID: '1.2.840.10008.5.1.4.1.1.7'
    SOPInstanceUID: [1x64 char]
    StudyDate: '20021117'
    SeriesDate: '20021117'
```

A continuación, y con el fin de encontrar los valores máximos de la información relativa de la imagen usamos el comando **max**. Con esto podremos cambiar, si es necesario, el número de bits de la imagen para ahorrar memoria, siempre y cuando no se pierda información.

```
>> max(max(mama))

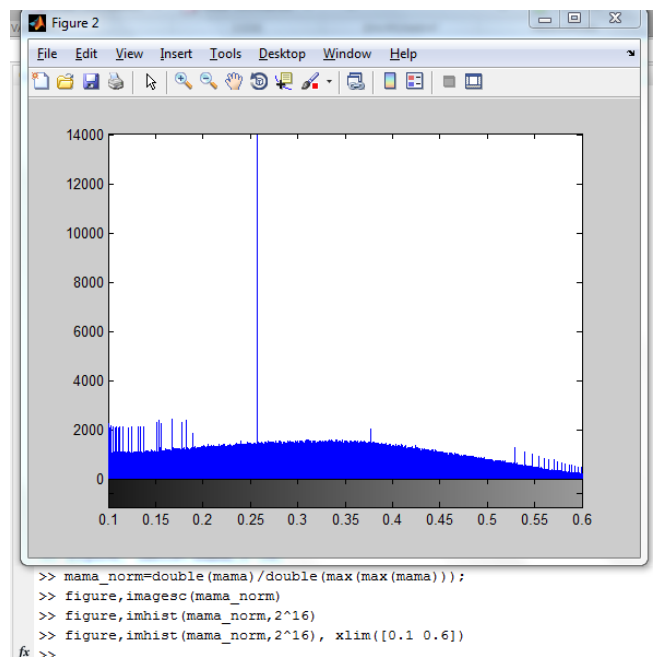
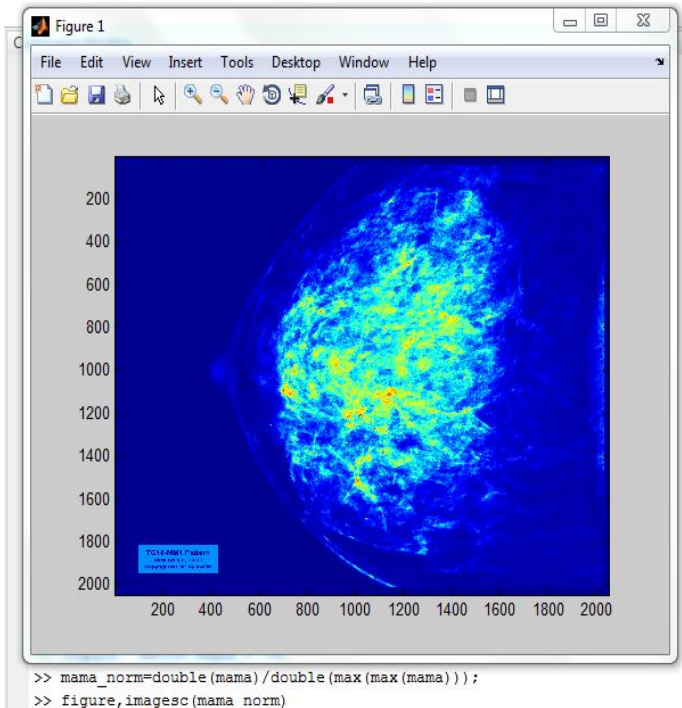
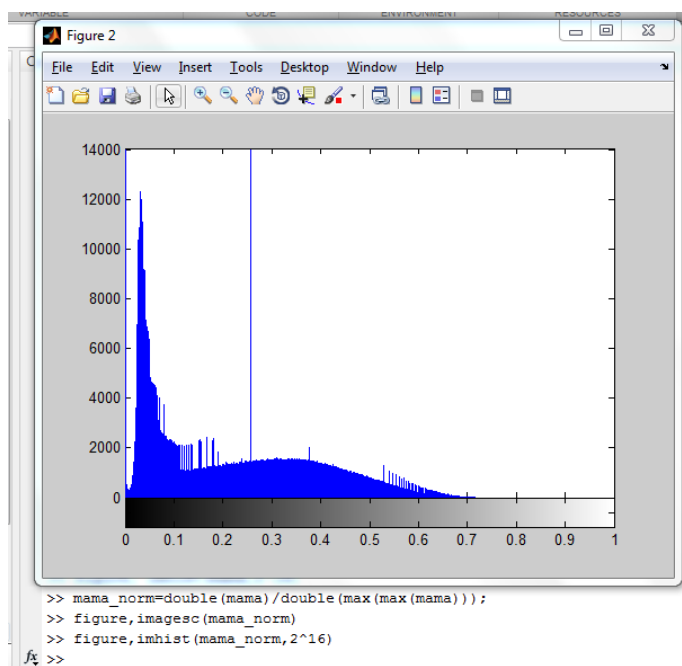
ans =

