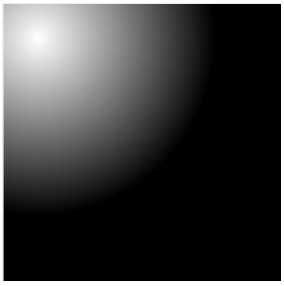
Procesamiento de Imágenes

1. Representación de imágenes digitales monocromáticas

a) Construye la matriz vinculada a la imagen de la figura 1

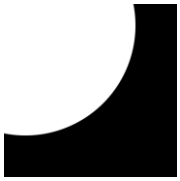


400x400 double

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	0.7282	0.7310	0.7337	0.7364	0.7391	0.7417	0.7443	0.7469	0.7494	0.7519	0.7544	0.7568	0.
2	0.7310	0.7338	0.7366	0.7393	0.7420	0.7446	0.7473	0.7499	0.7524	0.7550	0.7575	0.7599	0.
3	0.7337	0.7366	0.7393	0.7421	0.7448	0.7475	0.7502	0.7528	0.7554	0.7580	0.7605	0.7630	0.
4	0.7364	0.7393	0.7421	0.7449	0.7476	0.7504	0.7531	0.7557	0.7584	0.7609	0.7635	0.7660	0.
5	0.7391	0.7420	0.7448	0.7476	0.7504	0.7532	0.7559	0.7586	0.7613	0.7639	0.7665	0.7690	0.
6	0.7417	0.7446	0.7475	0.7504	0.7532	0.7560	0.7587	0.7615	0.7642	0.7668	0.7694	0.7720	0.
7	0.7443	0.7473	0.7502	0.7531	0.7559	0.7587	0.7615	0.7643	0.7670	0.7697	0.7723	0.7750	0.
8	0.7469	0.7499	0.7528	0.7557	0.7586	0.7615	0.7643	0.7671	0.7698	0.7725	0.7752	0.7779	0.
9	0.7494	0.7524	0.7554	0.7584	0.7613	0.7642	0.7670	0.7698	0.7726	0.7754	0.7781	0.7808	0.
10	0.7519	0.7550	0.7580	0.7609	0.7639	0.7668	0.7697	0.7725	0.7754	0.7782	0.7809	0.7836	0.
11	0.7544	0.7575	0.7605	0.7635	0.7665	0.7694	0.7723	0.7752	0.7781	0.7809	0.7837	0.7865	0.
12	0.7568	0.7599	0.7630	0.7660	0.7690	0.7720	0.7750	0.7779	0.7808	0.7836	0.7865	0.7893	0.
13	0.7592	0.7623	0.7654	0.7685	0.7715	0.7746	0.7775	0.7805	0.7834	0.7863	0.7892	0.7920	0.
14	0.7616	0.7647	0.7678	0.7709	0.7740	0.7771	0.7801	0.7831	0.7860	0.7890	0.7919	0.7947	0.
15	0.7639	0.7670	0.7702	0.7733	0.7764	0.7795	0.7826	0.7856	0.7886	0.7916	0.7945	0.7974	0.
16	0.7661	0.7693	0.7725	0.7757	0.7788	0.7819	0.7850	0.7881	0.7911	0.7941	0.7971	0.8000	0.
17	0.7683	0.7716	0.7748	0.7780	0.7812	0.7843	0.7874	0.7905	0.7936	0.7966	0.7997	0.8026	0.

```
>> %Coordenadas del foco (a,b)
a=50;
b=50;
%Construcción imagen MxN
M=400; N=400;
for x=1:M;
    for y=1:N;
I(x,y)=(255-sqrt((x-a)^2+(y-b)^2))/255;
    end
end
imshow(I)
```

- **¿Qué ocurre si no dividimos por 255?**
  La escala de grises dejaría de producirse. Dividir entre 255 permite que cada píxel tenga una intensidad comprendida entre 0-1.
  Si no se dividiera, el degradado no se produciría y por tanto solo se mostraría una zona blanca y otra negra.



