

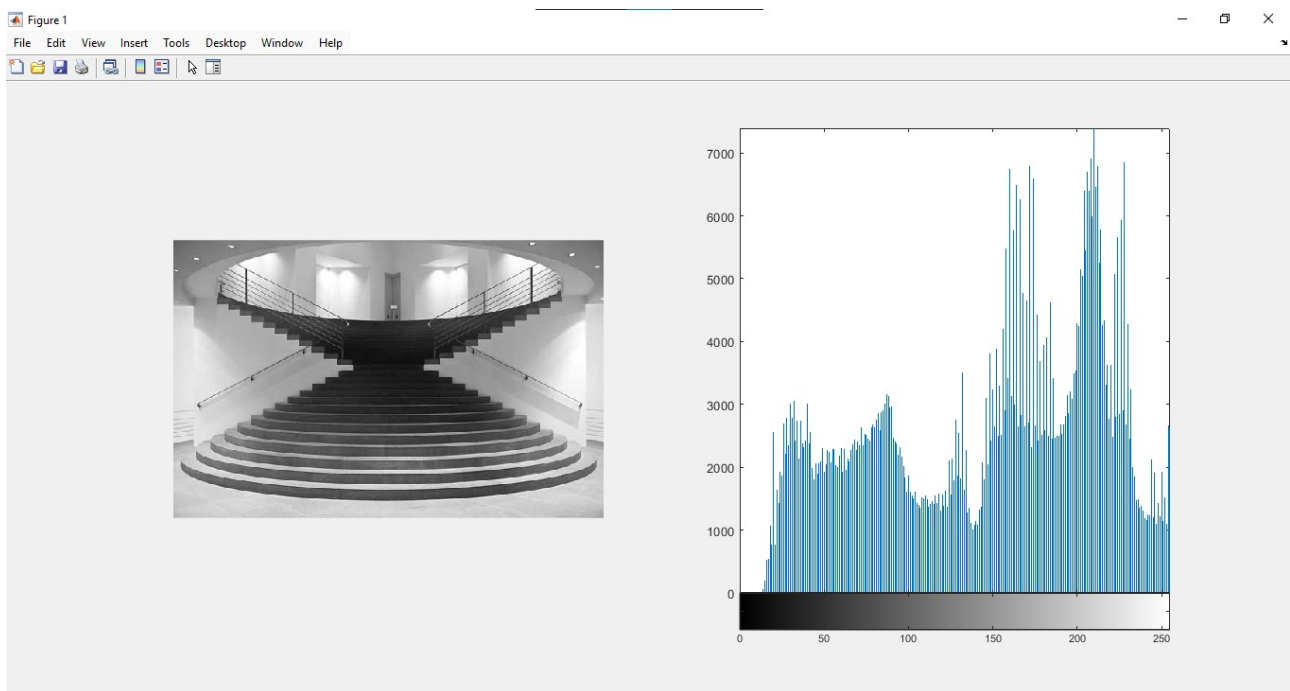
# Practica 3

## 6. Igualación del histograma

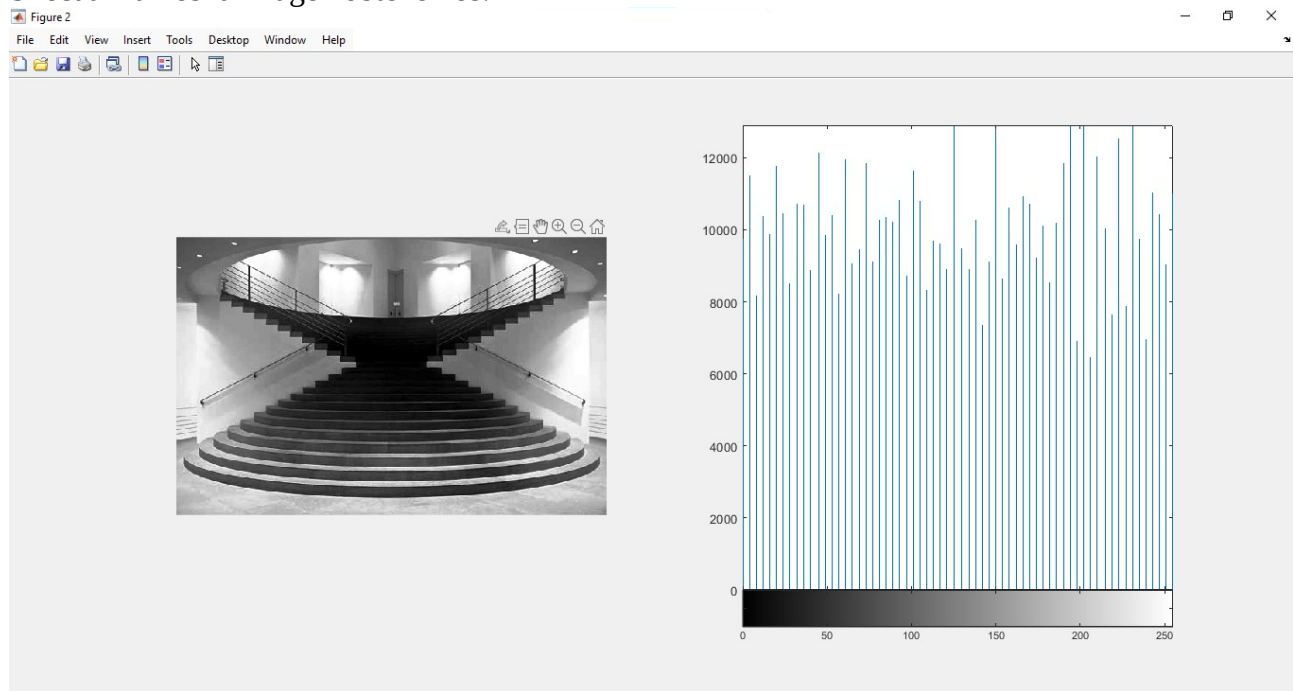
### A.

```
I=imread(' ../Imagenes/escalera2.jpg ');
figure
subplot(1,2,1)
imshow(I);
subplot(1,2,2)
%Mostramos el histograma de la imagen
imhist(I);
figure
subplot(1,2,1)
%Aumenta el contraste de una imagen ecualizando el histograma.
%Solo funciona con imagenes de tonos de grises
%Devuelve la imagen y el vector de la transformacion
imshow(histeq(I))
subplot(1,2,2)
imhist(histeq(I))
figure, imshow(I-histeq(I))
```