    3975

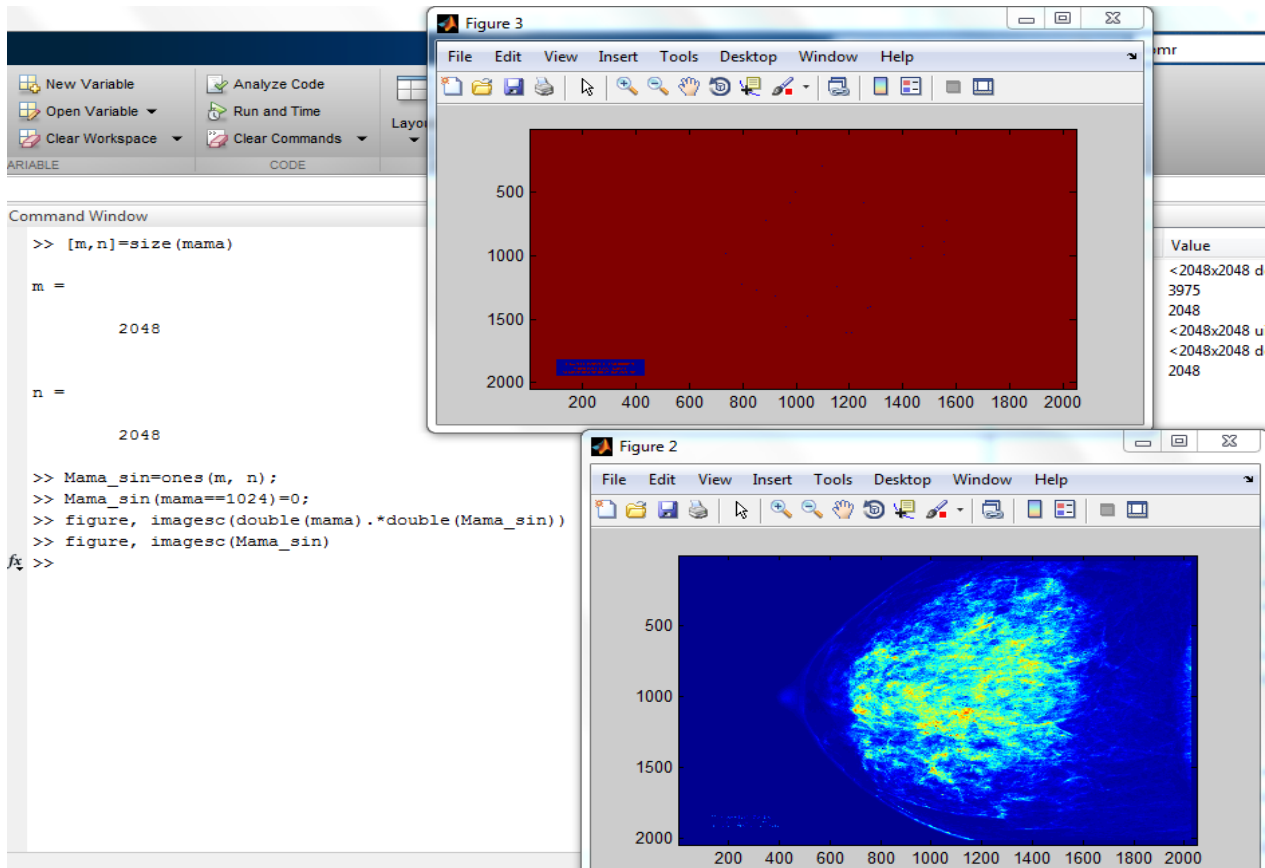
>>
```

Nuestra imagen posee como máximo un valor de gris igual a 3975, es decir, que con 12 bits, 4096 posibles valores de grises, sería suficiente, no obstante nuestra imagen tiene 16 bits.

A continuación, aprendemos a sacar el histograma de la imagen usando el comando `imhist`, además, podemos seleccionar el rango del histograma que queremos ver, y el número de bins que deseamos que tenga el histograma. `Figure, imhist(mama, número de bins), xlim([min max])`. Pero antes, normalizaremos la matriz (acción que facilita mucho la manipulación que ésta), y la visualizaremos. `mama_norm=double(mama)/double(max(max(mama)));`

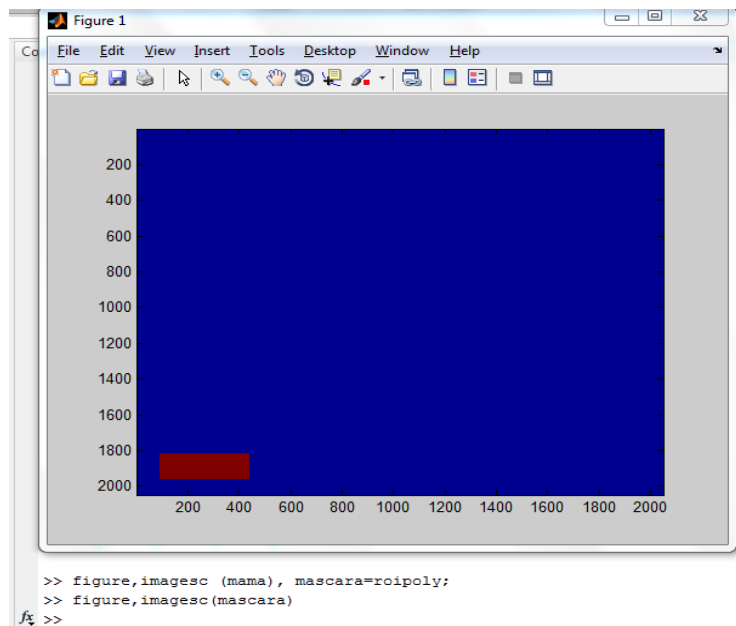


En el histograma vemos que hay un valor concreto que se repite excesivamente, corresponde con un artefacto de la imagen, con el cartel que en ésta aparece. A continuación lo eliminaremos generando una matriz *Mama\_sin* sólo con unos, en la cual, a los píxeles que poseen ese valor de intensidad (1024), vamos a darle el valor 0 (*Mama\_sin(mama==1024)=0;*), para luego, al multiplicar las dos matrices (*figure, imagesc(double(Mama).\*double(Mama\_sin));*), que se nos origine una matriz como la original pero donde los píxeles con valor 1024 tengan ahora el valor 0.

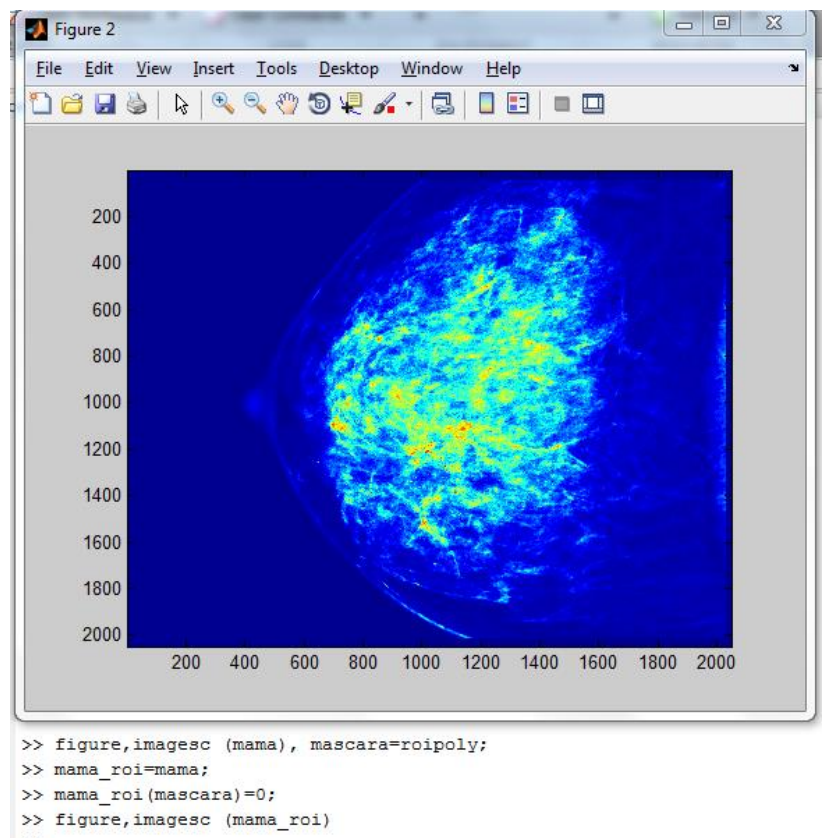


Sin embargo, este método no es muy eficaz porque, aunque eliminamos el cartel, es posible que dentro de la mama haya píxeles con el valor de intensidad 1024, por lo que estamos perdiendo información.

Vamos a probar a quitar el cartel de otra forma, a través de una máscara que seleccionaremos manualmente con el comando **roipoly**.

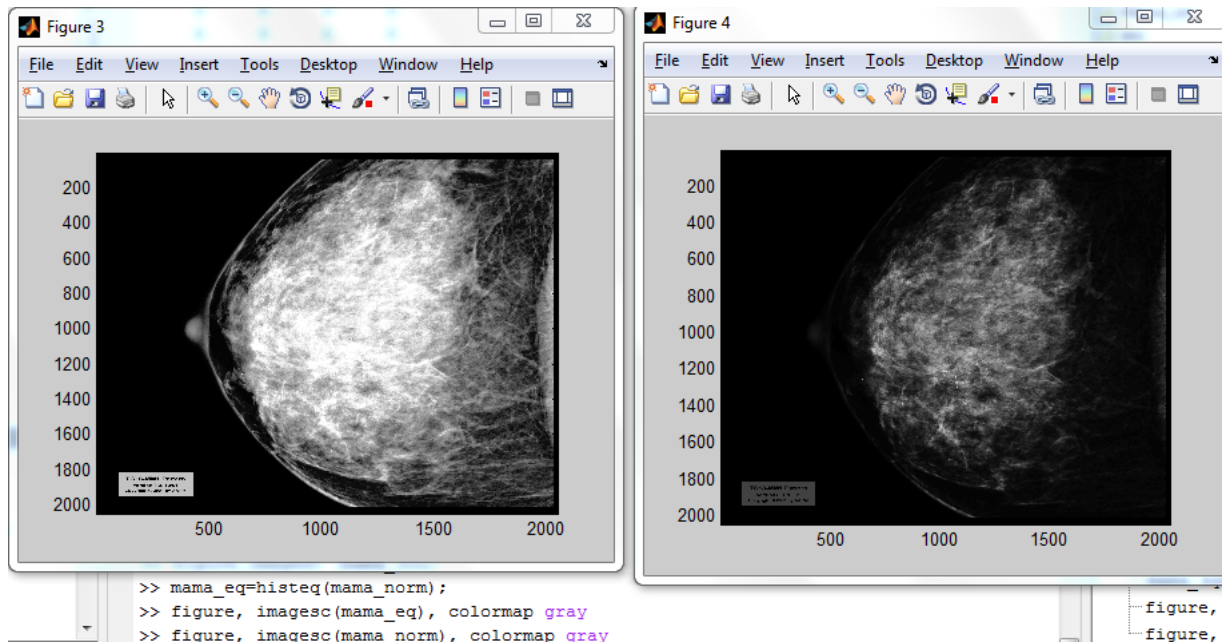


Una vez creamos la máscara la usamos sobre la imagen *mama\_roi* la cual la creamos para no perder la información original guardada en *mama*. Lo que conseguimos es lo que vemos en la imagen. El resultado final es mejor que en el caso anterior ya que no eliminamos píxeles con información relevante. No obstante, este método no sería muy eficiente si quisiera hacerse en muchas imágenes al mismo tiempo al automatizar la búsqueda de microcalcificaciones, por ejemplo.





Ahora vamos a ver cuál es el resultado de ecualizar la imagen, al ecualizar una imagen homogeneizamos el valor de los píxeles a lo largo de la profundidad de pixel, consiguiendo un mejor contraste. Usaremos el comando **histeq**, además, haremos que la imagen salga gris usando el comando **colormap gray**. Además, ponemos la imagen normalizada a la derecha para comparar.



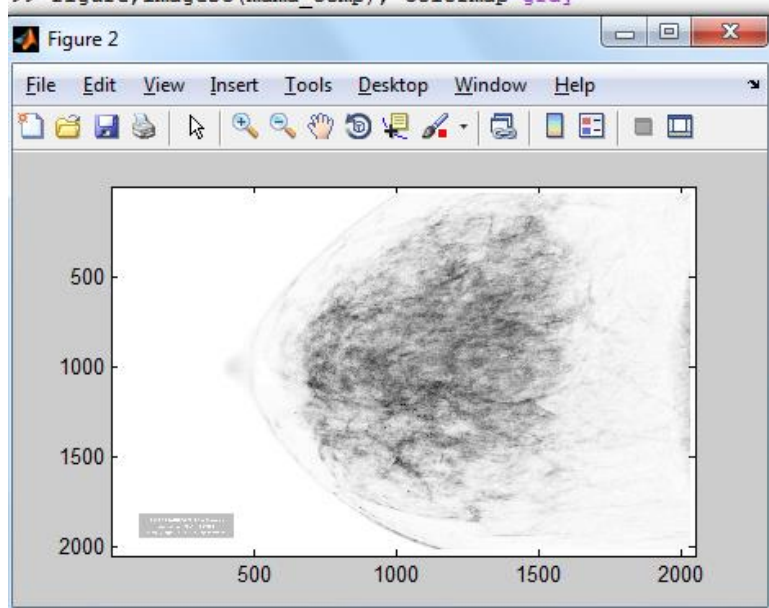
Por último, vamos a aprender a usar una ventana radiológica. Las microcalcificaciones normalmente tienen las intensidades más altas dentro de la imagen, sin embargo, nuestro ojo no es capaz de distinguirlas teniendo en la imagen toda una gama de niveles de grises, es por eso, que usando una ventana radiológica, podemos hacer que en la imagen solo aparezca un rango de niveles de gris, los que nosotros elijamos.

Usamos para ello el comando **imadjust**. Con este comando también podemos generar una imagen complementaria, es decir, invertida. Esto se consigue si en lugar de elegir un rango concreto, situamos el mínimo y el máximo de nuestra imagen de forma inversa, es decir, el máximo donde debería ir el mínimo y al revés (esto también se conseguiría con el comando **imcomplement**).

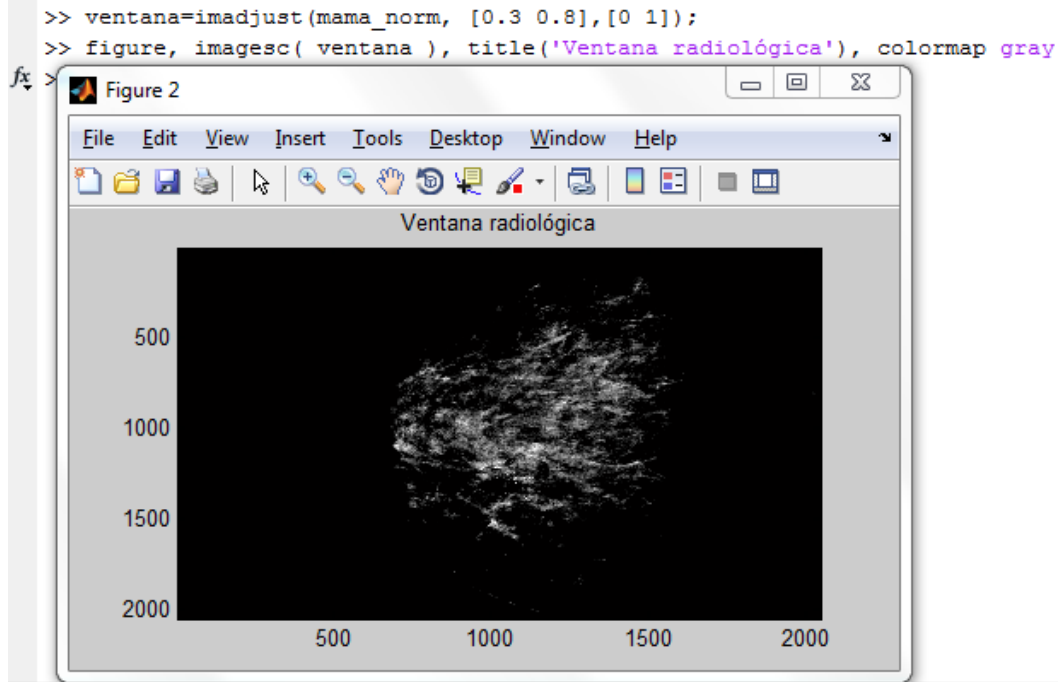
```