b) Construye la matriz vinculada a la imagen de la figura 2

35x50 double

	9	10	11	12	13	14	15	16	17	18	19	20	21
1	1	1	1	1	1	1	1	1	1	1	1	1	
2	1	1	1	1	1	1	1	1	1	1	1	1	
3	1	1	1	1	1	1	1	1	1	1	1	1	
4	1	1	1	1	1	1	1	1	1	1	1	1	
5	1	1	1	1	0	0	0	0	0	1	1	1	
6	1	1	1	1	0	0	0	0	0	1	1	1	
7	1	1	1	1	0	0	0	0	0	1	1	1	
8	1	1	1	1	0	0	0	0	0	1	1	1	
9	1	1	1	1	0	0	0	0	0	1	1	1	
10	1	1	1	1	0	0	0	0	0	1	1	1	
11	1	1	1	1	0	0	0	0	0	1	1	1	
12	1	1	1	1	0	0	0	0	0	1	1	1	
13	1	1	1	1	0	0	0	0	0	1	1	1	
14	1	1	1	1	0	0	0	0	0	1	1	1	
15	1	1	1	1	0	0	0	0	0	1	1	1	
16	1	1	1	1	0	0	0	0	0	1	1	1	
17	1	1	1	1	0	0	0	0	0	1	1	1	

```
>> %imagen de tamaño 35x50
I=ones(35,50);
I(5:24,13:17)=0;
imshow(I,'initialMagnification','fit')
>>
```

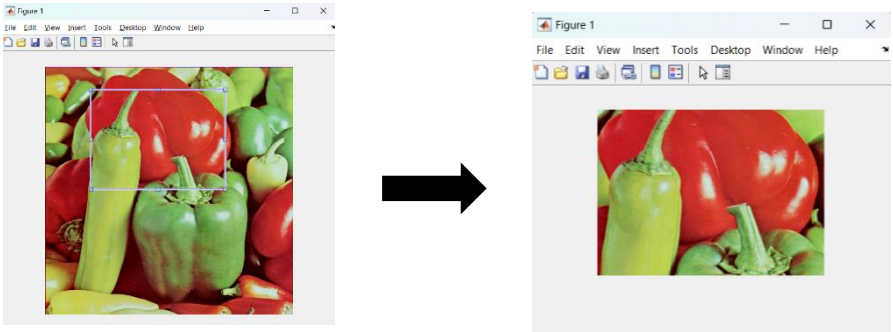
- **¿Qué significan cada uno de los parámetros de imshow? ¿Van juntos?**
  I: Representa la imagen que se quiere mostrar.
  initialMagnification: Controla la escala inicial de la imagen que se mostrará.
  fit: ajusta la imagen para que quepa dentro del área de visualización sin distorsionar su relación de aspecto.

c) Extrae de la imagen de la figura 3 la región de interés mostrada en la figura 4.

```
>> I=imread('5.1.14.tiff');
imshow(I)
J=imcrop;
%seleccionar con el ratón la región de
%interés
imshow(J)
```

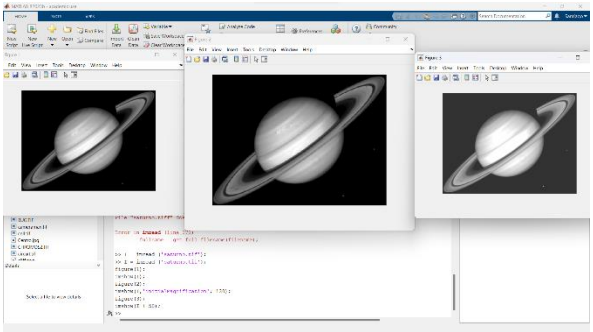
Imagen '5.1.14.tiff´

- **imcrop, ¿Funciona igual cuando la imagen es de color? ¿Recorta igual?**
  La función ´imcrop´ funciona de igual manera tanto para imágenes a color como monocromáticas.