La imagen original y su histograma se ven así:

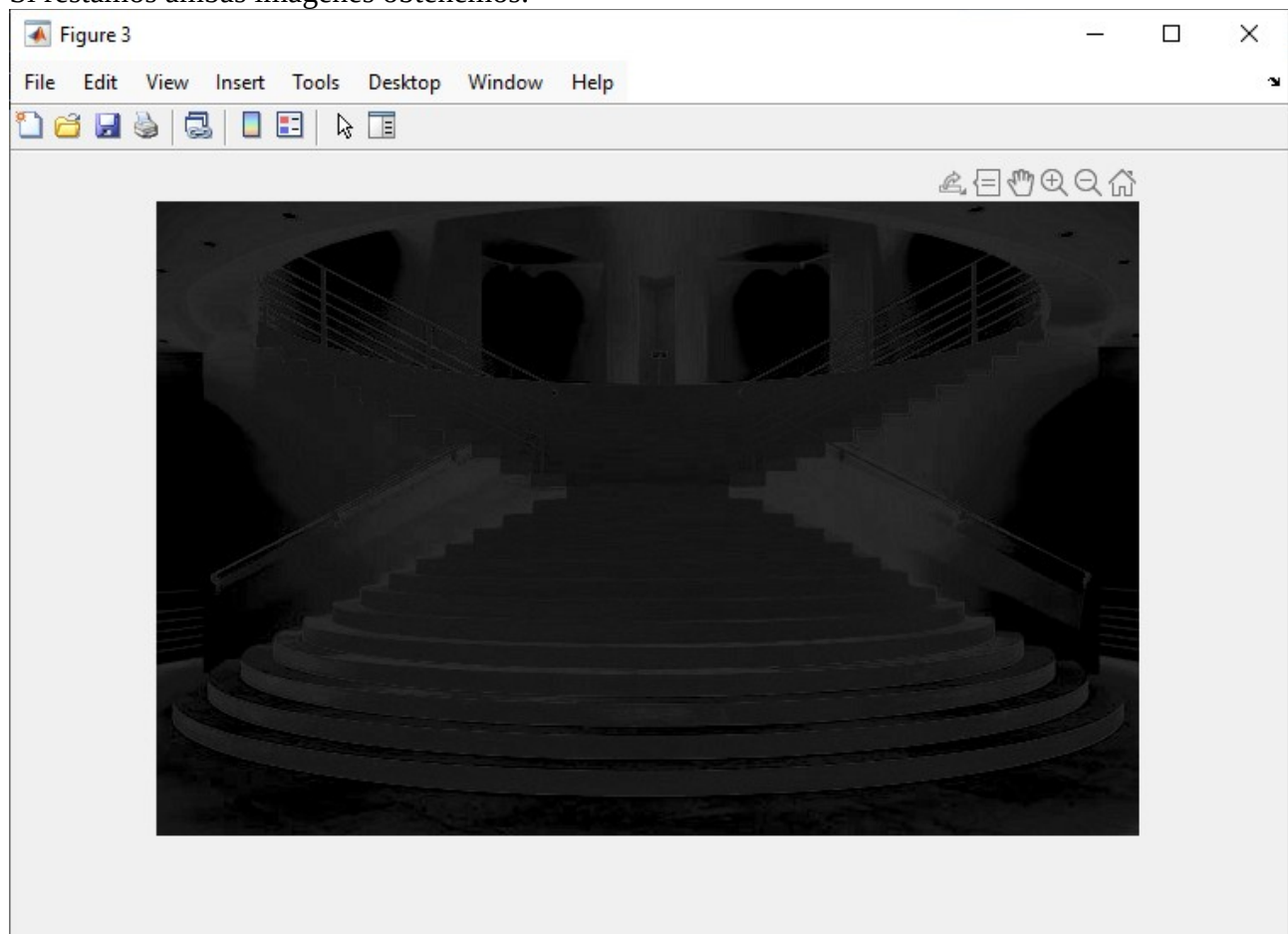


Si ecualizamos la imagen obtenemos:



Aunque no se aprecia demasiado, se puede notar que la imagen es un poco mas oscura, aunque si miramos los histogramas vemos como la imagen original tiende a ser mas clara, por lo que al ecualizarla, aumentan los valores oscuros.