>> mama_comp=imadjust(mama_norm,[0 1],[1 0]);
>> figure,imagesc(mama_comp), colormap gray

```



La ventana radiológica quedaría tal que así (en este caso observamos los valores de gris entre 0.3 y 0.8):



## 5.-Desarrollo del programa

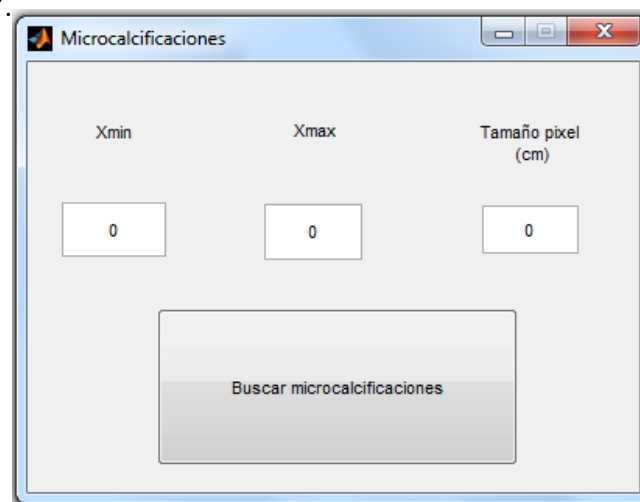
Matlab permite crear comandos propios a parte de los que ya trae de serie. En este caso, vamos a utilizar lo aprendido hasta ahora para crear un script capaz de encontrar microcalcificaciones en una imagen normalizada.

El script propuesto por el profesor se denomina buscaventana, y posee el siguiente código.