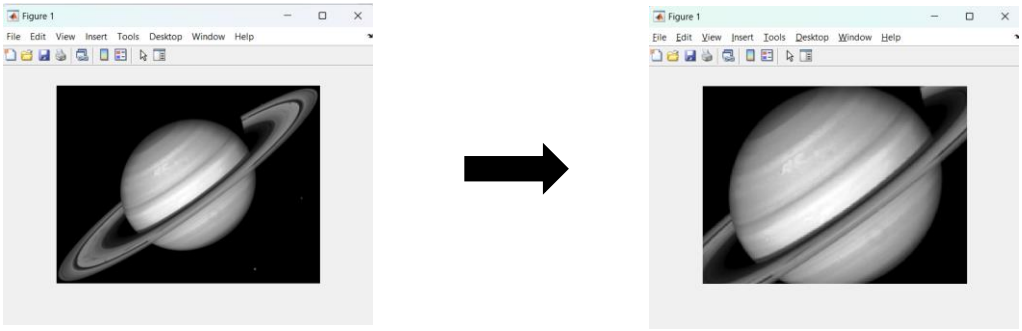
2) Operaciones aritméticas y geométricas con imágenes

a) Multiplica la imagen de la figura 5 por 1.3. Súmale 50 a cada píxel de la imagen de la figura Compara las imágenes resultantes. ¿Qué conclusiones sacas?



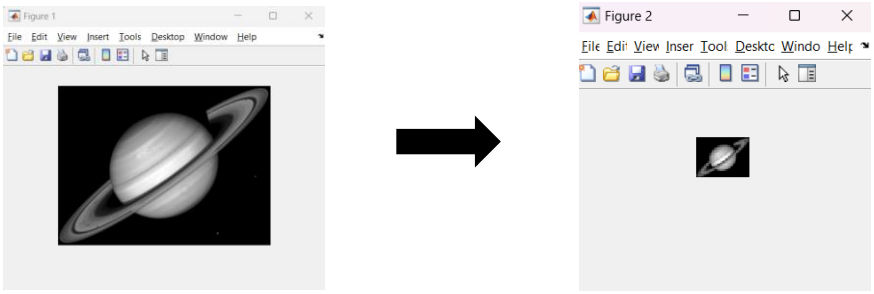
Se puede observar que la segunda imagen se satura más y la tercera aumenta la intensidad lumínica a cada uno de los píxeles.

b) Aumenta una cierta región de la imagen de la figura 5 para verla mejor.



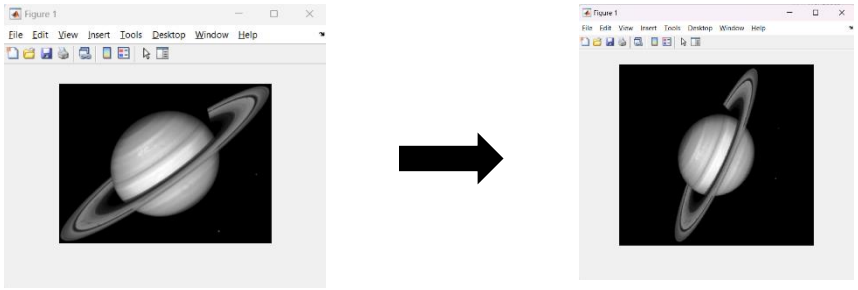
```
>> I=imread('saturno.tif');
imshow(I)
zoom on
```

c) Modifica el tamaño de la imagen de la figura 5.



```
>> I=imread('saturno.tif');
J=imresize(I,0.8,'bilinear');
imshow(J)
%también se puede poner
J1=imresize(I,[30 40]) %reduce la
%imagen al tamaño 40x30; 30 filas y
%40 columnas
figure, imshow(J1)
```