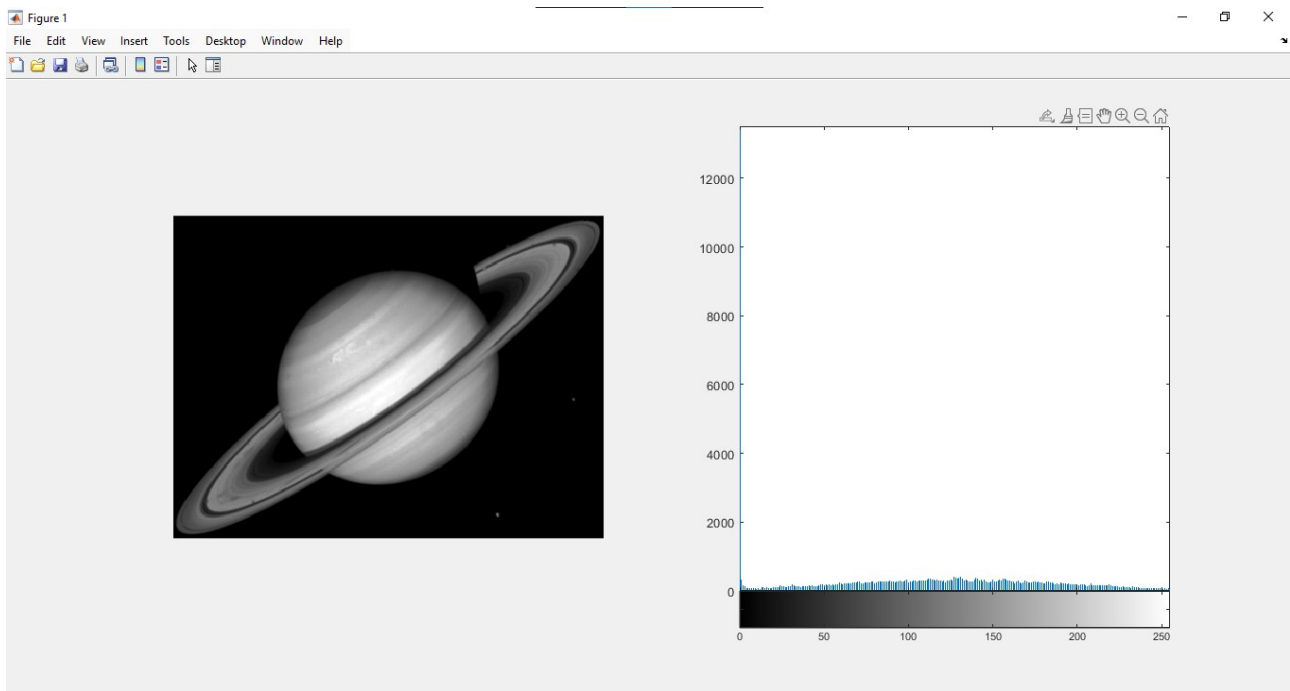
Si restamos ambas imágenes obtenemos:



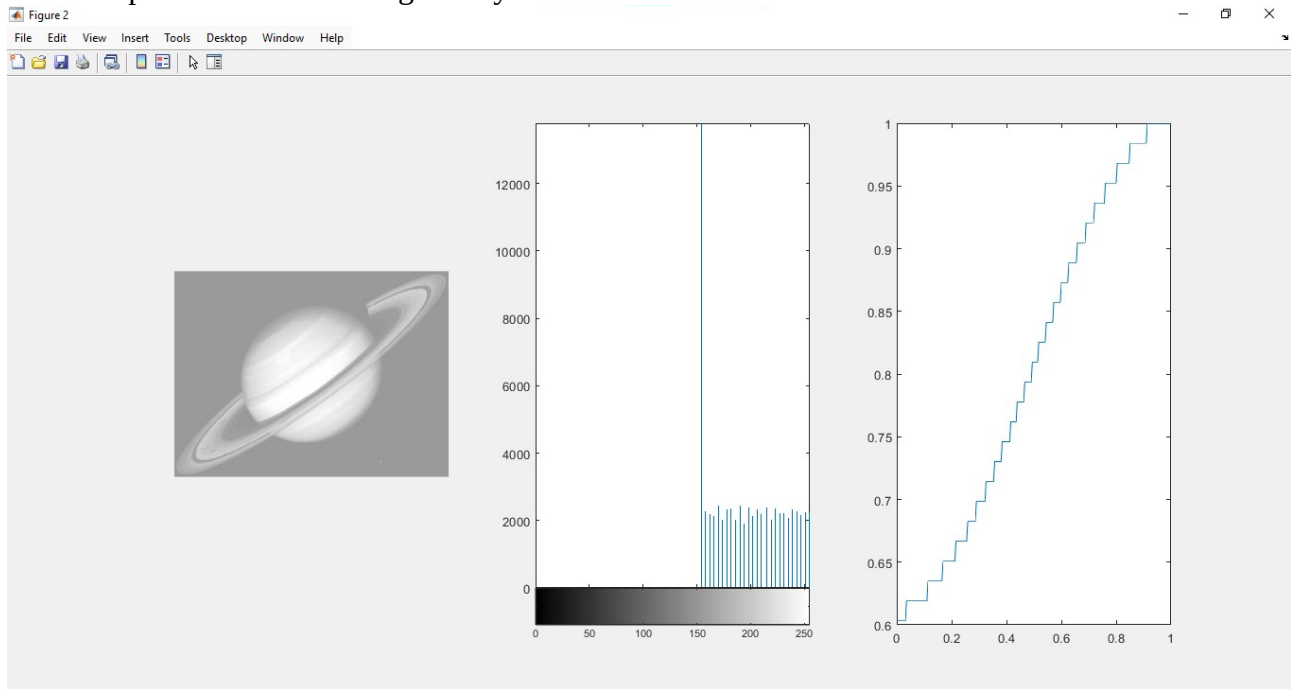
**B.**

```
I=imread(' ../Imagenes/saturno.tif');  
figure  
subplot(1,2,1)  
imshow(I);  
subplot(1,2,2)  
imhist(I);  
figure  
subplot(1,3,1)  
imshow(histeq(I))  
subplot(1,3,2)  
imhist(histeq(I))  
%Ecuálizamos la imagen y obtenemos la imagen y el vector de la ecualización  
[J,T]=histeq(I);  
subplot(1,3,3)  
%Ploteamos el vector para ver como ha afectado a la imagen  
plot((0:255)/255,T);
```

La imagen original se ve así:



Como se puede ver es una imagen muy oscura. Si la ecualizamos obtenemos:



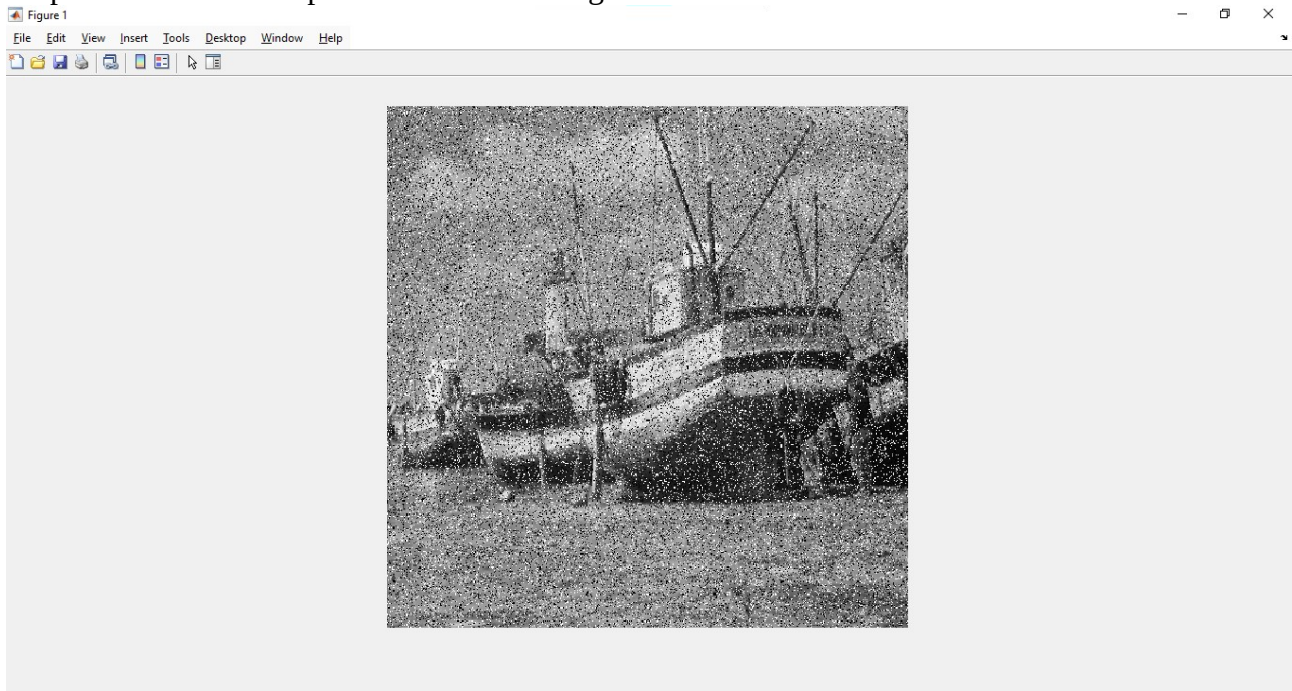
Las diferencias son claras, la imagen es mucho clara, todo el histograma se ha movido a la izquierda por lo que hemos perdido bastante contraste y detalle. A la derecha del todo vemos como la función ha igualado la imagen.

## 7. Restauración mediante filtrado en el dominio espacial

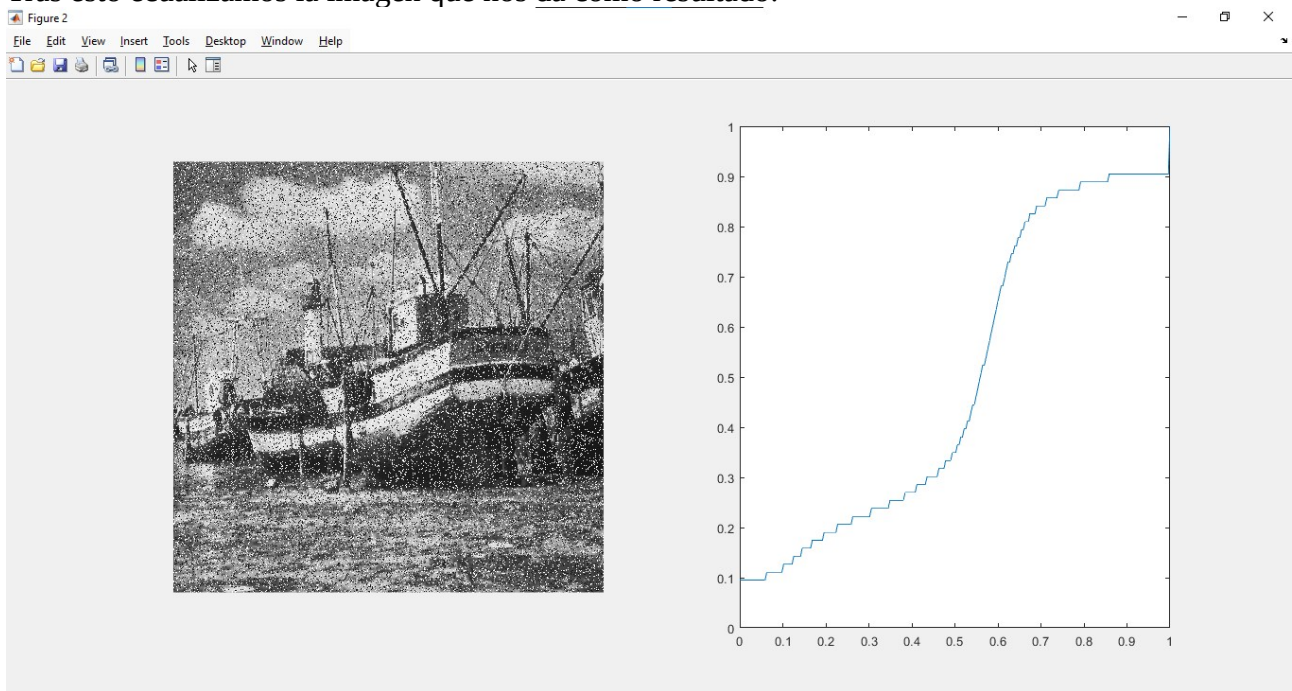
### A.

```
I=imread(' ../Imagenes/boat.512.tiff');  
%Le añadimos ruido a la imagen  
%Le tenemos que decir que metodo queremos y cuanto ruido queremos añadir  
I=imnoise(I,'salt & pepper',0.2);  
figure, imshow(I)  
%Ecualizamos la imagen  
[J,T]=histeq(I);  
figure  
subplot(1,2,1)  
imshow(J)  
subplot(1,2,2)  
plot((0:255)/255,T);  
%Le aplicamos un filtro de mediana de tamaño [n x n]  
%Cuanto mayor sea el tamaño, mas difuminada estara la imagen  
M=medfilt2(J,[5 5]);  
figure  
imshow(M)  
%Creamos un filtro del tipo que le digamos de tamaño [n x n]  
%Cuanto mayor sea el tamaño, mas difuminada estara la imagen  
g=fspecial('average',[9 9]);  
%Le aplicamos el filtro y hacemos que los valores de la imagen esten en el  
%rango [0,1]  
M=filter2(g,J)/255;  
figure  
imshow(M)
```