```
%Ejercicio para el ajuste automático de ventana.
function buscaventana(matriz,xmin,pixel)
%Parámetros
paso=0.01;
xmax=1;
margen=100;
%Búsqueda de ventana
while xmin<1
    ventana=imadjust(matriz,[xmin xmax],[0 1]);
    ventana_bw=im2bw(ventana);
    [x,y]=find(ventana_bw==1);
    x1=min(x);
    x2=max(x);
    y1=min(y);
    y2=max(y);
    distance=sqrt((x1-x2)^2+(y1-y2)^2)*pixel;
%Condición de malignidad considerando 1cm2
    if (distance<=sqrt(2))
        figure;imagesc(ventana);ylim([x1-margen x2+margen]);xlim([y1-margen y2+margen]);
        s=['Ventana Radiológica',num2str(xmin),' ',num2str(xmax); ];
        title(s);
        break
    end
    xmin=xmin+paso
end
```

Con él, podemos encontrar grupos de 3 o más microcalcificaciones en un área de  $\text{cm}^2$ . El comando buscará dichas agrupaciones desde el valor de intensidad (xmin) que deseemos hasta 1. Para ello sólo necesita como argumentos una matriz normalizada (matriz), el valor de xmin, y el tamaño de pixel (en centímetros).

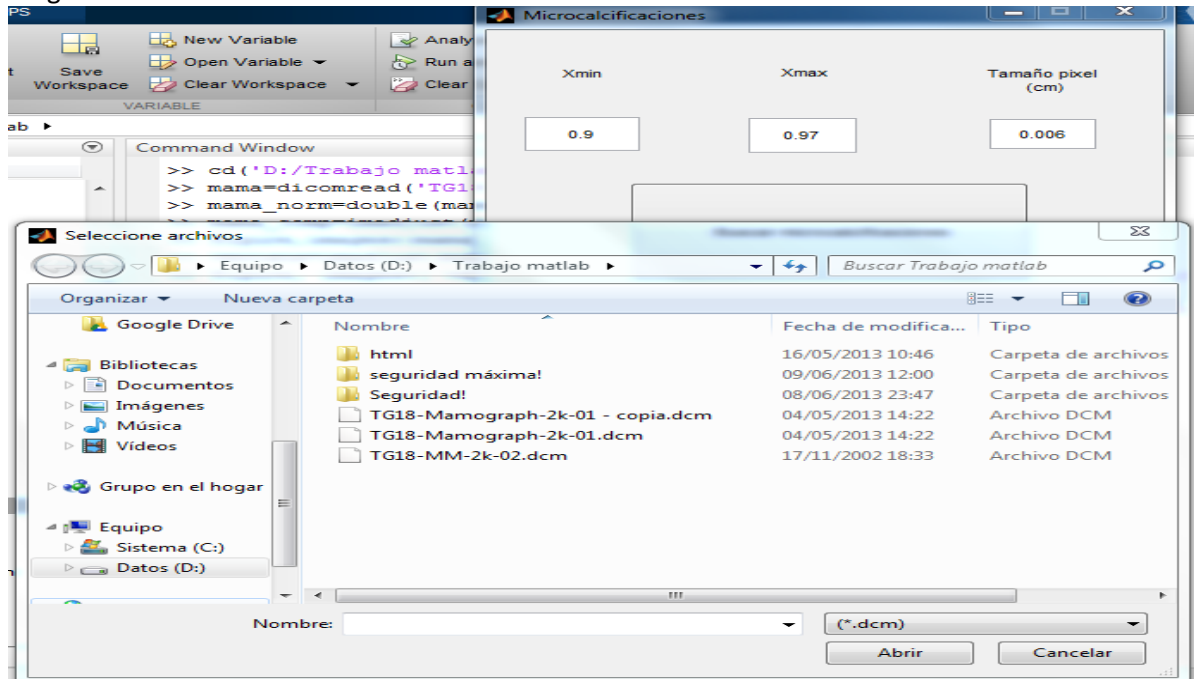
Como propuesta para mejorar dicho script, he generado una guide que he denominado “Microcalcificaciones”.



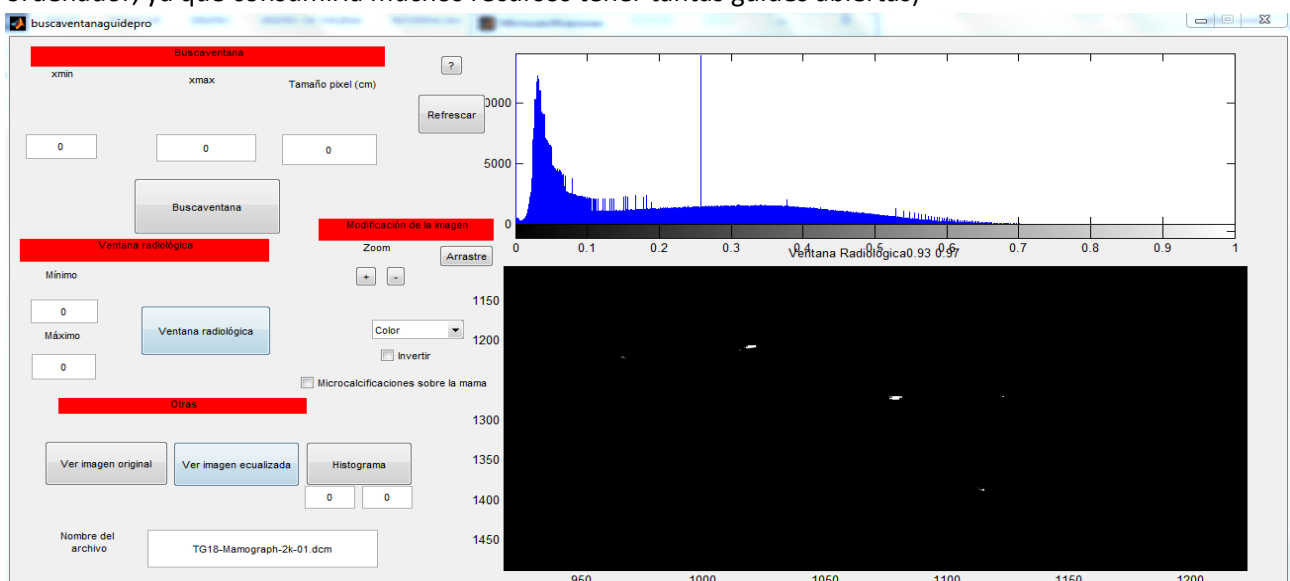
Microcalcificaciones es una guide creada para llamar a tantas subguides como archivos seleccionemos, me explico.

Como podemos observar Microcalcificaciones posee 3 cuadrados de texto editable, donde introduciremos los parámetros para que funcione el script buscaventana. En este caso, aparece un nuevo parámetro, que es xmax, con el cual podemos seleccionar hasta qué valor queremos que buscaventana busque microcalcificaciones sin que este busque hasta 1 automáticamente.

Una vez añadido xmin, xmax, y el tamaño de pixel, pulsamos el botón “Buscar microcalcificaciones”. Nos aparecerá ahora una ventana donde podremos seleccionar las imágenes sobre las que queremos correr el script buscaventana, podemos seleccionar un número indefinido de imágenes.



Cuando seleccionemos los archivos, Microcalcificaciones abrirá, para cada uno de ellos, la guide “buscaventanaguidepro”, en la cual aparecerá la imagen generada por el script buscaventana y el histograma de la imagen. Sin embargo, estas subguides se abrirán secuencialmente, es decir, hasta que no se cierre la buscaventanaguidepro que está abierta en ese momento, no se abrirá la siguiente correspondiente al siguiente archivo de todos los que seleccionamos (con esto evitamos saturar el ordenador, ya que consumiría muchos recursos tener tantas guides abiertas)



La subguide buscaventanaguidepro pretende hacer más versátil el manejo de las mamografías. Cuenta con multitud de opciones separadas en distintas zonas. Dichas opciones se detallan a continuación:

La primera zona es la que de Buscaventana. En ella seremos capaces de hacer correr el script buscaventana sobre la imagen ya abierta tal y como se hizo al utilizar la guide Microcalcificaciones. Al pulsar el botón, aparece la imagen en el eje inferior.

A la derecha de las opciones del buscaventana vemos dos botones “?” y “Refrescar”. Con el primero se nos abre un cuadro de texto con instrucciones sobre cómo usar buscaventanaguidepro (imagen1). Con el segundo reiniciamos la subguide con los valores que teníamos al empezar a utilizarla.

La segunda zona es la de Ventana radiológica. En ella podemos seleccionar un valor mínimo y máximo que usaremos para crear una ventana radiológica. La imagen de dicha ventana aparecerá en el eje inferior al pulsar el botón “Ventana radiológica” (imagen 2)

La zona modificación de la imagen nos da algunas opciones para interactuar con la imagen que se sitúe en el eje inferior. Podemos cambiar su color con las opciones que encontramos en el popup “color” (imagen 3), podemos hacer zoom “+” o quitar ese zoom “-”, podemos arrastrar la imagen si hicimos zoom en ella con el botón “arrastre”, o desactivar la opción de arrastrar pulsando de nuevo el botón. También podemos invertir la imagen situada en el eje inferior clicando en “invertir” (imagen 4), tenemos que tener en cuenta, que la imagen generada por buscaventana es la única que no se puede invertir, por el simple hecho, de que no nos proporcionaría una información relevante. Por último, clicando en “Microcalcificaciones sobre la mama” se nos abrirá una *figure* en la que veremos las microcalcificaciones (en rojo) sobre la imagen original de la mama (imagen 5).

Para acabar, en la zona otras tenemos 3 botones. El primero “Ver imagen original” (imagen 6) nos muestra la imagen original e la mama en el eje inferior. El segundo “Ver imagen ecualizada” (imagen 7) nos da la oportunidad de ver la imagen de la mama ecualizada también sobre el eje inferior. Y el tercero y último “Histograma” (imagen 8) nos da la posibilidad de generar un histograma en el eje superior, y de, introduciendo el mínimo y el máximo en los cuadros de texto de la izquierda y derecha respectivamente situados debajo del botón, observar sólo la parte del histograma que deseamos ver.

Para terminar, tenemos abajo del todo un cuadro de texto que nos informa del nombre de la imagen que estamos visualizando.

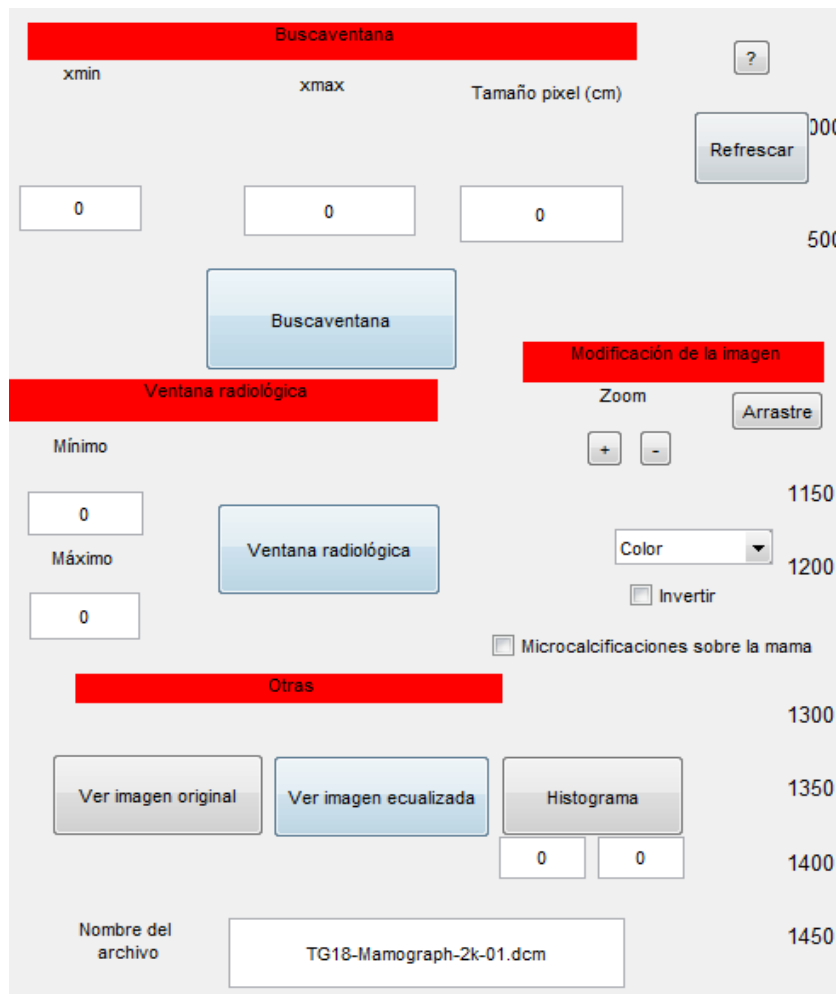