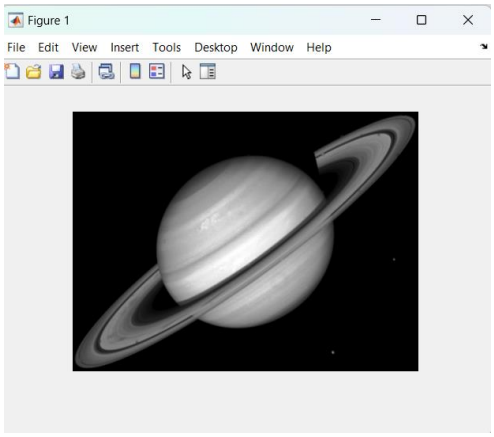
d) Gira 30º la imagen anterior (figura 6)

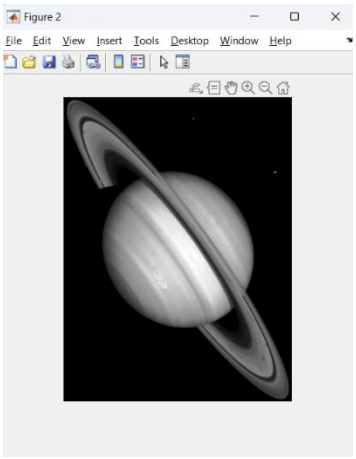


```
>> IR=imrotate(J,30);
imshow(IR)
```

- **Gira 90º la imagen sin utilizar el comando imrotate, es decir, implementa una función para devolver la imagen rotada.**

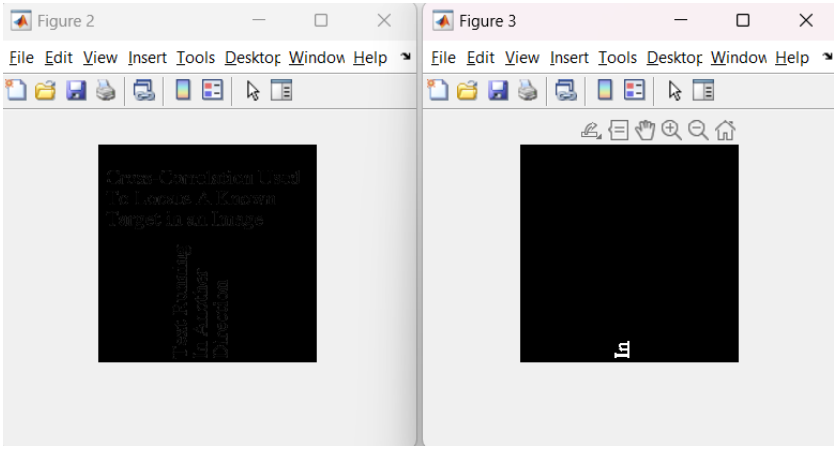
```
function imagen_rotada = rotar_imagen_90_grados(imagen)
    % Leer la imagen .tif
    imagen = imread(imagen);

    % Convertir la imagen a una matriz
    if ndims(imagen) == 3 % Si es una imagen a color
        imagen = rgb2gray(imagen); % Convertir a escala de grises
    else
        imagen = imagen; % La imagen ya está en escala de grises
    end
    % Obtener el tamaño de la imagen original
    [filas, columnas, ~] = size(imagen);

    % Transponer la imagen
    imagen_transpuesta = transpose(imagen);

    % Invertir las filas para rotación en sentido antihorario
    imagen_rotada = flipud(imagen_transpuesta);
    imshow(imagen_rotada);
end

Command Window
>> rotar_imagen_90_grados('saturno.tif')
```





3) Conectividad de píxeles

a) Determina el objeto de la imagen de la figura 7 que está en la posición (columna 90, fila 197) y el que está en la posición (16, 67), suponiendo que un objeto viene dado por un conjunto de píxeles de igual tonalidad y conectados por vecindad (entornos de 8 vecinos).




```
>> I=imread('text.tif');
c=[90 16]; r=[197 67];
I0=bwselect(I,c,r,8);
imshow(I)
figure, imshow(I0)
```