Lo primero a hacer es aplicarle ruido a la imagen. Esto nos da como resultado:

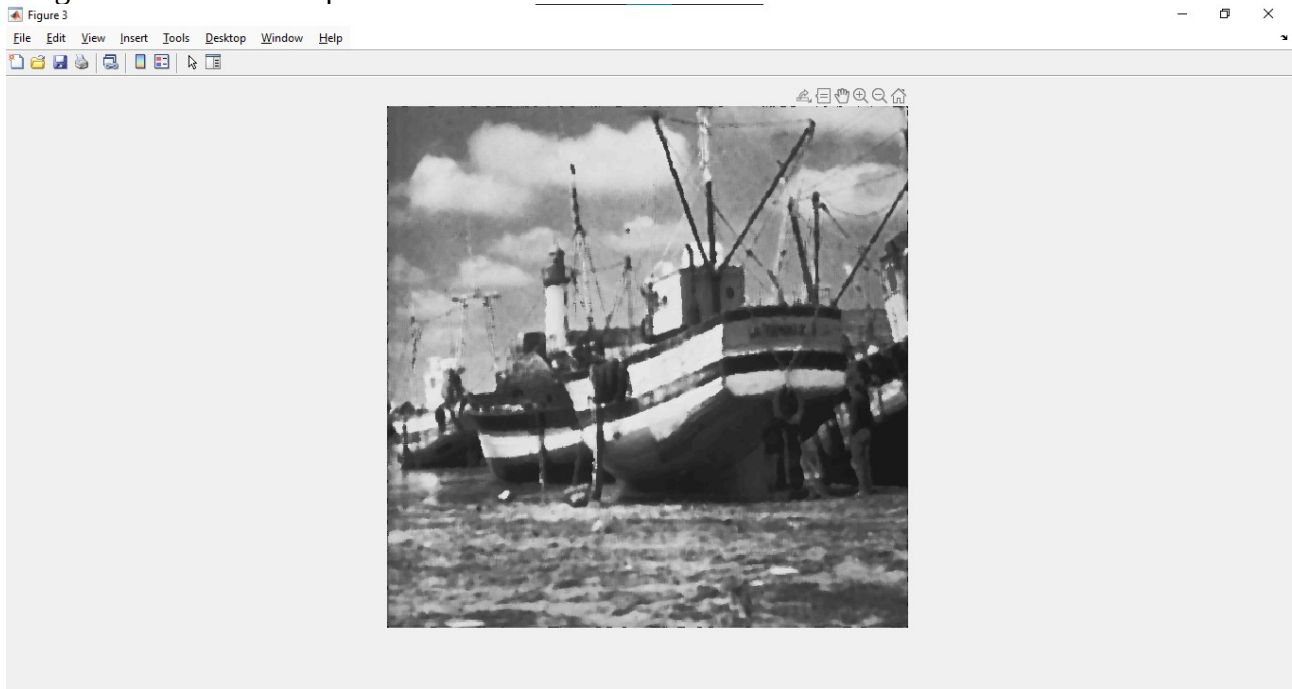


Tras esto ecualizamos la imagen que nos da como resultado:



A la derecha tenemos como la función usada ha ecualizado la imagen.

Ahora tenemos que restaurar la imagen, primero vamos a probar con un filtro de paso baja como el filtro mediana, para este caso he usado uno de tamaño 5x5 aunque ha dado como resultado una imagen bastante borrosa pero sin ruido: \_\_\_\_\_



Otro filtro a usar es el de la media, que he usado del mismo tamaño:



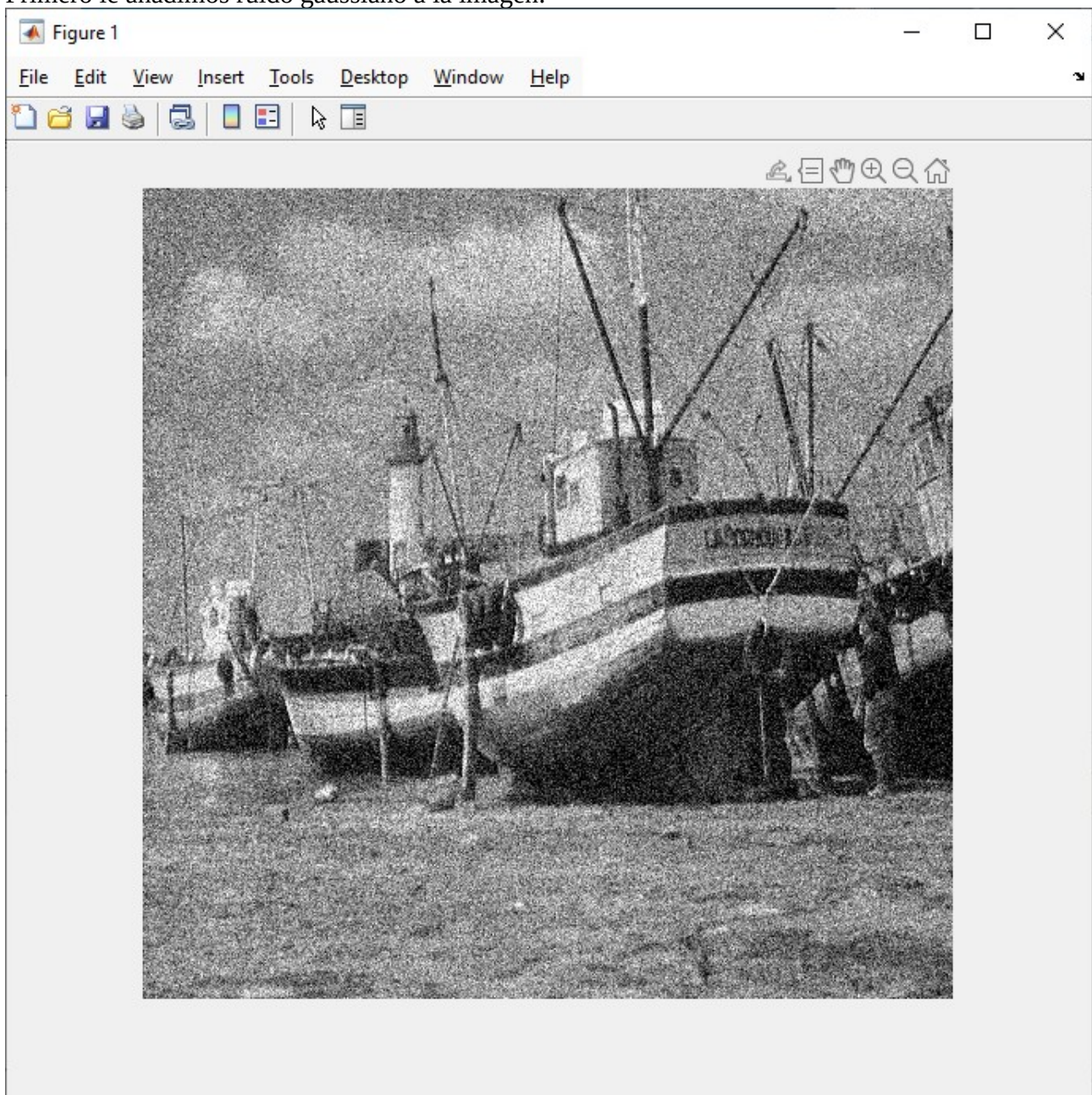
Que atenúa algo el ruido pero lo sigue sin quitar.