Imagen 1.  
Instrucciones

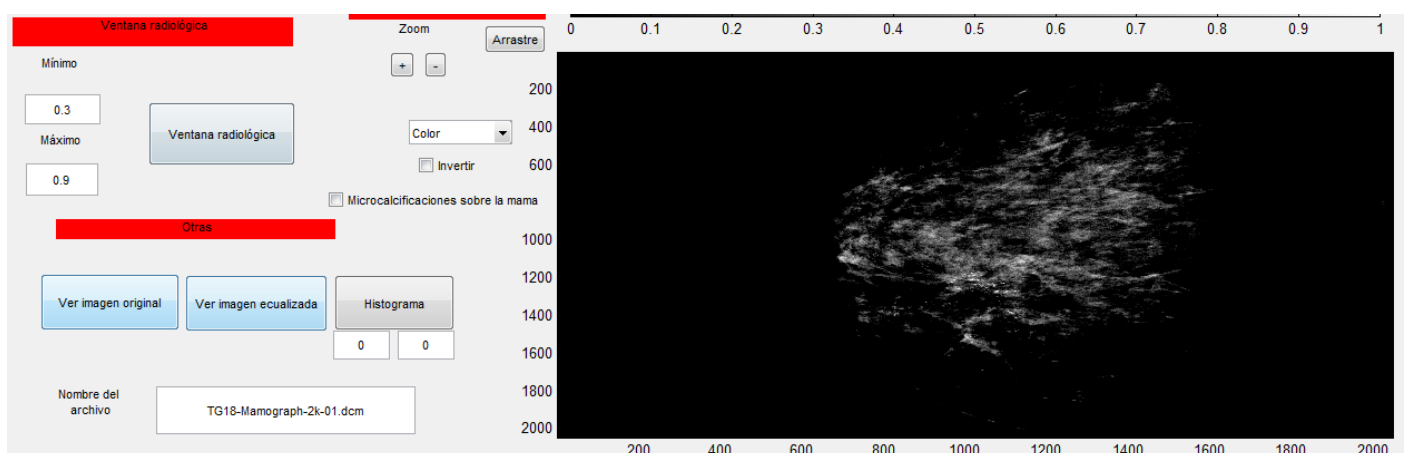
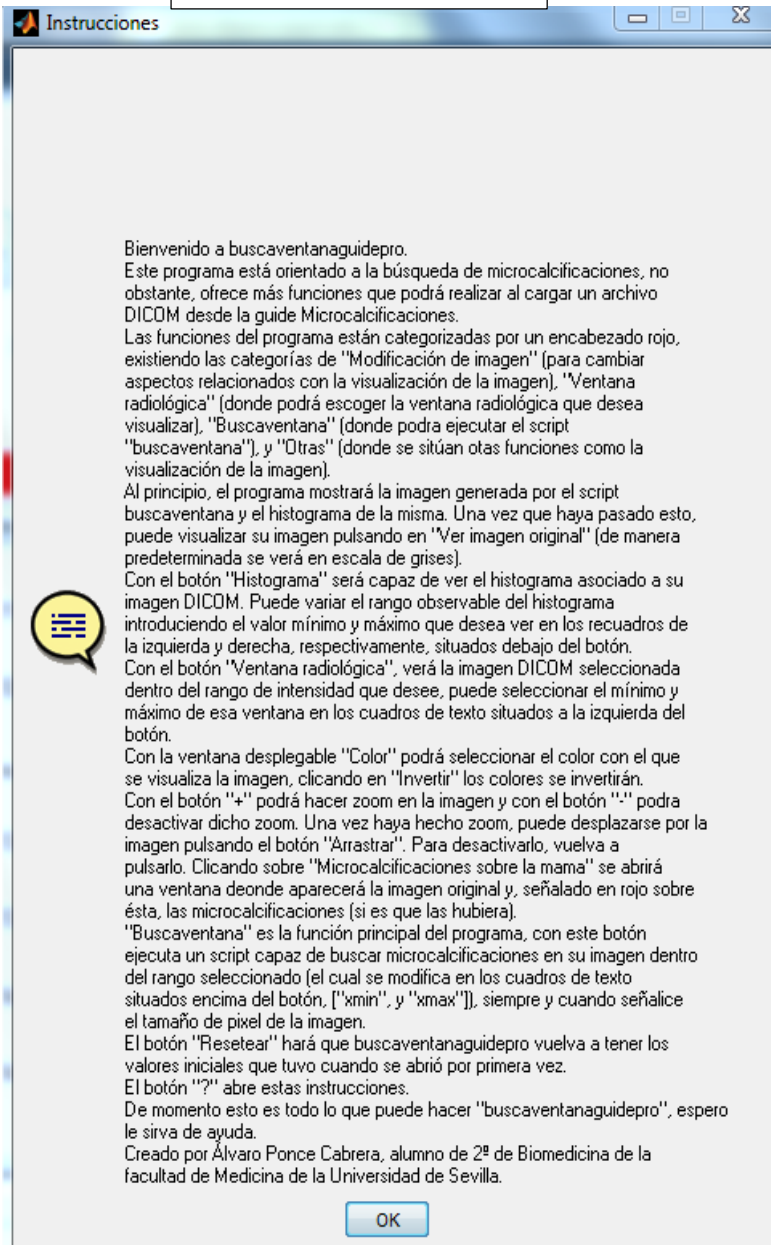


Imagen 2.  
Ejemplo de una ventana radiológica entre los valores 0.3 y 0.9



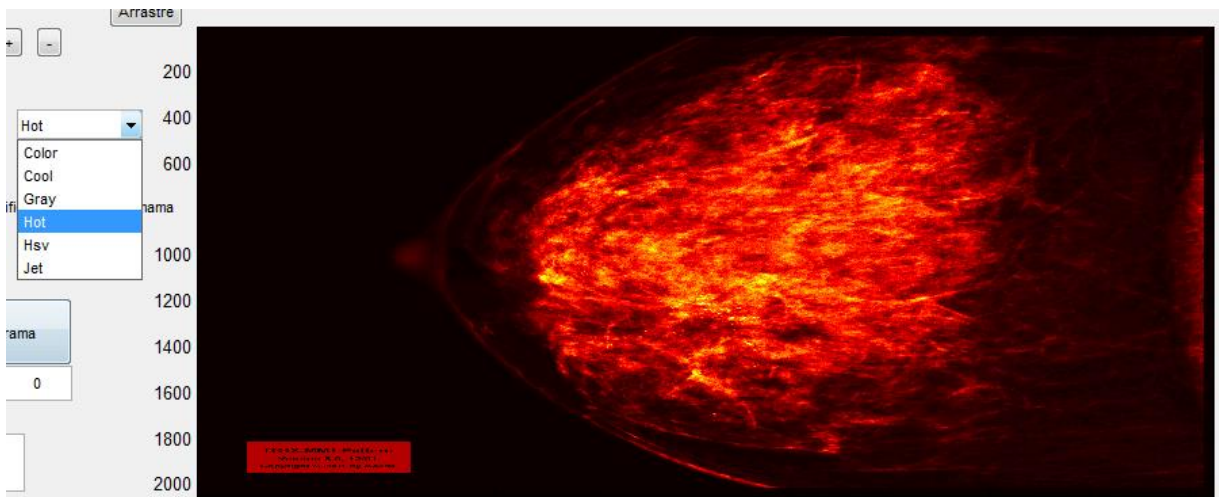


Imagen 3.  
Usamos el color "Hot" como ejemplo.

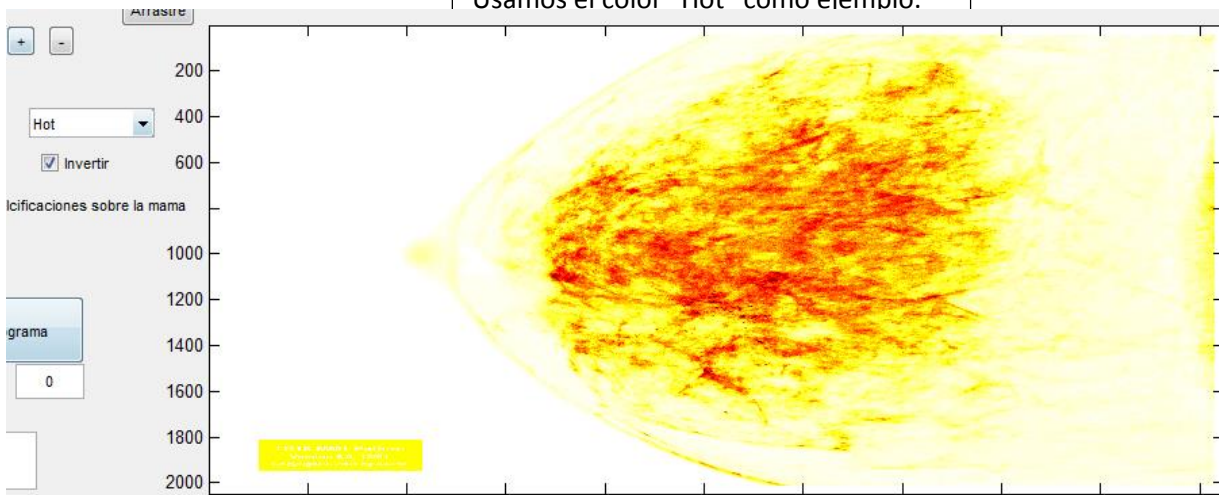


Imagen 4.  
Imagen invertida a partir de la anterior.

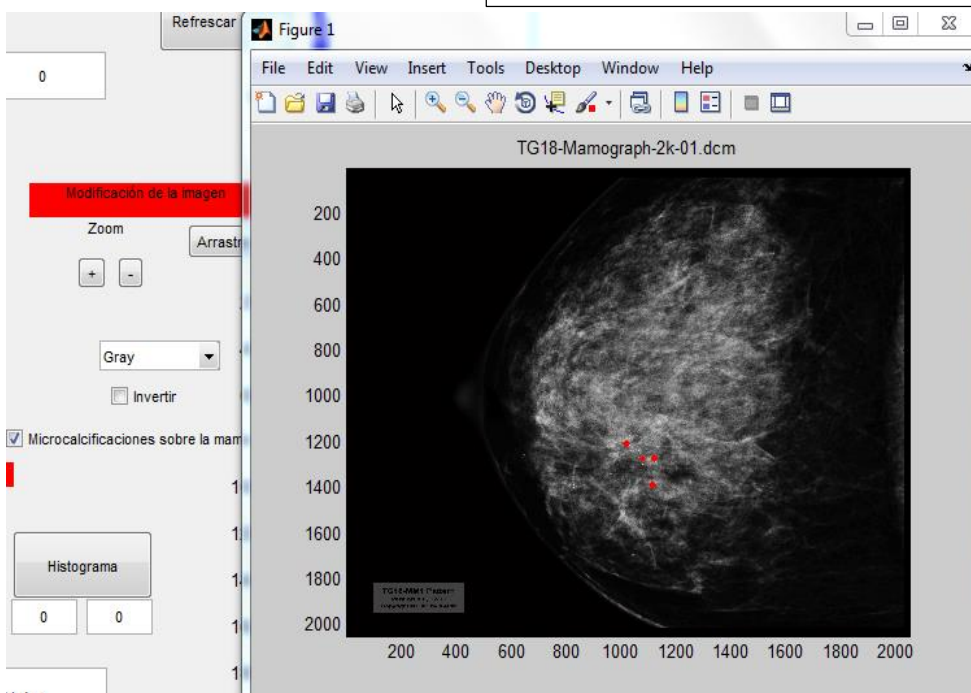


Imagen 5.  
Microcalcificaciones  
señaladas sobre la imagen  
original de la mama.

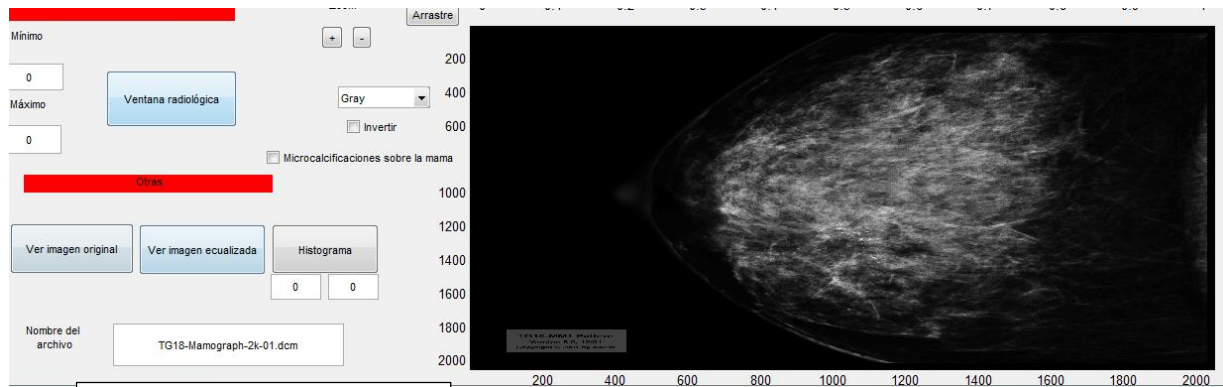


Imagen 6.  
Imagen original de la mama

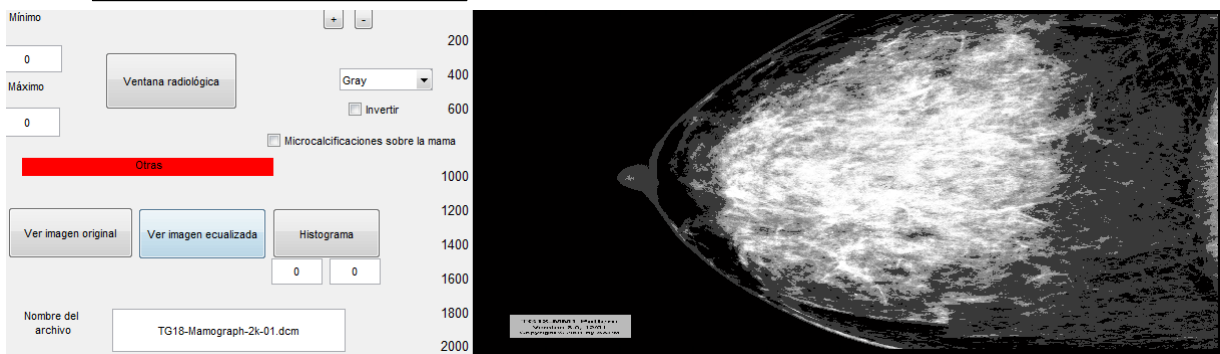


Imagen 7.  
Imagen ecualizada de la mama

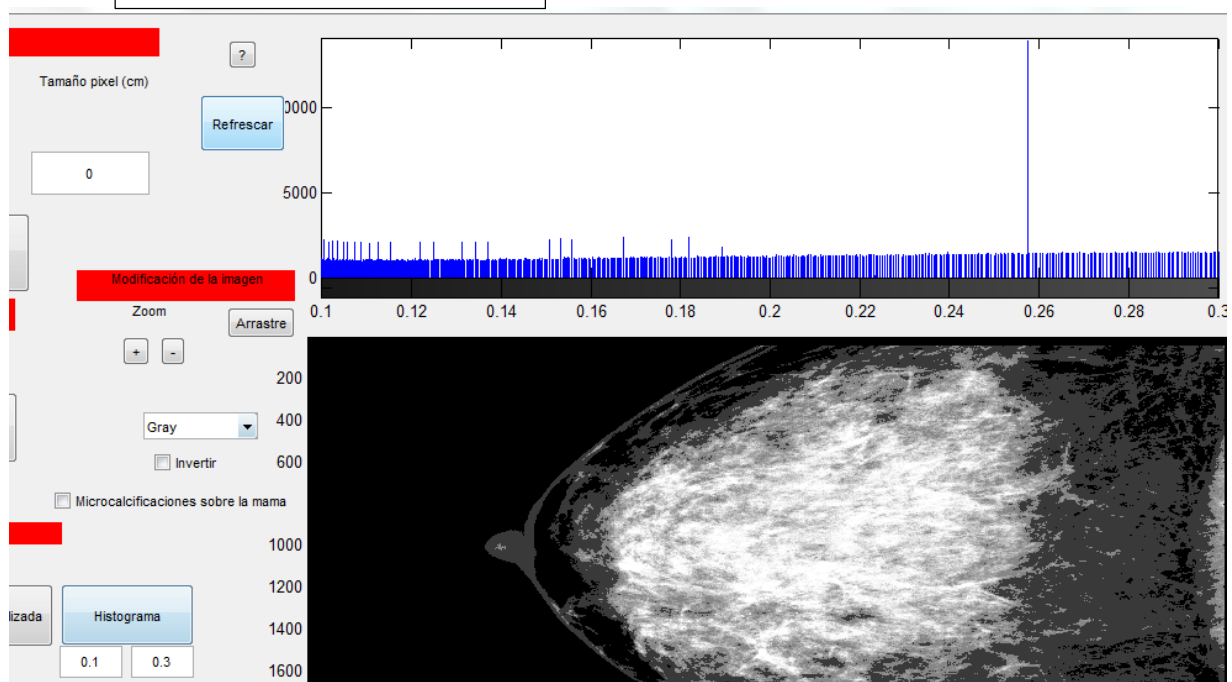


Imagen 8.  
Histograma visto en el intervalo 0.1 y 0.3

Para las imágenes usadas como ejemplo se ha usado el mismo archivo DICOM que se empleó en la práctica.



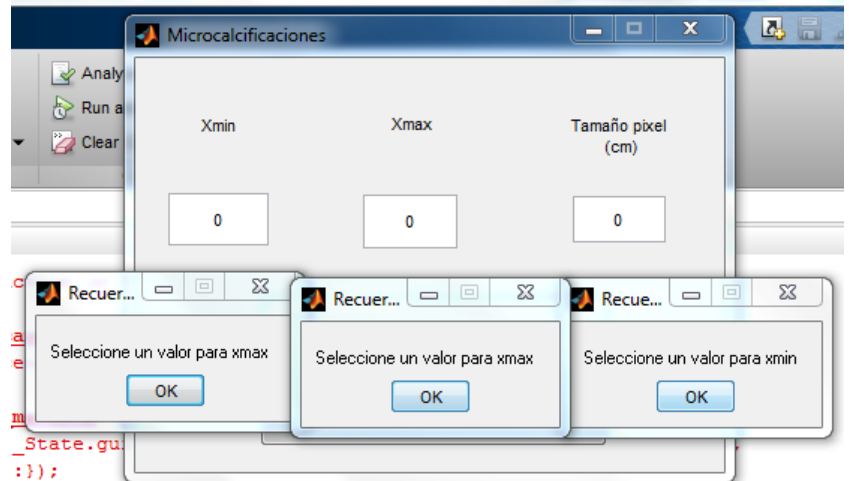
### Notificaciones:

Tanto Microcalcificaciones como buscaventanaguidepro tienen la capacidad de avisar al usuario de que está cometiendo ciertos errores, e indicarle que debe hacer para solucionarlos. Algunas veces los errores no impiden que el programa siga funcionando, sin embargo con éstos el resultado no será el esperado, por eso la necesidad de las notificaciones de error. A continuación se detalla cuales son:

#### Microcalcificaciones:

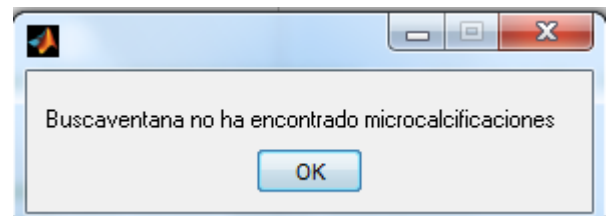
Cuando no se ha añadido un valor a alguno de los 3 cuadros de texto editables (Xmin, Xmax, y tamaño de pixel), aparece una notificación avisando de cual de esos valores es el que el usuario ha olvidado añadir.

En la imagen se ve un caso para el cual el usuario no ha añadido ninguno de los 3 valores, por eso saltan las 3 ventanas de notificación.

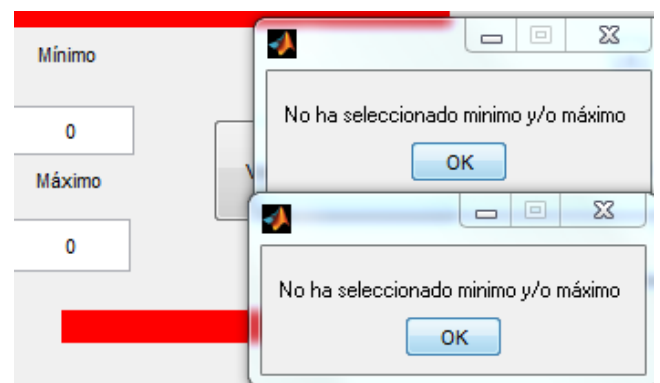


#### Buscaventanaguidepro:

La primera notificación se da cuando, al ejecutar el script buscaventana (ya sea desde el inicio de buscaventanaguidepro, usando el botón “Buscaventana”, o usando el botón “Refrescar”) no se encuentran microcalcificaciones. Es una notificación de aviso, ya que al no encontrar microcalcificaciones agrupadas en menos de 1 cm<sup>2</sup> no aparecerá imagen.



La siguiente y última notificación aparece al pulsar el botón “Ventana”. Si no se ha añadido valor alguno a los recuadros mínimo o máximo, aparece el siguiente mensaje (aparece dos veces porque no se ha añadido ninguno de los dos valores):



### **Errores y posibles mejoras.**

Antes de poner el código de ambas guides, y debido al extenso tamaño que ocupan, veo conveniente hablar de los errores y de las posibles mejoras de los programas.

Uno de los errores más evidentes está en el botón “invertir”, ya que no es posible invertir la imagen ecualizada (difícil de seleccionar al tener este botón tantas opciones) ni la imagen generada por buscaventana. Una mejora sería dar la posibilidad de hacer estas dos inversiones.

La traducción de la guide en varios idiomas podría ser también una gran ventaja. Crear una opción con un popup que, al seleccionarla haga set sobre todos los textos modificándolos podría ser la manera de hacerlo.

Otro problema de buscaventanaguidepro es que, a priori, da pocas posibilidades de comparación de imágenes, es decir, da muchas opciones a realizar con una sola imagen, no obstante, (y aunque con utilizar otro buscaventanaguidepro al lado con otra imagen bastaría) sería interesante que se diese la posibilidad de comparar varias imágenes (al menos 2) en la misma guide.

Algo a tener en cuenta es el uso de “Microcalcificaciones sobre la mama” ya que, si buscaventana no encuentra microcalcificaciones, este botón realmente no servirá para nada, y sin embargo, aunque no haya microcalcificaciones, el botón representa ciertos puntos rojos sobre la imagen.

A parte de éstas, son muchas las mejoras que pueden hacerse en ambas guides, pero no será hasta que se le dé uso, cuando empiecen a surgir ideas de lo que realmente podría añadirse para hacer de estas unas guides mejores. Sobre todo dependerá del uso médico de éstas ya que, aunque prácticamente sólo se está discutiendo lo que lleva de programación ambas guides, también deberíamos de tener en cuenta los detalles médicos que todo esto lleva detrás, por ejemplo, al principio comentamos que dependiendo de la forma de las microcalcificaciones la malignidad es mayor o peor. El reconocimiento de estas formas sería una gran mejora para el programa, e incluso, se podría calcular el porcentaje de malignidad según la forma, y el número de microcalcificaciones y de agrupaciones de éstas, no obstante, esto es algo, que, de momento, está fuera de mi alcance.

Habría que considerar también hasta qué punto es válido el procesamiento de la imagen que hemos llevado a cabo, y hasta qué punto es posible su uso en cualquier tipo de imagen o no, es decir, cuáles serían sus limitaciones a la hora de considerarlo como un programa de uso universal de imágenes DICOM.

## Códigos

Una vez explicado el uso de las guides, escribo a continuación el código de ambas, explicando en cada punto el uso y el por qué de lo que está escrito.

### **Código de Microcalcificaciones:**