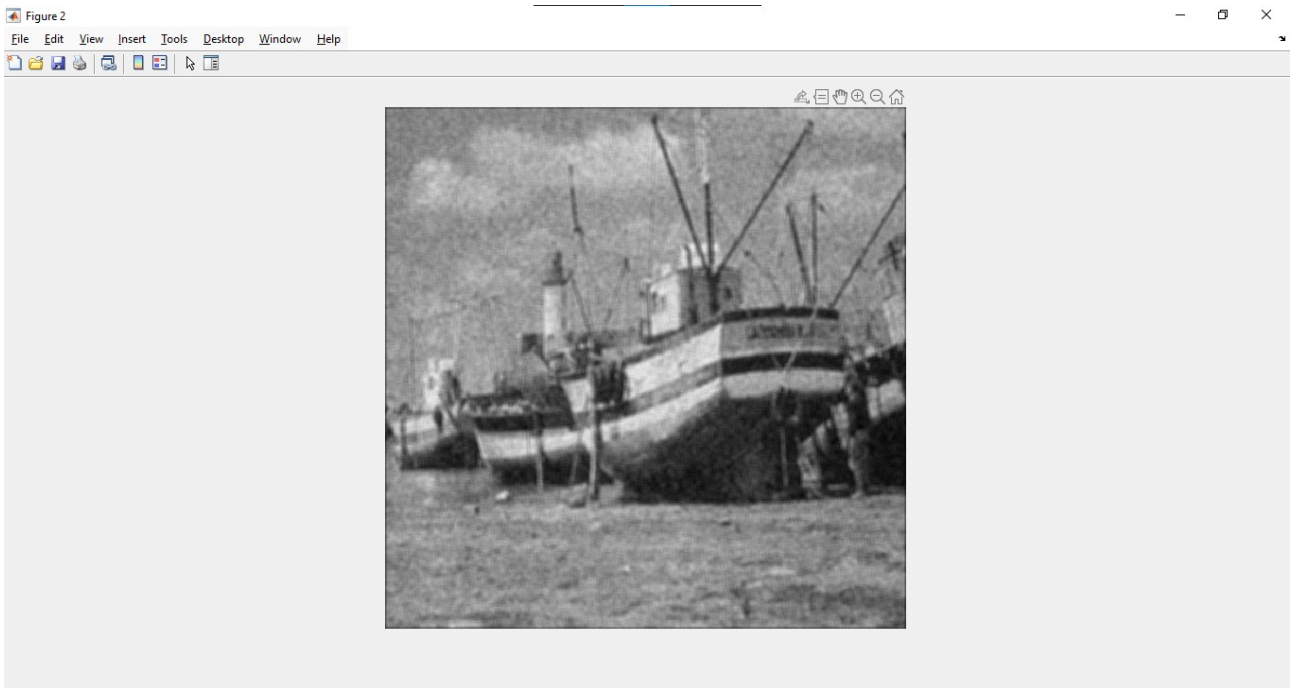
## B.

```
I=imread(' ../Imagenes/boat.512.tiff');  
%Añadimos ruido gaussiano a la imagen  
J=imnoise(I,'gaussian',0,0.02);  
figure,imshow(J)  
%Creamos un filtro de media  
g=fspecial('average',[5 5]);  
%Le aplicamos el filtro a la imagen y hacemos que los valores esten en el  
%rango [0, 1]  
M=filter2(g,J)/255;  
figure,imshow(M)  
%Le aplicamos a la imagen un filtro de mediana  
M1=medfilt2(J,[5 5]);  
figure, imshow(M1)
```

Primero le añadimos ruido gaussiano a la imagen:



Tras esto le aplicamos un filtro media de tamaño 5x5:



Como se puede ver, atenúa algo el ruido pero no lo quita del todo. También podemos usar un filtro mediana:



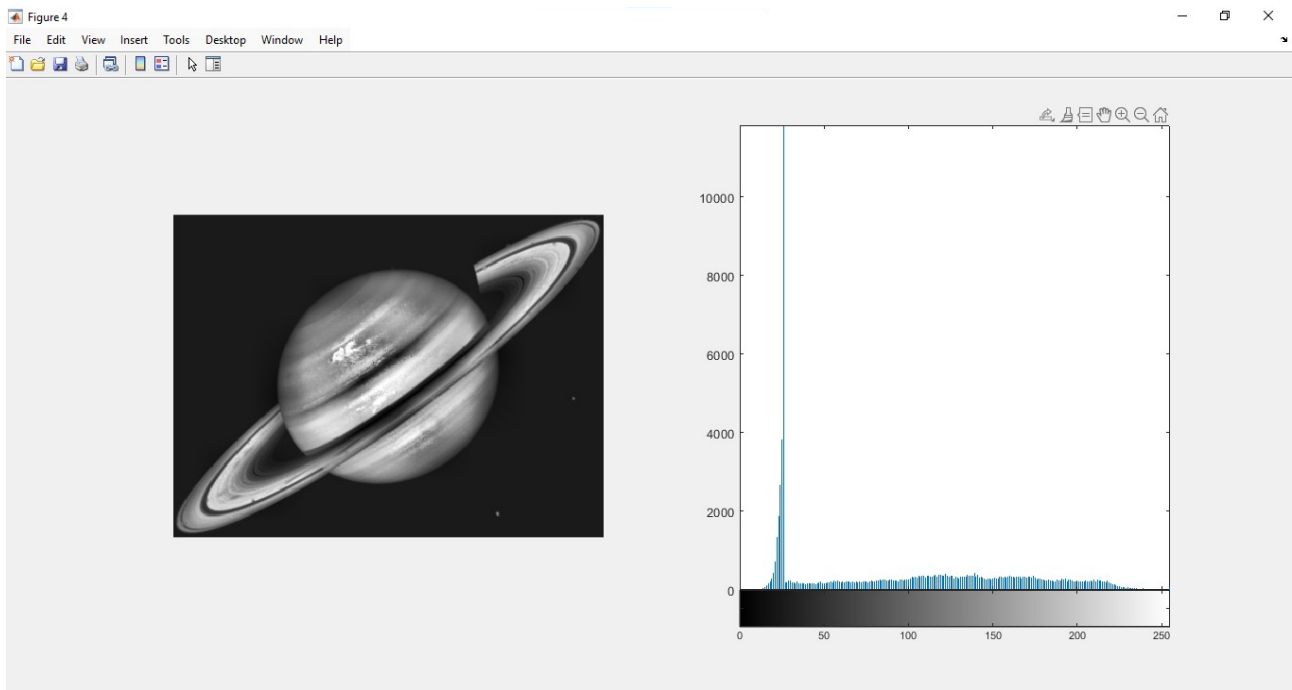
Que nos da un resultado bastante parecido al del filtro media.



# Preguntas

**1. Como alternativa al uso de histeq, puede realizar una ecualización adaptativa de histograma limitada por el contraste (CLAHE) utilizando la función adapthisteq. Pruébala y compara las diferencias en el apartado 6b.**

```
subplot(1,2,1)
imshow(adapthisteq(I, ClipLimit= 0.1))
subplot(1,2,2)
imhist(adapthisteq(I, ClipLimit= 0.1))
```