```
function varargout = Microcalcificaciones(varargin)

gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @Microcalcificaciones_OpeningFcn, ...
                  'gui_OutputFcn',  @Microcalcificaciones_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end

function Microcalcificaciones_OpeningFcn(hObject, eventdata, handles,
varargin)

handles.output = hObject;

guidata(hObject, handles);

function varargout = Microcalcificaciones_OutputFcn(hObject, eventdata,
handles)

varargout{1} = handles.output;

%CUADRO DE TEXTO "XMIN".
%En él se coge(get) y se guarda en handles el valor de xmin

function Txmin_Callback(hObject, eventdata, handles)

handles.xmin=str2double(get(hObject, 'String'));
guidata(hObject, handles);
% --- Executes during object creation, after setting all properties.
function Txmin_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
%CUADRO DE TEXTO "XMAX".
%En él se coge(get) y se guarda en handles el valor de xmax

function Txmax_Callback(hObject, eventdata, handles)

handles.xmax=str2double(get(hObject, 'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function Txmax_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%CUADRO DE TEXTO "PIXEL".
%En él se coge(get) y se guarda en handles el tamaño de pixel

function Tpixel_Callback(hObject, eventdata, handles)

handles.pixel=str2double(get(hObject, 'String'));
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function Tpixel_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%Botón que inicia la función principal de la guide

% --- Executes on button press in subguide.
function subguide_Callback(hObject, eventdata, handles)
%Recogida de los valores de los cuadros de texto para la posterior
%notificación si ese valor es 0.
pixel=str2double(get(handles.Tpixel, 'string'));
xmin=str2double(get(handles.Txmin, 'string'));
xmax=str2double(get(handles.Txmax, 'string'));
%Uigetfile es usado para seleccionar los archivos con los que se desea
%trabajar
uiget=uigetfile('.dcm','Seleccione archivos','Multiselect','on')
%Notificación de error si no se añaden valores.
%Si el usuario ha olvidado añadir alguno de los parámetros salta un mensaje
%informando del error.
if pixel==0
    msgbox ('Seleccione un tamaño de pixel','Recuerde')
end
if xmin==0
    msgbox ('Seleccione un valor para xmin','Recuerde')
end
if xmax==0
    msgbox ('Seleccione un valor para xmax','Recuerde')
end
```

```
%Bucle for.
%Es usado para abrir para cada uno de los archivos buscaventanaguidepro
for i=1:length(uiget)
    %También hace la lectura de cada uno de los archivos
    mamai=dicomread(uiget{i});
    %los argumentos de buscaventanaguidepro son los datos que se pasarán
    %desde Microcalcificaciones a buscaventanaguidepro.

    buscaventanaguidepro(mamai,handles.pixel,handles.xmin,handles.xmax,uiget{i})
    %Con este comando hago que se abra buscaventanaguidepro del siguiente
    %archivo únicamente cuando cierro el buscaventanaguidepro que está abierto.
    uiwait

end
guidata(hObject, handles);
```

### Código de buscaventanaguidepro:

```
function varargout = buscaventanaguidepro(varargin)
% BUSCAVENTANAGUIDEPRO MATLAB code for buscaventanaguidepro.fig
%     BUSCAVENTANAGUIDEPRO, by itself, creates a new BUSCAVENTANAGUIDEPRO
% or raises the existing
%     singleton*.
%
%     H = BUSCAVENTANAGUIDEPRO returns the handle to a new
% BUSCAVENTANAGUIDEPRO or the handle to
%     the existing singleton*.
%
%     BUSCAVENTANAGUIDEPRO('CALLBACK',hObject,eventData,handles,...) calls
% the local
%     function named CALLBACK in BUSCAVENTANAGUIDEPRO.M with the given
% input arguments.
%
%     BUSCAVENTANAGUIDEPRO('Property','Value',...) creates a new
% BUSCAVENTANAGUIDEPRO or raises the
%     existing singleton*. Starting from the left, property value pairs
% are
%     applied to the GUI before buscaventanaguidepro_OpeningFcn gets
% called. An
%     unrecognized property name or invalid value makes property
% application
%     stop. All inputs are passed to buscaventanaguidepro_OpeningFcn via
% varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help buscaventanaguidepro

% Last Modified by GUIDE v2.5 09-Jun-2013 19:20:19

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @buscaventanaguidepro_OpeningFcn, ...
```

```

        'gui_OutputFcn', @buscaventanaguidepro_OutputFcn, ...
        'gui_LayoutFcn', [] , ...
        'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

%--- Outputs from this function are returned to the command line.

% --- Executes just before buscaventanaguidepro is made visible.
function buscaventanaguidepro_OpeningFcn(hObject, eventdata, handles,
varargin)

%Parámetros de la guide madre son guardados en handles.
mamai=varargin{1};
handles.mamai=mamai
handles.pixel=varargin{2};
handles.xmin=varargin{3};
handles.xmax=varargin{4};
filename=varargin{5};
handles.filename=filename

%Se usa set para poner el nombre de la imagen correspondiente en el cuadro
%de texto de la parte de abajo de la guide (llamado "edit10")

set(handles.edit10,'String',filename)
%A continuación se realizan algunas operaciones y se guarda en handles los
%resultados para hacer más cómodo el trabajo luego.

%Matriz doble
handles.double=double(mamai);
%Matriz normalizada
handles.norm=(handles.double)/(max(max(handles.double)))
%Los datos serán guardados con guidata al final de esta function.

% Choose default command line output for buscaventanaguidepro
handles.output = hObject;

%A continuación añadimos el script buscaventana usando los datos cargados
%anteriormente. Este script presenta una mejora frente al original, y es
%que, no busca microcalcificaciones desde un valor determinado y hasta 1,
%sino que es posible añadir un rango de búsqueda, un mínimo, y un máximo.

%Parámetros
pixel=handles.pixel
xmin=handles.xmin
xmax=handles.xmax
paso=0.01;
margen=100;

```

```
%Búsqueda de ventana
while xmin<xmax
    ventana=imadjust(handles.norm,[xmin xmax],[0 1]);
    ventana_bw=im2bw(ventana);
    [x,y]=find(ventana_bw==1);
    x1=min(x);
    x2=max(x);
    y1=min(y);
    y2=max(y);
    distance=sqrt((x1-x2)^2+(y1-y2)^2)*pixel;
%Condición de malignidad considerando 1cm2

    if (distance<=sqrt(2))
        %Si se encontrasen microcalcificaciones, éstas se cargan sobre el
        %eje1 (axes1) y además se pone en color gris.
        axes(handles.axes1);imagesc(ventana);ylim([x1-margen
x2+margen]);xlim([y1-margen y2+margen]);
        s=['Ventana Radiológica',num2str(xmin),' ',num2str(xmax); ];
        colormap gray
        title(s);
        break
    end
    xmin=xmin+paso
%Si no se encontrasen microcalcificaciones sale el mensaje siguiente como
%un aviso.
end
if(distance>sqrt(2))
    msgbox('Buscaventana no ha encontrado microcalcificaciones')
end

%TERMINA BUSCAVENTANA
%Guardamos en handles las coordenadas de las microcalcificaciones para más
%adelante
handles.x1=x
handles.y1=y
%Representamos el histograma de la imagen en el eje2 (axes2)
axes(handles.axes2);imhist(handles.norm,2^16);
%Guardamos los handles con guidata
guidata(hObject, handles);

function varargout = buscaventanaguidepro_OutputFcn(hObject, eventdata,
handles)
% varargout    cell array for returning output args (see VARARGOUT);
% hObject     handle to figure
% eventdata   reserved - to be defined in a future version of MATLAB
% handles     structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;
%

%Estos 3 son textos estáticos de la guide
% --- Executes during object creation, after setting all properties.
function text1_CreateFcn(hObject, eventdata, handles)
```

```
% --- Executes during object creation, after setting all properties.
function text2_CreateFcn(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function text3_CreateFcn(hObject, eventdata, handles)

%CON LOS 3 BOTONES SIGUIENTES RECOGEMOS LOS VALORES DE LOS CUADROS DE TEXTO
%Valores que luego se guardarán en handles y se usarán luego para ejecutar
%de nuevo buscaventana si así se quisiera.
%Recogemos el calor de xmax
function edit1_Callback(hObject, eventdata, handles)

handles.xmax_1=str2double(get(hObject, 'String'));

guidata(hObject, handles);
% Hints: get(hObject,'String') returns contents of edit1 as text
%        str2double(get(hObject,'String')) returns contents of edit1 as a
double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%CON ESTO RECOGEMOS EL VALOR DEL PIXEL

function edit2_Callback(hObject, eventdata, handles)
% hObject      handle to edit2 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
handles.pixel_1=str2double(get(hObject, 'String'));

guidata(hObject, handles);
% Hints: get(hObject,'String') returns contents of edit2 as text
%        str2double(get(hObject,'String')) returns contents of edit2 as a
double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```



```
%Recogemos y guardamos en handles el valor de xmin
function edit3_Callback(hObject, eventdata, handles)

handles.xmin_1=str2double(get(hObject, 'String'));

guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%Cuadro de texto editable donde se introduce el nombre de la imagen
function edit10_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit10_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%Botón con el que se ejecuta el script buscaventana.
%Una vez se empieza a trabajar con buscaventanaguidepro, se puede
%ejecutar para la imagen cargada el script buscaventana en otro rango
%distinto al que previamente se eligió.
% --- Executes on button press in buscaventana.
function buscaventana_Callback(hObject, eventdata, handles)

%Parámetros
paso=0.01;
margen=100;
xmin=handles.xmin_1
xmax=handles.xmax_1
pixel=handles.pixel_1

%Búsqueda de ventana
while xmin<xmax
    ventana=imadjust(handles.norm,[xmin xmax],[0 1]);
    ventana_bw=im2bw(ventana);
    [x,y]=find(ventana_bw==1);
    x1=min(x);
```

```

x2=max(x);
y1=min(y);
y2=max(y);
distance=sqrt((x1-x2)^2+(y1-y2)^2)*pixel;
%Condición de malignidad considerando 1cm2

if (distance<=sqrt(2))
    %Se coloca la imagen en color gris sobre el eje 1 si se encuentra
    %grupos de 3 o más microcalcificaciones en menos de 1 cm2

    axes(handles.axes1);imagesc(ventana);ylim([x1-margen
x2+margen]);xlim([y1-margen y2+margen]);
    s=['Ventana Radiológica',num2str(xmin),' ',num2str(xmax); ];
    colormap gray
    title(s);
    break
end
xmin=xmin+paso

end
%Se guardan en handles las coordenadas donde están las microcalcificaciones
handles.x1=x
handles.y1=y
%guardar los handles
guidata(hObject,handles);
%mensaje de aviso por si no se encuentran microcalcificaciones
if(distance>sqrt(2))
    msgbox('Buscaventana no ha encontrado microcalcificaciones')
end

% --- Executes during object creation, after setting all properties.
function buscaventana_CreateFcn(hObject, eventdata, handles)

%Función correspondiente al eje1
% --- Executes during object creation, after setting all properties.
function axes1_CreateFcn(hObject, eventdata, handles)

%BOTÓN PARA VISUALIZAR LA IMAGEN ORIGINAL SIN MODIFICACIONES.
function imagen_Callback(hObject, eventdata, handles)
%vista de la imagen original en el axes

axes(handles.axes1);imagesc(handles.mamai)

colormap gray
% --- Executes during object creation, after setting all properties.
function imagen_CreateFcn(hObject, eventdata, handles)

```

```
%TEXTOS ESTÁTICOS DE LA VENTANA RADIOLOGICA
```

```
% --- Executes during object creation, after setting all properties.
function text7_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
% --- Executes during object creation, after setting all properties.
function text8_CreateFcn(hObject, eventdata, handles)
% hObject    handle to text8 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called
```

```
%A PARTIR DE AQUI SE AÑADE LA FUNCIÓN DE VENTANA RADIOLÓGICA
%AQUI SE RECOGE EL MÍNIMO DEL CUADRO DE TEXTO EDITABLE
```

```
function MINIMO_Callback(hObject, eventdata, handles)
```

```
handles.minimo=str2double(get(hObject, 'String'));
```

```
guidata(hObject, handles);
```

```
% --- Executes during object creation, after setting all properties.
function MINIMO_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
```

```
%Aquí se recoge el máximo del cuadro de texto editable
```

```
function MAXIMO_Callback(hObject, eventdata, handles)
```

```
handles.maximo=str2double(get(hObject, 'String'));
```

```
guidata(hObject, handles);
```

```
% --- Executes during object creation, after setting all properties.
function MAXIMO_CreateFcn(hObject, eventdata, handles)
```

```
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end
```

```
%Con este botón se inicia la función la ventana radiológica y se sitúa la
%imagen en el eje1
```

```
% --- Executes on button press in ventana.
```

```
function ventana_Callback(hObject, eventdata, handles)
```

```
%Si no se ha añadido valor alguno para minimo o maximo se abre un mensaje de
%error
```

```

if handles.minimo==0
    msgbox('No ha seleccionado minimo y/o máximo')
end
if handles.maximo==0
    msgbox('No ha seleccionado minimo y/o máximo')
end
    %Si se han añadido los valores en los cuadros de texto, se hace la
    %ventana radiológica y se establece ésta en el eje1
if handles.minimo+handles.maximo~=0
    ventana=imadjust(handles.norm,[handles.minimo handles.maximo],[0 1]);
    axes(handles.axes1);imagesc(ventana);
end

% --- Executes during object creation, after setting all properties.
function ventana_CreateFcn(hObject, eventdata, handles)

%Botón de histograma
% --- Executes on button press in histograma.
function histograma_Callback(hObject, eventdata, handles)
% hObject      handle to histograma (see GCBO)
%Se obtiene y se guarda en handles los valores minimo y maximo del rango
%del histograma que deseamos ver
handles.v1=str2double(get(handles.edit14, 'string'));
handles.v2=str2double(get(handles.edit15, 'string'));
%si no se han añadido valores sale el histograma de la función

    if handles.v1+ handles.v2==0
        axes(handles.axes2);imhist(handles.norm,2^16);
    else
        %Si se han añadido valores sale el histograma, pero solo para los
        %valores seleccionados
        axes(handles.axes2);imhist(handles.norm,2^16); xlim([handles.v1
handles.v2]);
    end
    guidata(hObject, handles);

% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

%Botón "?".
%Cuando se pulsa aparecen las instrucciones con el consiguiente mensaje.
% --- Executes on button press in instrucciones.
function instrucciones_Callback(hObject, eventdata, handles)

helpdlg({'Bienvenido a buscaventanaguidepro.' 'Este programa está orientado
a la búsqueda de microcalcificaciones, no obstante, ofrece más funciones que
podrá realizar al cargar un archivo DICOM desde la guide
Microcalcificaciones.' 'Las funciones del programa están categorizadas por
un encabezado rojo, existiendo las categorías de "Modificación de imagen"
(para cambiar aspectos relacionados con la visualización de la imagen),
"Ventana radiológica" (donde podrá escoger la ventana radiológica que desea
visualizar), "Buscaventana" (donde podrá ejecutar el script "buscaventana"),
y "Otras" (donde se sitúan otras funciones como la visualización de la
imagen).' 'Al principio, el programa mostrará la imagen generada por el
script buscaventana y el histograma de la misma. Una vez que haya pasado
esto, puede visualizar su imagen pulsando en "Ver imagen original" (de

```

```
manera predeterminada se verá en escala de grises).' 'Con el botón
"Histograma" será capaz de ver el histograma asociado a su imagen DICOM.
Puede variar el rango observable del histograma introduciendo el valor
mínimo y máximo que desea ver en los recuadros de la izquierda y derecha,
respectivamente, situados debajo del botón.' 'Con el botón "Ventana
radiológica", verá la imagen DICOM seleccionada dentro del rango de
intensidad que desee, puede seleccionar el mínimo y máximo de esa ventana en
los cuadros de texto situados a la izquierda del botón.' 'Con la ventana
desplegable "Color" podrá seleccionar el color con el que se visualiza la
imagen, clicando en "Invertir" los colores se invertirán. Con el botón "+"
podrá hacer zoom en la imagen y con el botón "-" podrá desactivar dicho
zoom. Una vez haya hecho zoom, puede desplazarse por la imagen pulsando el
botón "Arrastrar". Para desactivarlo, vuelva a pulsarlo. Clicando sobre
"Microcalcificaciones sobre la mama" se abrirá una ventana donde aparecerá
la imagen original y, señalado en rojo sobre ésta, las microcalcificaciones
(si es que las hubiera).' '"Buscaventana" es la función principal del
programa, con este botón ejecuta un script capaz de buscar
microcalcificaciones en su imagen dentro del rango seleccionado (el cual se
modifica en los cuadros de texto situados encima del botón, ["xmin", y
"xmax"]), siempre y cuando señalice el tamaño de pixel de la imagen.' 'El
botón "Resetear" hará que buscaventanaguidepro vuelva a tener los valores
iniciales que tuvo cuando se abrió por primera vez.' 'El botón "?" abre
estas instrucciones.' 'De momento esto es todo lo que puede hacer
"buscaventanaguidepro", espero le sirva de ayuda.' 'Creado por Álvaro Ponce
Cabrera, alumno de 2º de Biomedicina de la facultad de Medicina de la
Universidad de Sevilla.'}, 'Instrucciones')
```

```
%BOTON DE ZOOM -
% --- Executes on button press in pushbutton10.
function pushbutton10_Callback(hObject, eventdata, handles)

axes(handles.axes1)
zoom out
zoom off
%BOTÓN DE ZOOM +
% --- Executes on button press in pushbutton9.
function pushbutton9_Callback(hObject, eventdata, handles)

axes(handles.axes1)
zoom on

%Botón que activa y desactiva el arrastre. Lo activa para a==1 y lo
%desactiva para a==0, o lo que es lo mismo, para cuando el botón está
%clicado o no.
% --- Executes on button press in arrastre.
function arrastre_Callback(hObject, eventdata, handles)
a=get(hObject, 'value')
if a==1

axes(handles.axes1); pan
end

    if a==0

axes(handles.axes1); pan
end
```

```
%Con este botón seleccionamos el color que queremos darle a la imagen
% --- Executes on selection change in color.
function color_Callback(hObject, eventdata, handles)

v=get(hObject,'value')
%Según la opción elegida se aactivará uno u otro
switch v
    case 2
        axes(handles.axes1);colormap cool;
    case 3

        axes(handles.axes1);colormap gray;
    case 4
        axes(handles.axes1);colormap hot;
    case 5
        axes(handles.axes1);colormap hsv;
    case 6
        axes(handles.axes1);colormap jet;

end
% --- Executes during object creation, after setting all properties.
function color_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%Edit 14 y 15 son los cuadros de texto editable donde se ponen el valor
%minimo y maximo del rango del histograma que desea observarse.

function edit14_Callback(hObject, eventdata, handles)
%
% --- Executes during object creation, after setting all properties.
function edit14_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit15_Callback(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function edit15_CreateFcn(hObject, eventdata, handles)

if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

%BOTÓN INVERTIR (no invierte la imagen ecualizada)
% --- Executes on button press in invertir.
function invertir_Callback(hObject, eventdata, handles)
```

```

a=get(hObject,'Value')

%Debido a un error handles.maximo y handles.minimo no son reconocidos si no
%añado valores al cuadro de texto (cuando de por sí el valor debería ser 0)
%así que me veo obligado a hacer un get de ese valor inicial==0.
handles.minimo1=str2double(get(handles.MINIMO, 'string'));
handles.maximo1=str2double(get(handles.MAXIMO, 'string'));

v1=handles.minimo1
v2=handles.maximo1

%CUando se clicca en el botón a==1, entonces, dependiendo de la situación
%del resto de valores se invertirá una cosa u otra, coincidiendo siempre
%con aquello que esté situado en el eje
if a==1

    %invertido imagen original, se hace cuando los valores maximo y
    %minimo de la ventana radiologica son 0
    if v1+v2==0
        matriz1=imcomplement(handles.norm);
        axes(handles.axes1);imagesc(matriz1);
    end
    %invertido ventana radiologica
    %Si los valores de la ventana radiologica no son 0 se invierte
    %esta.
    if v1+v2~=0
        ventanal=imadjust(handles.norm,[v1 v2],[v2 v1]);
        axes(handles.axes1);imagesc(ventanal);
    end
    %Cuando se desclica invertir...
else
    %Si los valores para la ventana siguen a 0, aparece la imagen
    %original.
    if v1+v2==0
        matriz2=imadjust(handles.norm,[0 1],[0 1]);
        axes(handles.axes1);imagesc(matriz2);
        %si los valores de la ventana no son 0, aparece la ventana
        %radiológica correspondiente
    else
        ventana3=imadjust(handles.norm,[v1 v2],[0 1]);
        axes(handles.axes1);imagesc(ventana3);
    end
end

% Hint: get(hObject,'Value') returns toggle state of invertir

%Botón que representa la imagen ecualizada en el eje1
% --- Executes on button press in ecualizada.
function ecualizada_Callback(hObject, eventdata, handles)

ecualizada=histeq(handles.mamai);
axes(handles.axes1);imagesc(ecualizada);

```

%Botón refrescar. Resetea la guide poniendo todos los valores de los textos %editables a 0. Además corre de nuevo el script buscaventana con los datos %que se tenían al abrirse ésta a partir de Microcalcificaciones.

```
% --- Executes on button press in refrescar.
function refrescar_Callback(hObject, eventdata, handles)
%Valores a 0
set(handles.edit3,'string',0);
set(handles.edit1,'string',0);
set(handles.edit2,'string',0);
set(handles.edit14,'string',0);
set(handles.edit15,'string',0);
set(handles.MINIMO,'string',0);
set(handles.MAXIMO,'string',0);
%Imitación del inicio de la GUIDE

%Parámetros
pixel=handles.pixel
xmin=handles.xmin
xmax=handles.xmax
paso=0.01;
margen=100;

%Búsqueda de ventana
while xmin<xmax
    ventana=imadjust(handles.norm,[xmin xmax],[0 1]);
    ventana_bw=im2bw(ventana);
    [x,y]=find(ventana_bw==1);
    x1=min(x);
    x2=max(x);
    y1=min(y);
    y2=max(y);
    distance=sqrt((x1-x2)^2+(y1-y2)^2)*pixel;
%Condición de malignidad considerando 1cm2

    if (distance<=sqrt(2))
        axes(handles.axes1);imagesc(ventana);ylim([x1-margen
x2+margen]);xlim([y1-margen y2+margen]);

        colormap gray

        break
    end
    xmin=xmin+paso
end
if (distance>sqrt(2))
    msgbox('Buscaventana no ha encontrado microcalcificaciones')
end

%TERMINA BUSCAVENTANA
axes(handles.axes2);imhist(handles.norm,2^16);
handles.x1=x
handles.y1=y
guidata(hObject, handles);
%añadir que se haga buscaventana tal como se mando (proyecto subgui)

%Con este botón somos capaces de ver las microcalcificaciones (marcadas en
```



```
%rojo) sobre la imagen de la mama original.

% --- Executes on button press in checkbox2.
function checkbox2_Callback(hObject, eventdata, handles)
a=get(hObject,'Value')
%Cuando se clicca en la imagen aparece un figure en el que se carga la
%imagen en color gris, se le da el nombre de la imagen como título.
if a==1
figure, imagesc(handles.mamai);    colormap gray
s=[handles.filename]
title(s);
%Con el comando scatter pintamos de rojo los pixeles correspondientes a las
%coordenadas x e y donde se encuentran las microcalcificaciones dentro de
%la imagen.
hold on
scatter(handles.y1,handles.x1,4,'r','fill');
hold off
end
%Cuando se desclica se cierra la figure.
if a==0
    close figure 1;
end
```