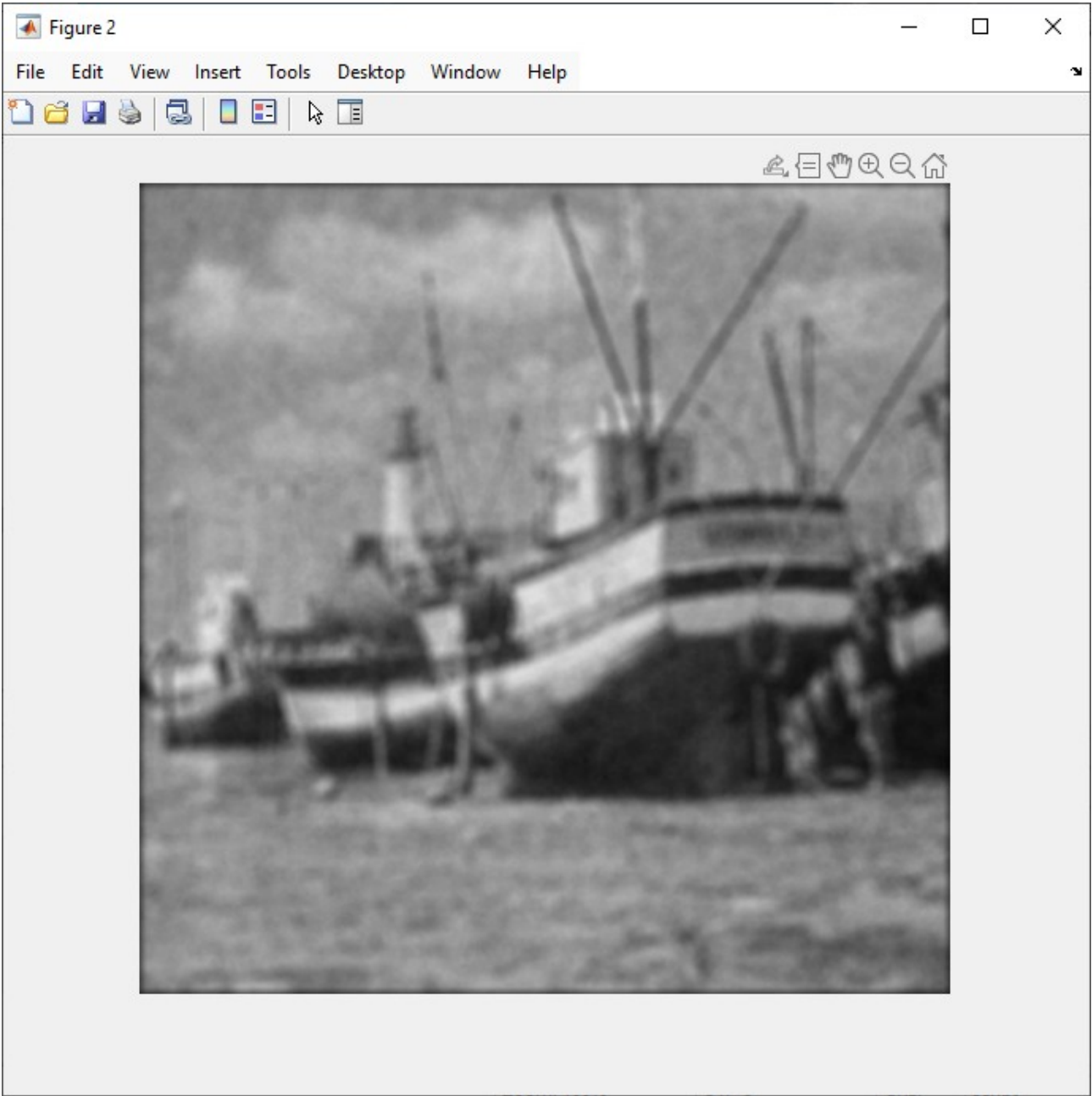


Como podemos ver, para este tipo de imagen funciona mejor que la otra función para ecualizar imágenes ya que aumenta el contraste de manera adaptativa.

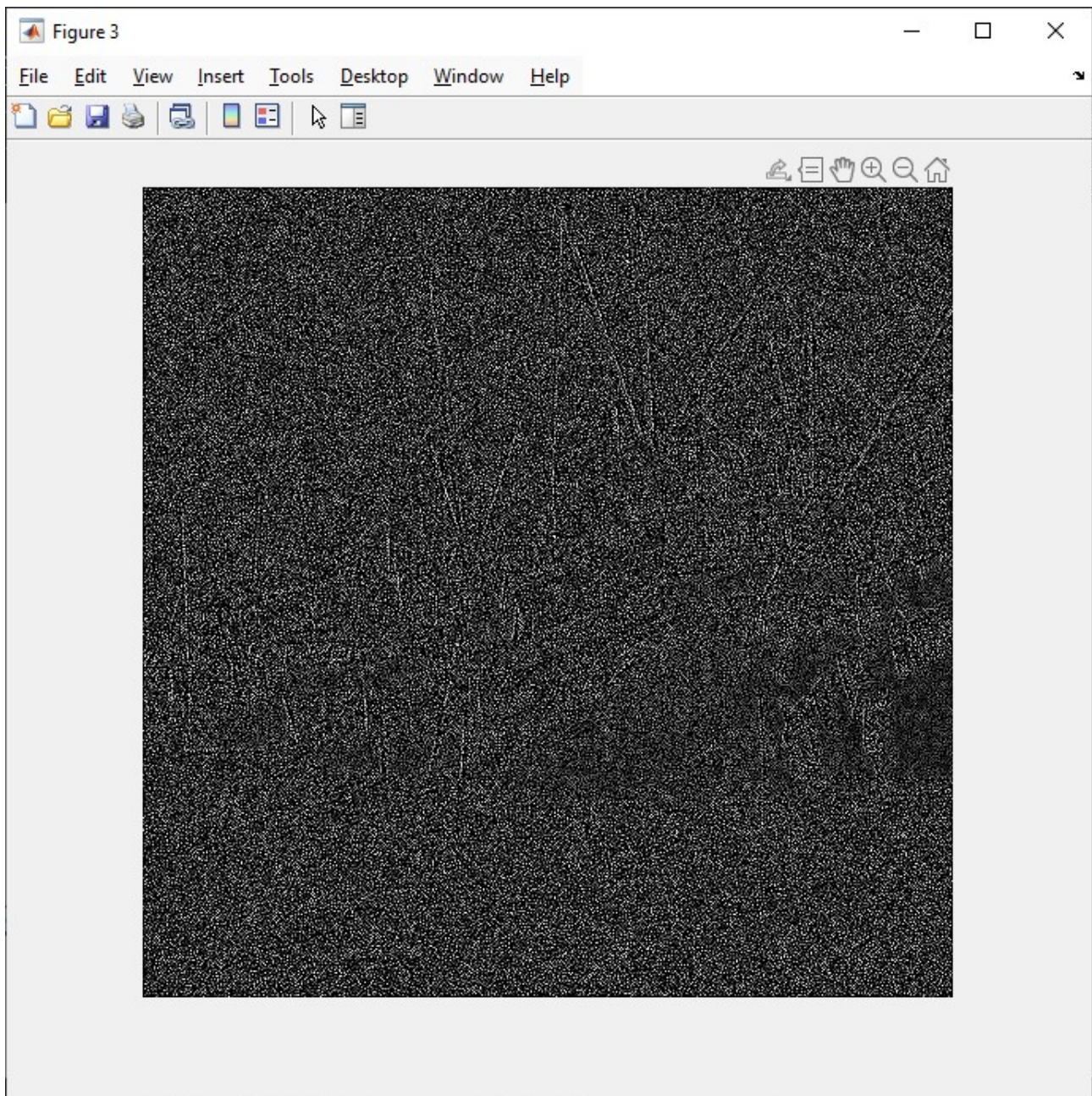
**2. En el apartado 7b, trata de usar otros filtros que defina la función fspecial para restaurar la imagen y compáralos.**

```
I=imread(' ../Imagenes/boat.512.tiff');
%Añadimos ruido gaussiano a la imagen
J=imnoise(I,'gaussian',0,0.02);
figure,imshow(J)
a=fspecial('disk');
b=filter2(a,J)/255;
figure,imshow(b)
c=fspecial('laplacian');
d=filter2(c,J)/255;
figure,imshow(d)
e=fspecial('motion');
f=filter2(e,J)/255;
figure,imshow(f)
```

El de tipo disco nos da como resultado una imagen muy borrosa:



El de tipo laplaciano nos devuelve una imagen que de por si no sirve para mucho pero es un tipo de filtro que es usado generalmente para detección de bordes o resaltar detalles:



Por ultimo, use el filtro de tipo motion, que aplica a la imagen un filtro como si se hubiera capturado en movimiento por lo que parece movida:

